libquentier

Generated by Doxygen 1.9.4

Chapter 1

libquentier

Set of Qt/C++ APIs for feature rich desktop clients for Evernote service

1.1 What's this

This library presents a set of Qt/C++ APIs useful for applications working as feature rich desktop clients for Evernote service. The most important and useful components of the library are the following:

- Local storage - persistence of data downloaded from Evernote service in a local SQLite database
- Synchronization - the logics of exchanging new and/or modified data with Evernote service
- Note editor - the UI component capable for notes displaying and editing

The library is based on the lower level functionality provided by `QEverCloud` library. It also serves as the functional core of `Quentier` application.

1.1.1 WARNING: libquentier is in alpha state right now, neither API nor ABI can be considered stable yet!

1.2 How to build/install

Please see the building/installation guide.

1.3 How to contribute

Please see the contribution guide for detailed info.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

File Index

4.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 5

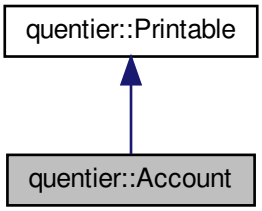
Class Documentation

5.1 quentier::Account Class Reference

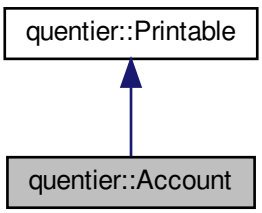
The Account class encapsulates some details about the account: its name, whether it is local or synchronized to Evernote and for the latter case - some additional details like upload limit etc.

`#include <Account.h>`

Inheritance diagram for quentier::Account:



Collaboration diagram for quentier::Account:



Public Types

- enum class **Type** { **Local** , **Evernote** }
- enum class **EvernoteAccountType** { **Free** , **Plus** , **Premium** , **Business** }

Public Member Functions

- **Account** (QString name, Type type, qevercloud::UserID userId=-1, EvernoteAccountType evernoteAccountType=Evernote↩ AccountType::Free, QString evernoteHost={}, QString shardId={})
- **Account** (const Account &other)
- **Account** (Account &&other) noexcept
- Account & **operator=** (const Account &other)
- Account & **operator=** (Account &&other) noexcept
- bool **operator==** (const Account &other) const noexcept
- bool **operator!=** (const Account &other) const noexcept
- bool isEmpty () const
- QString name () const
- void **setName** (QString name)

  setName sets the username to the account
- QString displayName () const
- void setDisplayName (QString displayName)
- Type type () const
- qevercloud::UserID id () const
- EvernoteAccountType evernoteAccountType () const
- QString evernoteHost () const
- QString shardId () const
- void **setEvernoteAccountType** (EvernoteAccountType evernoteAccountType)
- void **setEvernoteHost** (QString evernoteHost)
- void **setShardId** (QString shardId)
- qint32 **mailLimitDaily** () const
- qint64 **noteSizeMax** () const
- qint64 **resourceSizeMax** () const
- qint32 **linkedNotebookMax** () const
- qint32 **noteCountMax** () const
- qint32 **notebookCountMax** () const
- qint32 **tagCountMax** () const
- qint32 **noteTagCountMax** () const
- qint32 **savedSearchCountMax** () const
- qint32 **noteResourceCountMax** () const
- void **setEvernoteAccountLimits** (const qevercloud::AccountLimits &limits)
- QTextStream & print (QTextStream &strm) const override

Friends

- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, Type type)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, Type type)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, EvernoteAccountType type)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, EvernoteAccountType type)

5.1.1 Detailed Description

The Account class encapsulates some details about the account: its name, whether it is local or synchronized to Evernote and for the latter case - some additional details like upload limit etc.

5.1.2 Member Function Documentation

5.1.2.1 displayName()

```
QString quentier::Account::displayName ( ) const
```

Returns

Printable user's name that is not used to uniquely identify the account, so this name may repeat across different local and Evernote accounts

5.1.2.2 evernoteAccountType()

```
EvernoteAccountType quentier::Account::evernoteAccountType ( ) const
```

Returns

The type of the Evernote account; if applied to free account, returns "Free"

5.1.2.3 evernoteHost()

```
QString quentier::Account::evernoteHost ( ) const
```

Returns

The Evernote server host with which the account is associated

5.1.2.4 id()

```
qevercloud::UserID quentier::Account::id ( ) const
```

Returns

User id for Evernote accounts, -1 for local accounts (as the concept of user id is not defined for local accounts)

5.1.2.5 isEmpty()

```
bool quentier::Account::isEmpty ( ) const
```

Returns

True if either the account is local but the name is empty or if the account is Evernote but user id is negative; in all other cases return false

5.1.2.6 name()

```
QString quentier::Account::name ( ) const
```

Returns

Username for either local or Evernote account

5.1.2.7 print()

```
QTextStream & quentier::Account::print (
            QTextStream & strm ) const  [override], [virtual]
```

Implements quentier::Printable.

5.1.2.8 setDisplayName()

```
void quentier::Account::setDisplayName (
            QString displayName )
```

Set the printable name of the account

5.1.2.9 shardId()

```
QString quentier::Account::shardId ( ) const
```

Returns

Shard id for Evernote accounts, empty string for local accounts (as the concept of shard id is not defined for local accounts)

5.1.2.10 type()

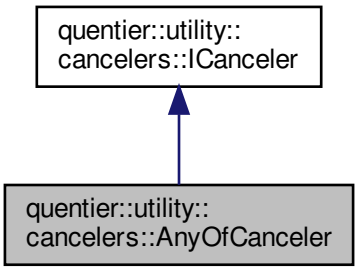
```
Type quentier::Account::type ( ) const
```

Returns

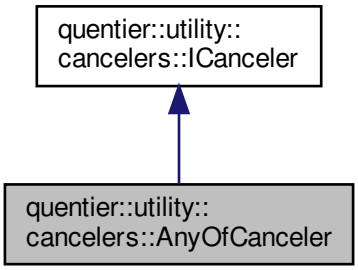
The type of the account: either local of Evernote

5.2 quentier::utility::cancelers::AnyOfCanceler Class Reference

Inheritance diagram for quentier::utility::cancelers::AnyOfCanceler:



Collaboration diagram for quentier::utility::cancelers::AnyOfCanceler:



Public Member Functions

- **AnyOfCanceler** (QList< ICancelerPtr > cancelers)
- **AnyOfCanceler** (AnyOfCanceler &&other) noexcept
- AnyOfCanceler & **operator=** (AnyOfCanceler &&other) noexcept
- bool isCanceled () const noexcept override

5.2.1 Member Function Documentation

5.2.1.1 isCanceled()

```
bool quentier::utility::cancelers::AnyOfCanceler::isCanceled ( ) const  [override], [virtual],
[noexcept]
```

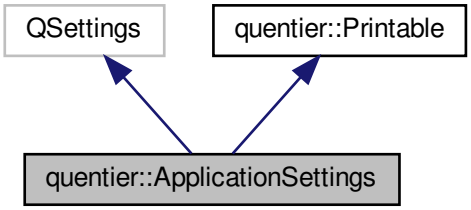
Implements quentier::utility::cancelers::ICanceler.

5.3 quentier::ApplicationSettings Class Reference

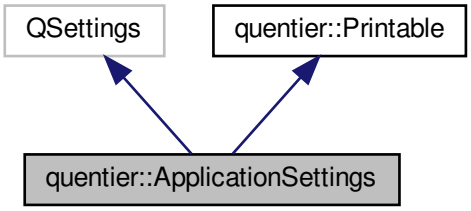
The ApplicationSettings class enhances the functionality of QSettings, in particular it simplifies the way of working with either application-wide or account-specific settings.

```
#include <ApplicationSettings.h>
```

Inheritance diagram for quentier::ApplicationSettings:



Collaboration diagram for quentier::ApplicationSettings:



Classes

- struct ArrayCloser
- struct GroupCloser

Public Member Functions

- ApplicationSettings (const QString &settingsName={})
- ApplicationSettings (const Account &account, const QString &settingsName={})
- ApplicationSettings (const Account &account, const char *settingsName, int settingsNameSize=-1)
- ApplicationSettings (const Account &account, std::string_view settingsName)
- ~ApplicationSettings () override
- void beginGroup (const QString &prefix)
- void beginGroup (const char *prefix, int size=-1)
- void beginGroup (std::string_view prefix)
- int beginReadArray (const QString &prefix)
- int beginReadArray (const char *prefix, int size=-1)
- int beginReadArray (std::string_view prefix)
- void beginWriteArray (const QString &prefix, int arraySize=-1)
- void beginWriteArray (const char *prefix, int arraySize=-1, int prefixSize=-1)
- void beginWriteArray (std::string_view prefix, int arraySize=-1)
- bool contains (const QString &key) const
- bool contains (const char *key, int size=-1) const
- bool contains (std::string_view key) const
- void remove (const QString &key)
- void remove (const char *key, int size=-1)
- void remove (std::string_view key)
- void setValue (const QString &key, const QVariant &value)
- void setValue (const char *key, const QVariant &value, int keySize=-1)
- void setValue (std::string_view key, const QVariant &value)
- QVariant value (const QString &key, const QVariant &defaultValue={}) const
- QVariant value (const char *key, const QVariant &defaultValue={}, int keySize=-1) const
- QVariant value (std::string_view key, const QVariant &defaultValue={}) const
- QTextStream & print (QTextStream &strm) const override

5.3.1 Detailed Description

The ApplicationSettings class enhances the functionality of QSettings, in particular it simplifies the way of working with either application-wide or account-specific settings.

5.3.2 Constructor & Destructor Documentation

5.3.2.1 ApplicationSettings() [1/4]

```
quentier::ApplicationSettings::ApplicationSettings (
            const QString & settingsName = {} )  [explicit]
```

Constructor for application settings not being account-specific

Parameters

settingsName	If not empty, the created application settings would manage the settings stored in a file with a specific name within the common settings storage; otherwise they would be stored in the default settings file for the account

5.3.2.2 ApplicationSettings() [2/4]

```
quentier::ApplicationSettings::ApplicationSettings (
            const Account & account,
            const QString & settingsName = {} )  [explicit]
```

Constructor for application settings specific to the account

Parameters

account	The account for which the settings are to be stored or read
settingsName	If not empty, the created application settings would manage the settings stored in a file with a specific name within the account's settings storage; otherwise they would be stored in the default settings file for the account

5.3.2.3 ApplicationSettings() [3/4]

```
quentier::ApplicationSettings::ApplicationSettings (
            const Account & account,
            const char * settingsName,
            int settingsNameSize = -1 )
```

Constructor for application settings specific to the account

Parameters

account	The account for which the settings are to be stored or read
settingsName	If not nullptr, the created application settings would manage the settings stored in a file with a specific name within the account's settings storage; otherwise they would be stored in the default settings file for the account. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
settingsNameSize	Size of the settingsName string. If negative (the default), the settingsName size is taken to be strlen(settingsName)

5.3.2.4 ApplicationSettings() [4/4]

```
quentier::ApplicationSettings::ApplicationSettings (
```

```
            const Account & account,
            std::string_view settingsName )
```

Constructor for application settings specific to the account

Parameters

account	The account for which the settings are to be stored or read
settingsName	If not empty, the created application settings would manage the settings stored in a file with a specific name within the account's settings storage; otherwise they would be stored in the default settings file for the account. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8

5.3.2.5 ~ApplicationSettings()

```
quentier::ApplicationSettings::~ApplicationSettings ( )  [override]
```

Destructor

5.3.3 Member Function Documentation

5.3.3.1 beginGroup() [1/3]

```
void quentier::ApplicationSettings::beginGroup (
            const char * prefix,
            int size = -1 )
```

Appends prefix to the current group. Overload of beginGroup accepting const char * and optionally the size of the string

Parameters

prefix	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
size	Size of the prefix sring. If negative (the default), the prefix size is taken to be stren(prefix).

5.3.3.2 beginGroup() [2/3]

```
void quentier::ApplicationSettings::beginGroup (
            const QString & prefix )
```

Appends prefix to the current group. The call is redirected to QSettings::beginGroup. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

prefix	String containing the prefix name

5.3.3.3 beginGroup() [3/3]

```
void quentier::ApplicationSettings::beginGroup (
            std::string_view prefix )
```

Appends prefix to the current group. Overload of beginGroup accepting std::std::string_view

Parameters

prefix	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
size	Size of the prefix sring. If negative (the default), the prefix size is taken to be stren(prefix).

5.3.3.4 beginReadArray() [1/3]

```
int quentier::ApplicationSettings::beginReadArray (
            const char * prefix,
            int size = -1 )
```

Adds prefix to the current group and starts reading from an array. Overload of beginReadArray accepting const char * and optionally the size of the string

Parameters

prefix	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
size	Size of the prefix sring. If negative (the default), the prefix size is taken to be stren(prefix)

5.3.3.5 beginReadArray() [2/3]

```
int quentier::ApplicationSettings::beginReadArray (
            const QString & prefix )
```

Adds prefix to the current group and starts reading from an array. The call is redirected to QSettings::beginRead↩ Array. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

prefix	String containing the prefix name

Returns

The size of the array

5.3.3.6 beginReadArray() [3/3]

```
int quentier::ApplicationSettings::beginReadArray (
            std::string_view prefix )
```

Adds prefix to the current group and starts reading from an array. Overload of beginReadArray accepting std↩ ::string_view

Parameters

prefix	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8

5.3.3.7 beginWriteArray() [1/3]

```
void quentier::ApplicationSettings::beginWriteArray (
            const char * prefix,
            int arraySize = -1,
            int prefixSize = -1 )
```

Adds prefix to the current group and starts writing an array of size arraySize. Overload of beginWriteArray accepting const char * and optionally the size of the string

Parameters

prefix	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
arraySize	Size of the array to be written. If negative (the default), it is automatically determined based on the indexes of the entries written.
prefixSize	Size of the prefix sring. If negative (the default), the prefix size is taken to be stren(prefix)

5.3.3.8 beginWriteArray() [2/3]

```
void quentier::ApplicationSettings::beginWriteArray (
```

```
            const QString & prefix,
            int arraySize = -1 )
```

Adds prefix to the current group and starts writing an array of size arraySize. The call is redirected to QSettings↩ ::beginWriteArray. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

prefix	String containing the prefix name
arraySize	Size of the array to be written. If negative (the default), it is automatically determined based on the indexes of the entries written.

5.3.3.9 beginWriteArray() [3/3]

```
void quentier::ApplicationSettings::beginWriteArray (
            std::string_view prefix,
            int arraySize = -1 )
```

Adds prefix to the current group and starts writing an array of size arraySize. Overload of beginWriteArray accepting std::string_view

Parameters

prefix	String containing the prefix name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
arraySize	Size of the array to be written. If negative (the default), it is automatically determined based on the indexes of the entries written.

5.3.3.10 contains() [1/3]

```
bool quentier::ApplicationSettings::contains (
            const char * key,
            int size = -1 ) const
```

Overload of contains accepting const char $*$ and optionally the size of the string

Parameters

key	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
size	Size of the key sring. If negative (the default), the key size is taken to be stren(key)

Returns

True if there exists a setting called key; false otherwise

5.3.3.11 contains() [2/3]

```
bool quentier::ApplicationSettings::contains (
            const QString & key ) const
```

The call is redirected to QSettings::contains. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

key	The key being checked for presence

Returns

True if there exists a setting called key; false otherwise

5.3.3.12 contains() [3/3]

```
bool quentier::ApplicationSettings::contains (
            std::string_view key ) const
```

Overload of contains accepting std::string_view

Parameters

key	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8

Returns

True if there exists a setting called key; false otherwise

5.3.3.13 print()

```
QTextStream & quentier::ApplicationSettings::print (
            QTextStream & strm ) const  [override], [virtual]
```

Implements quentier::Printable.

5.3.3.14 remove() [1/3]

```
void quentier::ApplicationSettings::remove (
            const char * key,
            int size = -1 )
```

Removes the setting key and any sub-settings of key. Overload of remove accepting const char * and optionally the size of the string

Parameters

key	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
size	Size of the key sring. If negative (the default), the key size is taken to be stren(key).

5.3.3.15 remove() [2/3]

```
void quentier::ApplicationSettings::remove (
            const QString & key )
```

Removes the setting key and any sub-settings of key. The call is redirected to QSettings::remove. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

key	String containing the setting name

5.3.3.16 remove() [3/3]

```
void quentier::ApplicationSettings::remove (
            std::string_view key )
```

Removes the setting key and any sub-settings of key. Overload of remove accepting std::string_view

Parameters

key	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8

5.3.3.17 setValue() [1/3]

```
void quentier::ApplicationSettings::setValue (
            const char * key,
            const QVariant & value,
            int keySize = -1 )
```

Sets the value of setting. Overload of setValue accepting const char * and optionally the size of the string

Parameters

key	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
value	Value for setting key
keySize	Size of the key string. If negative (the default), the key size is taken to be strlen(key).

5.3.3.18 setValue() [2/3]

```
void quentier::ApplicationSettings::setValue (
            const QString & key,
            const QVariant & value )
```

Sets the value of setting. The call is redirected to QSettings::setValue. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

key	String containing the setting name
value	Value for setting key

5.3.3.19 setValue() [3/3]

```
void quentier::ApplicationSettings::setValue (
            std::string_view key,
            const QVariant & value )
```

Sets the value of setting. Overload of setValue accepting std::string_view

Parameters

key	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
value	Value for setting key

5.3.3.20 value() [1/3]

```
QVariant quentier::ApplicationSettings::value (
            const char * key,
            const QVariant & defaultValue = {},
            int keySize = -1 ) const
```

Fetches the value of setting. Overload of value accepting const char * and optionally the size of the string

Parameters

key	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
defautValue	Default value returned if the setting doesn't exist
keySize	Size of the key sring. If negative (the default), the key size is taken to be stren(key)

Returns

The value for setting key. If the setting doesn't exist, returns defaultValue. If no default value is specified, a default QVariant is returned.

5.3.3.21 value() [2/3]

```
QVariant quentier::ApplicationSettings::value (
            const QString & key,
            const QVariant & defaultValue = {} ) const
```

Fetches the value of setting. The call is redirected to QSettings::value. It is required in this class only to workaround hiding QSettings method due to overloads

Parameters

key	String containing the setting name
defautValue	Default value returned if the setting doesn't exist

Returns

The value for setting key. If the setting doesn't exist, returns defaultValue. If no default value is specified, a default QVariant is returned.

5.3.3.22 value() [3/3]

```
QVariant quentier::ApplicationSettings::value (
            std::string_view key,
            const QVariant & defaultValue = {} ) const
```

Fetches the value of setting. Overload of value accepting std::string_view

Parameters

key	String containing the setting name. Must be UTF-8 encoded as internally it is converted to QString via QString::fromUtf8
defautValue	Default value returned if the setting doesn't exist

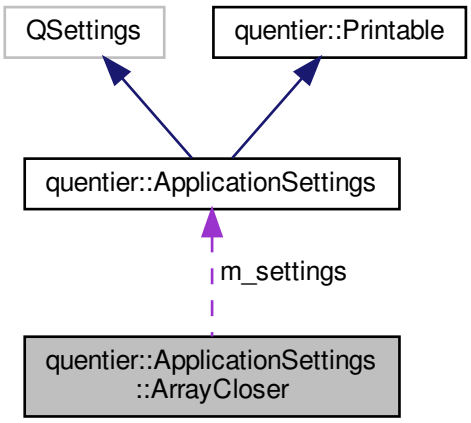
Returns

The value for setting key. If the setting doesn't exist, returns defaultValue. If no default value is specified, a default QVariant is returned.

5.4 quentier::ApplicationSettings::ArrayCloser Struct Reference

`#include <ApplicationSettings.h>`

Collaboration diagram for quentier::ApplicationSettings::ArrayCloser:



Public Member Functions

- **ArrayCloser** (ApplicationSettings &settings)

Public Attributes

- ApplicationSettings & **m_settings**

5.4.1 Detailed Description

Helper struct for RAII style of ensuring the array once began would be closed even if exception is thrown after beginning the array

5.5 quentier::synchronization::AuthenticationExpiredError Struct Reference

`#include <Errors.h>`

5.5.1 Detailed Description

Authentication expired error indicates that used authentication token has expired so authentication should be repeated before the next attempt to run synchronization.

5.6 quentier::synchronization::ISyncConflictResolver::ConflictResolution Struct Reference

The ConflictResolution struct is a namespace inside which several other structs determining actual conflict resolutions.

`#include <ISyncConflictResolver.h>`

Classes

- struct IgnoreMine

  The IgnoreMine conflict resolution means "use theirs version and ignore mine version as it doesn't really conflict with theirs version".
- struct MoveMine

  The MoveMine conflict resolution means "before using theirs version change mine version as specified". Note: the data item inside this conflict resolution might refer to something different than mine version passed to the conflict resolution function. It can be that way because the actual conflict might be with another local data item instead of the passed one.
- struct UseMine

  The UseMine conflict resolution means "override theirs version with mine version".
- struct UseTheirs

  The UseTheirs conflict resolution means "override mine version with theirs version".

5.6.1 Detailed Description

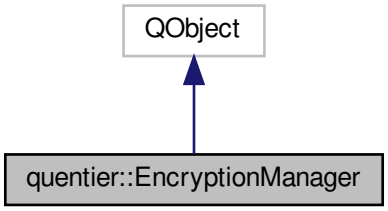
The ConflictResolution struct is a namespace inside which several other structs determining actual conflict resolutions.

5.7 quentier::EncryptionManager Class Reference

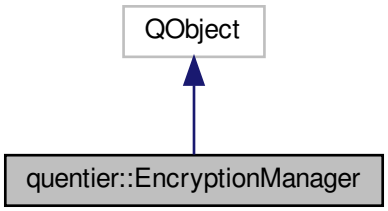
The EncryptionManager class provides both synchronous methods to encrypt or decrypt given text with password, cipher and key length and their signal-slot based potentially asynchronous counterparts.

`#include <EncryptionManager.h>`

Inheritance diagram for quentier::EncryptionManager:



Collaboration diagram for quentier::EncryptionManager:



Public Slots

- void **onDecryptTextRequest** (QString encryptedText, QString passphrase, QString cipher, size_t keyLength, QUuid requestId)
- void **onEncryptTextRequest** (QString textToEncrypt, QString passphrase, QString cipher, size_t keyLength, QUuid requestId)

Signals

- void **decryptedText** (QString text, bool success, ErrorString errorDescription, QUuid requestId)
- void **encryptedText** (QString encryptedText, bool success, ErrorString errorDescription, QUuid requestId)

Public Member Functions

- **EncryptionManager** (QObject *parent=nullptr)
- bool **decrypt** (const QString &encryptedText, const QString &passphrase, const QString &cipher, size_t keyLength, QString &decryptedText, ErrorString &errorDescription)
- bool **encrypt** (const QString &textToEncrypt, const QString &passphrase, QString &cipher, size_t &key↩ Length, QString &encryptedText, ErrorString &errorDescription)

5.7.1 Detailed Description

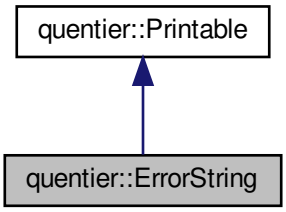
The EncryptionManager class provides both synchronous methods to encrypt or decrypt given text with password, cipher and key length and their signal-slot based potentially asynchronous counterparts.

5.8 quentier::ErrorString Class Reference

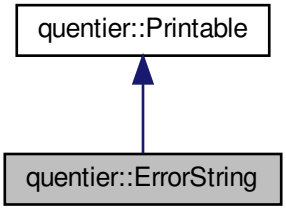
The ErrorString class encapsulates two (or more) strings which are meant to contain translatable (base) and non-translatable (details) parts of the error description.

`#include <ErrorString.h>`

Inheritance diagram for quentier::ErrorString:



Collaboration diagram for quentier::ErrorString:



Public Member Functions

- **ErrorString** (const char *error=nullptr)
- **ErrorString** (const QString &error)
- **ErrorString** (const ErrorString &other)
- **ErrorString** (ErrorString &&other) noexcept

- ErrorString & **operator=** (const ErrorString &other)
- ErrorString & **operator=** (ErrorString &&other) noexcept
- const QString & **base** () const noexcept
- QString & **base** ()
- const QStringList & **additionalBases** () const noexcept
- QStringList & **additionalBases** ()
- const QString & **details** () const noexcept
- QString & **details** ()
- void **setBase** (QString error)
- void **setBase** (const char ∗error)
- void **appendBase** (const QString &error)
- void **appendBase** (const QStringList &errors)
- void **appendBase** (const char ∗error)
- void **setDetails** (const QString &error)
- void **setDetails** (const char ∗error)
- bool **isEmpty** () const
- void **clear** ()
- QString **localizedString** () const
- QString **nonLocalizedString** () const
- QTextStream & print (QTextStream &strm) const override

5.8.1 Detailed Description

The ErrorString class encapsulates two (or more) strings which are meant to contain translatable (base) and non-translatable (details) parts of the error description.

1. base() methods return const and non-const links to the primary translatable string
2. details() methods ruturn const and non-const links to non-translatable string (coming from some third party library etc)
3. additionalBases() methods return const and non-const links to additional translatable strings; one translatable string is not always enough because the error message might be composed from different parts

5.8.2 Member Function Documentation

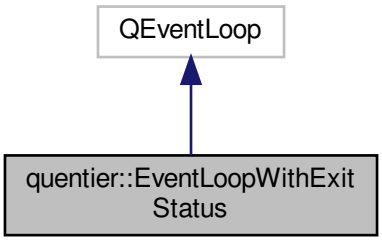
5.8.2.1 print()

```
QTextStream & quentier::ErrorString::print (
            QTextStream & strm ) const  [override], [virtual]
```

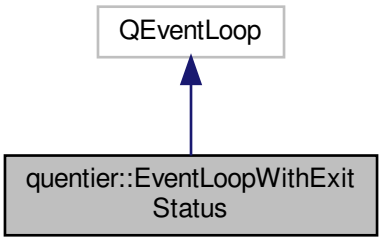
Implements quentier::Printable.

5.9 quentier::EventLoopWithExitStatus Class Reference

Inheritance diagram for quentier::EventLoopWithExitStatus:



Collaboration diagram for quentier::EventLoopWithExitStatus:



Public Types

- enum class **ExitStatus** { **Success** , **Failure** , **Timeout** }

Public Slots

- void **exitAsSuccess** ()
- void **exitAsFailure** ()
- void **exitAsFailureWithError** (QString errorDescription)
- void **exitAsFailureWithErrorString** (ErrorString errorDescription)
- void **exitAsTimeout** ()

Public Member Functions

- **EventLoopWithExitStatus** (QObject ∗parent=nullptr)
- ExitStatus **exitStatus** () const
- const ErrorString & **errorDescription** () const

Friends

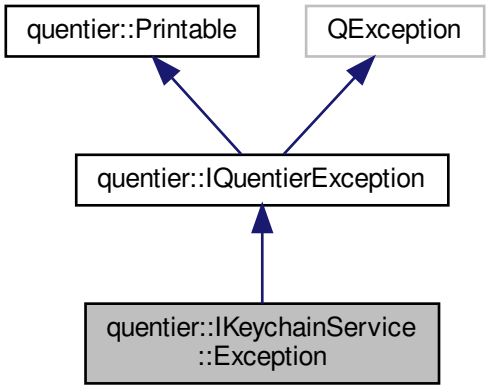
- QDebug & **operator**<< (QDebug &dbg, ExitStatus status)
- QTextStream & **operator**<< (QTextStream &strm, ExitStatus status)

5.10 quentier::IKeychainService::Exception Class Reference

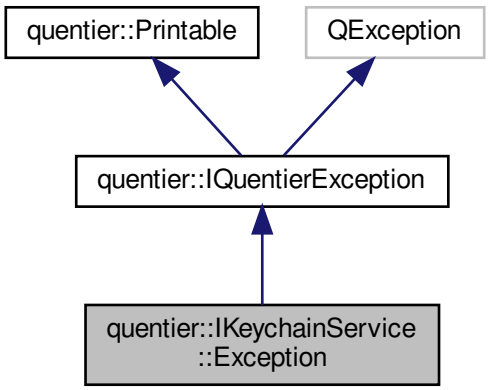
The IKeychainService::Exception class is the base class for exceptions returned inside QFutures from methods of IKeychainService.

`#include <IKeychainService.h>`

Inheritance diagram for quentier::IKeychainService::Exception:



Collaboration diagram for quentier::IKeychainService::Exception:



Public Member Functions

- **Exception** (ErrorCode errorCode) noexcept
- **Exception** (ErrorCode errorCode, ErrorString errorDescription) noexcept
- ErrorCode **errorCode** () const noexcept
- QString exceptionDisplayName () const override
- void **raise** () const override
- Exception * **clone** () const override

Additional Inherited Members

5.10.1 Detailed Description

The IKeychainService::Exception class is the base class for exceptions returned inside QFutures from methods of IKeychainService.

5.10.2 Member Function Documentation

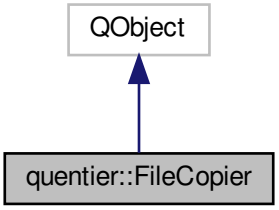
5.10.2.1 exceptionDisplayName()

```
QString quentier::IKeychainService::Exception::exceptionDisplayName ( ) const  [override],
[virtual]
```

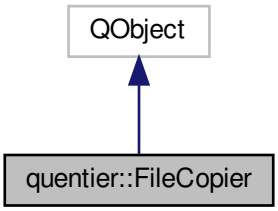
Implements quentier::IQuentierException.

5.11 quentier::FileCopier Class Reference

Inheritance diagram for quentier::FileCopier:



Collaboration diagram for quentier::FileCopier:



Public Types

- enum class **State** { **Idle** = 0 , **Copying** , **Cancelling** }

Public Slots

- void **copyFile** (QString sourcePath, QString destPath)
- void **cancel** ()

Signals

- void **progressUpdate** (double progress)
- void **finished** (QString sourcePath, QString destPath)
- void **cancelled** (QString sourcePath, QString destPath)
- void **notifyError** (ErrorString error)

Public Member Functions

- **FileCopier** (QObject *parent=nullptr)
- State **state** () const
- QString **sourceFilePath** () const
- QString **destinationFilePath** () const
- double **currentProgress** () const

Friends

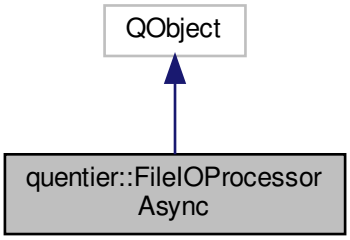
- QDebug & **operator<<** (QDebug &dbg, State state)
- QTextStream & **operator<<** (QTextStream &strm, State state)

5.12 quentier::FileIOProcessorAsync Class Reference

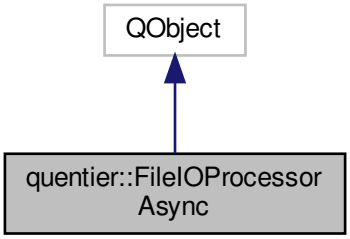
The FileIOProcessorAsync class is a wrapper under simple file IO operations, it is meant to be used for simple asynchronous IO.

`#include <FileIOProcessorAsync.h>`

Inheritance diagram for quentier::FileIOProcessorAsync:



Collaboration diagram for quentier::FileIOProcessorAsync:



Public Slots

- void onWriteFileRequest (QString absoluteFilePath, QByteArray data, QUuid requestId, bool append)

  onWriteFileRequest slot processes file write requests with given request ids
- void onReadFileRequest (QString absoluteFilePath, QUuid requestId)

  onReadFileRequest slot processes file read requests with given request ids

Signals

- void **readyForIO** ()

  readyForIO signal is emitted when the queue for file IO is empty for some time (30 seconds by default, can also be configured via setIdleTimePeriod method) after the last IO event to signal listeners that they can perform some IO via the FileIOProcessorAsync
- void writeFileRequestProcessed (bool success, ErrorString errorDescription, QUuid requestId)

  writeFileRequestProcessed signal is emitted when the file write request with given id is finished
- void readFileRequestProcessed (bool success, ErrorString errorDescription, QByteArray data, QUuid requestId)

  readFileRequestProcessed signal is emitted when the file read request with given id is finished

Public Member Functions

- **FileIOProcessorAsync** (QObject *parent=nullptr)
- void setIdleTimePeriod (qint32 seconds)

  setIdleTimePeriod sets time period defining the idle state of FileIOProcessorAsync: once the time measured since the last IO operation is over the specified number of seconds, the class emits readyForIO signal to any interested listeners of this event. If this method is not called ever, the default idle time period would be 30 seconds.

5.12.1 Detailed Description

The FileIOProcessorAsync class is a wrapper under simple file IO operations, it is meant to be used for simple asynchronous IO.

5.12.2 Member Function Documentation

5.12.2.1 onReadFileRequest

```
void quentier::FileIOProcessorAsync::onReadFileRequest (
            QString absoluteFilePath,
            QUuid requestId )  [slot]
```

onReadFileRequest slot processes file read requests with given request ids

Parameters

absoluteFilePath	Absolute file path to be read
requestId	Unique identifier of the file read request

5.12.2.2 onWriteFileRequest

```
void quentier::FileIOProcessorAsync::onWriteFileRequest (
            QString absoluteFilePath,
```

```
            QByteArray data,
            QUuid requestId,
            bool append )  [slot]
```

onWriteFileRequest slot processes file write requests with given request ids

Parameters

absoluteFilePath	Absolute file path to be written
data	Data to be written to the file
requestId	Unique identifier of the file write request
append	If true, the data would be appended to file, otherwise the entire file would be erased before with the data is written

5.12.2.3 readFileRequestProcessed

```
void quentier::FileIOProcessorAsync::readFileRequestProcessed (
            bool success,
            ErrorString errorDescription,
            QByteArray data,
            QUuid requestId )  [signal]
```

readFileRequestProcessed signal is emitted when the file read request with given id is finished

Parameters

success	True if read operation was successful, false otherwise
errorDescription	Textual description of the error
data	Data read from file
requestId	Unique identifier of the file read request

5.12.2.4 setIdleTimePeriod()

```
void quentier::FileIOProcessorAsync::setIdleTimePeriod (
            qint32 seconds )
```

setIdleTimePeriod sets time period defining the idle state of FileIOProcessorAsync: once the time measured since the last IO operation is over the specified number of seconds, the class emits readyForIO signal to any interested listeners of this event. If this method is not called ever, the default idle time period would be 30 seconds.

Parameters

seconds	Number of seconds for idle time period

5.12.2.5 writeFileRequestProcessed

```
void quentier::FileIOProcessorAsync::writeFileRequestProcessed (
            bool success,
            ErrorString errorDescription,
            QUuid requestId ) [signal]
```

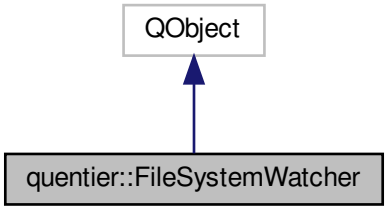
writeFileRequestProcessed signal is emitted when the file write request with given id is finished

Parameters

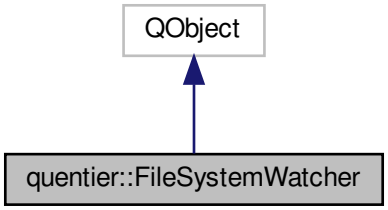
success	True if write operation was successful, false otherwise
errorDescription	Textual description of the error
requestId	Unique identifier of the file write request

5.13 quentier::FileSystemWatcher Class Reference

Inheritance diagram for quentier::FileSystemWatcher:



Collaboration diagram for quentier::FileSystemWatcher:



Signals

- void **directoryChanged** (const QString &path)
- void **directoryRemoved** (const QString &path)
- void **fileChanged** (const QString &path)
- void **fileRemoved** (const QString &path)

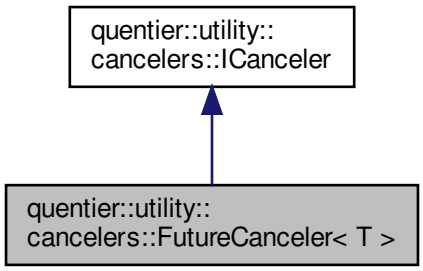
Public Member Functions

- **FileSystemWatcher** (int removalTimeoutMSec=FILE_SYSTEM_WATCHER_DEFAULT_REMOVAL_↩ TIMEOUT_MSEC, QObject ∗parent=nullptr)
- **FileSystemWatcher** (const QStringList &paths, int removalTimeoutMSec=FILE_SYSTEM_WATCHER_↩ DEFAULT_REMOVAL_TIMEOUT_MSEC, QObject ∗parent=nullptr)
- void **addPath** (const QString &path)
- void **addPaths** (const QStringList &paths)
- QStringList **directories** () const
- QStringList **files** () const
- void **removePath** (const QString &path)
- void **removePaths** (const QStringList &paths)

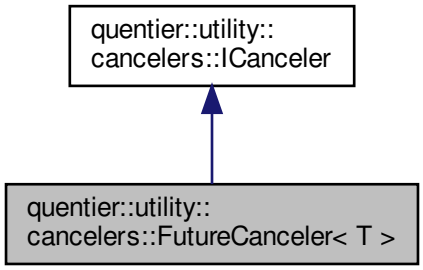
5.14 quentier::utility::cancelers::FutureCanceler< T > Class Template Reference

`#include <FutureCanceler.h>`

Inheritance diagram for quentier::utility::cancelers::FutureCanceler< T >:



Collaboration diagram for quentier::utility::cancelers::FutureCanceler< T >:



Public Member Functions

- **FutureCanceler** (QFuture< T > future)
- bool isCanceled () const noexcept override

5.14.1 Detailed Description

template<class T>
class quentier::utility::cancelers::FutureCanceler< T >

ICanceler implementation which tracks the canceled status of a future.

5.14.2 Member Function Documentation

5.14.2.1 isCanceled()

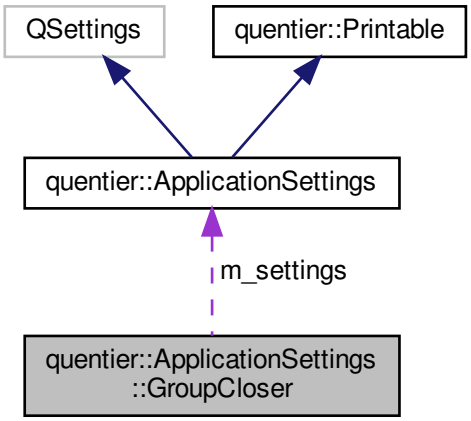
```
template<class T >
bool quentier::utility::cancelers::FutureCanceler< T >::isCanceled ( ) const  [inline], [override],
[virtual], [noexcept]
```

Implements quentier::utility::cancelers::ICanceler.

5.15 quentier::ApplicationSettings::GroupCloser Struct Reference

`#include <ApplicationSettings.h>`

Collaboration diagram for quentier::ApplicationSettings::GroupCloser:



Public Member Functions

- **GroupCloser** (ApplicationSettings &settings)

Public Attributes

- ApplicationSettings & **m_settings**

5.15.1 Detailed Description

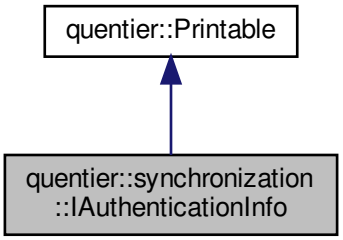
Helper struct for RAII style of ensuring the group once opened would be closed even if exception is thrown after beginning the group

5.16 quentier::synchronization::IAuthenticationInfo Class Reference

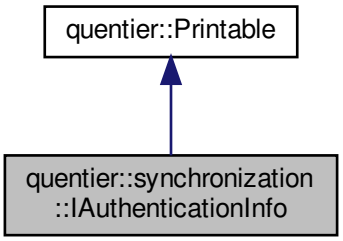
The IAuthenticationInfo interface represents the information obtained through OAuth and necessary to access Evernote API.

`#include <IAuthenticationInfo.h>`

Inheritance diagram for quentier::synchronization::IAuthenticationInfo:



Collaboration diagram for quentier::synchronization::IAuthenticationInfo:



Public Member Functions

- virtual qevercloud::UserID userId () const =0
- virtual QString authToken () const =0
- virtual qevercloud::Timestamp authTokenExpirationTime () const =0
- virtual qevercloud::Timestamp authenticationTime () const =0
- virtual QString shardId () const =0
- virtual QString noteStoreUrl () const =0
- virtual QString webApiUrlPrefix () const =0
- virtual QList< QNetworkCookie > userStoreCookies () const =0

5.16.1 Detailed Description

The IAuthenticationInfo interface represents the information obtained through OAuth and necessary to access Evernote API.

5.16.2 Member Function Documentation

5.16.2.1 authenticationTime()

```
virtual qevercloud::Timestamp quentier::synchronization::IAuthenticationInfo::authentication↩
Time ( ) const  [pure virtual]
```

Timestamp at which authentication info was received from Evernote

5.16.2.2 authToken()

```
virtual QString quentier::synchronization::IAuthenticationInfo::authToken ( ) const  [pure
virtual]
```

Authentication token which needs to be used for access to Evernote API

5.16.2.3 authTokenExpirationTime()

```
virtual qevercloud::Timestamp quentier::synchronization::IAuthenticationInfo::authToken↩
ExpirationTime ( ) const  [pure virtual]
```

Expiration timestamp for the authentication token

5.16.2.4 noteStoreUrl()

```
virtual QString quentier::synchronization::IAuthenticationInfo::noteStoreUrl ( ) const  [pure
virtual]
```

Url of the note store service for this user

5.16.2.5 shardId()

```
virtual QString quentier::synchronization::IAuthenticationInfo::shardId ( ) const  [pure virtual]
```

Shard identifier which needs to be used for access to Evernote API along with the authentication token

5.16.2.6 userId()

```
virtual qevercloud::UserID quentier::synchronization::IAuthenticationInfo::userId ( ) const
[pure virtual]
```

Identifier of the authenticated user

5.16.2.7 userStoreCookies()

```
virtual QList< QNetworkCookie > quentier::synchronization::IAuthenticationInfo::userStore↩
Cookies ( ) const  [pure virtual]
```

The list of network cookies received during OAuth procedure. Although is is not mentioned anywhere in Evernote docs, these cookies might have to be used for access to user store. See this discussion for reference: https↩
://discussion.evernote.com/forums/topic/124257-calls-to-userstore-from-evernote-api-stop

5.16.2.8 webApiUrlPrefix()

```
virtual QString quentier::synchronization::IAuthenticationInfo::webApiUrlPrefix ( ) const
[pure virtual]
```

Url prefix for Evernote Web API.

See also

qevercloud::PublicUserInfo::webApiUrlPrefix

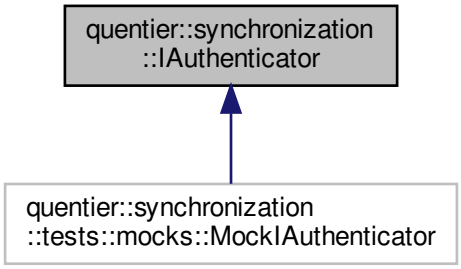
5.17 quentier::synchronization::IAuthenticationInfoBuilder Class Reference

Public Member Functions

- virtual IAuthenticationInfoBuilder & **setUserId** (qevercloud::UserID userId)=0
- virtual IAuthenticationInfoBuilder & **setAuthToken** (QString token)=0
- virtual IAuthenticationInfoBuilder & **setAuthTokenExpirationTime** (qevercloud::Timestamp expiration↩
Time)=0
- virtual IAuthenticationInfoBuilder & **setAuthenticationTime** (qevercloud::Timestamp authenticationTime)=0
- virtual IAuthenticationInfoBuilder & **setShardId** (QString shardId)=0
- virtual IAuthenticationInfoBuilder & **setNoteStoreUrl** (QString noteStoreUrl)=0
- virtual IAuthenticationInfoBuilder & **setWebApiUrlPrefix** (QString webApiUrlPrefix)=0
- virtual IAuthenticationInfoBuilder & **setUserStoreCookies** (QList< QNetworkCookie > cookies)=0
- virtual IAuthenticationInfoPtr **build** ()=0

5.18 quentier::synchronization::IAuthenticator Class Reference

Inheritance diagram for quentier::synchronization::IAuthenticator:



Public Member Functions

- virtual QFuture< IAuthenticationInfoPtr > **authenticateNewAccount** ()=0
- virtual QFuture< IAuthenticationInfoPtr > **authenticateAccount** (Account account)=0

5.19 quentier::ResourceRecognitionIndexItem::IBarcodeItem Struct Reference

Public Member Functions

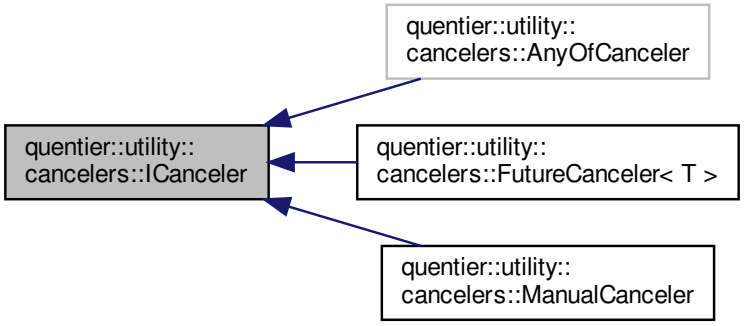
- virtual QString **barcode** () const =0
- virtual int **weight** () const =0

5.20 quentier::utility::cancelers::ICanceler Class Reference

The ICanceler interface provides isCanceled method which can be used to check whether some processing can be skipped because it was canceled.

`#include <ICanceler.h>`

Inheritance diagram for quentier::utility::cancelers::ICanceler:



Public Member Functions

- virtual bool **isCanceled** () const =0

5.20.1 Detailed Description

The ICanceler interface provides isCanceled method which can be used to check whether some processing can be skipped because it was canceled.

5.21 quentier::enml::IConverter Class Reference

The IConverter interface encapsulates a set of methods performing conversions between ENML and other note content formats, namely HTML.

`#include <IConverter.h>`

Public Types

- enum class EnexExportTags { **Yes** = 0 , **No** }

  The EnexExportTags enum allows to specify whether export of note(s) to ENEX should include the names of note's tags or not.

Public Member Functions

- virtual Result< QString, ErrorString > convertHtmlToEnml (const QString &html, IDecryptedTextCache &decryptedTextCache, const QList< conversion_rules::ISkipRulePtr > &skipRules={}) const =0
- virtual Result< void, ErrorString > convertHtmlToDoc (const QString &html, QTextDocument &doc, const QList< conversion_rules::ISkipRulePtr > &skipRules={}) const =0
- virtual Result< QString, ErrorString > convertHtmlToXml (const QString &html) const =0
- virtual Result< QString, ErrorString > convertHtmlToXhtml (const QString &html) const =0
- virtual Result< IHtmlDataPtr, ErrorString > convertEnmlToHtml (const QString &enml, IDecryptedTextCache &decryptedTextCache) const =0
- virtual Result< QString, ErrorString > convertEnmlToPlainText (const QString &enml) const =0
- virtual Result< QStringList, ErrorString > convertEnmlToWordsList (const QString &enml) const =0
- virtual QStringList convertPlainTextToWordsList (const QString &plainText) const =0
- virtual Result< void, ErrorString > validateEnml (const QString &enml) const =0
- virtual Result< QString, ErrorString > validateAndFixupEnml (const QString &enml) const =0
- virtual Result< QString, ErrorString > exportNotesToEnex (const QList< qevercloud::Note > &notes, const QHash< QString, QString > &tagNamesByTagLocalIds, EnexExportTags exportTagsOption, const QString &version={}) const =0
- virtual Result< QList< qevercloud::Note >, ErrorString > importEnex (const QString &enex) const =0

5.21.1 Detailed Description

The IConverter interface encapsulates a set of methods performing conversions between ENML and other note content formats, namely HTML.

5.21.2 Member Function Documentation

5.21.2.1 convertEnmlToHtml()

```
virtual Result< IHtmlDataPtr, ErrorString > quentier::enml::IConverter::convertEnmlToHtml (
            const QString & enml,
            IDecryptedTextCache & decryptedTextCache ) const  [pure virtual]
```

Converts ENML into HTML representation of note content

Parameters

enml	ENML representation of note content
decryptedTextCache	cache of decrypted text fragments

Returns

Result with HTML data in case of success or error string in case of failure

5.21.2.2 convertEnmlToPlainText()

```
virtual Result< QString, ErrorString > quentier::enml::IConverter::convertEnmlToPlainText (
            const QString & enml ) const  [pure virtual]
```

Converts ENML into plain text representation of note content

Parameters

enml	ENML representation of note content

Returns

Result with plain text representation of note content in case of success or error string in case of failure

5.21.2.3 convertEnmlToWordsList()

```
virtual Result< QStringList, ErrorString > quentier::enml::IConverter::convertEnmlToWordsList
(
            const QString & enml ) const  [pure virtual]
```

Converts ENML into a list of words

Parameters

enml	ENML representation of note content

Returns

Result with list of words in case of success or error string in case of failure

5.21.2.4 convertHtmlToDoc()

```
virtual Result< void, ErrorString > quentier::enml::IConverter::convertHtmlToDoc (
            const QString & html,
            QTextDocument & doc,
            const QList< conversion_rules::ISkipRulePtr > & skipRules = {} ) const  [pure
virtual]
```

Convert HTML representation of note content into QTextDocument

Parameters

html	HTML representation of note content
doc	QTextDocument into which the converted note content is put
skipRules	skip rules to be used during the conversion

Returns

Valid result in case of success or error string in case of failure

5.21.2.5 convertHtmlToEnml()

```
virtual Result< QString, ErrorString > quentier::enml::IConverter::convertHtmlToEnml (
            const QString & html,
            IDecryptedTextCache & decryptedTextCache,
            const QList< conversion_rules::ISkipRulePtr > & skipRules = {} ) const  [pure
virtual]
```

Converts HTML representation of note content into ENML

Parameters

html	HTML representation of note content
decryptedTextCache	cache of decrypted text fragments
skipRules	skip rules to be used during the conversion

Returns

Result with ENML in case of success or error string in case of failure

5.21.2.6 convertHtmlToXhtml()

```
virtual Result< QString, ErrorString > quentier::enml::IConverter::convertHtmlToXhtml (
            const QString & html ) const  [pure virtual]
```

Convert HTML representation of note content into a valid XHTML document

Parameters

html	HTML representation of note content

Returns

Result with XHTML in case of success of error string in case of failure

5.21.2.7 convertHtmlToXml()

```
virtual Result< QString, ErrorString > quentier::enml::IConverter::convertHtmlToXml (
            const QString & html ) const  [pure virtual]
```

Convert HTML representation of note content into a valid XML document

Parameters

html	HTML representation of note content

Returns

Result with XML in case of success of error string in case of failure

5.21.2.8 convertPlainTextToWordsList()

```
virtual QStringList quentier::enml::IConverter::convertPlainTextToWordsList (
            const QString & plainText ) const  [pure virtual]
```

Converts plain text into a list of words

Parameters

plainText	plain text representation of note content

Returns

list of words

5.21.2.9 exportNotesToEnex()

```
virtual Result< QString, ErrorString > quentier::enml::IConverter::exportNotesToEnex (
            const QList< qevercloud::Note > & notes,
```

```
            const QHash< QString, QString > & tagNamesByTagLocalIds,
            EnexExportTags exportTagsOption,
            const QString & version = {} ) const  [pure virtual]
```

Exports a list of notes into ENEX

Parameters

notes	notes to be exported into ENEX
tagNamesByTagLocalIds	mapper from tag local ids into tag names
exportTagsOption	option controlling the export of tag names
version	optional version tag for ENEX, omitted if not set

Returns

Result with ENEX in case of success or error string in case of failure

5.21.2.10 importEnex()

```
virtual Result< QList< qevercloud::Note >, ErrorString > quentier::enml::IConverter::import↩
Enex (
            const QString & enex ) const  [pure virtual]
```

Import notes from ENEX

Parameters

enex	ENEX to be used for import

Returns

Result with list of notes in case of success or error string in case of failure

Note

if tag names are present in ENEX, corresponding notes would have their tagNames field filled

5.21.2.11 validateAndFixupEnml()

```
virtual Result< QString, ErrorString > quentier::enml::IConverter::validateAndFixupEnml (
            const QString & enml ) const  [pure virtual]
```

Validates ENML and attempts to fix it automatically if it's not valid

Parameters

enml	ENML representation of note content

Returns

Result with either unchanged or fixed up ENML in case of success or error string in case of failure

5.21.2.12 validateEnml()

```
virtual Result< void, ErrorString > quentier::enml::IConverter::validateEnml (
            const QString & enml ) const  [pure virtual]
```

Validates ENML against rules

Parameters

enml	ENML representation of note content

Returns

valid Result in case of success or error string in case of failure

5.22 quentier::enml::IDecryptedTextCache Class Reference

Public Types

- enum class **RememberForSession** { **Yes** , **No** }

Public Member Functions

- virtual void **addDecryptexTextInfo** (const QString &encryptedText, const QString &decryptedText, const QString &passphrase, const QString &cipher, std::size_t keyLength, RememberForSession rememberFor↩Session)=0
- virtual std::optional< std::pair< QString, RememberForSession > > **findDecryptedTextInfo** (const QString &encryptedText) const =0
- virtual std::optional< QString > **updateDecryptedTextInfo** (const QString &originalEncryptedText, const QString &newDecryptedText)=0
- virtual void **removeDecryptedTextInfo** (const QString &encryptedText)=0
- virtual void **clearNonRememberedForSessionEntries** ()=0

Friends

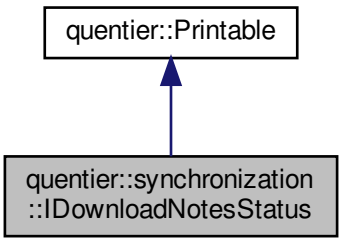
- QUENTIER_EXPORT QDebug & **operator<<** (QDebug &dbg, RememberForSession rememberFor↩Session)
- QUENTIER_EXPORT QTextStream & **operator<<** (QTextStream &strm, RememberForSession rememberForSession)

5.23 quentier::synchronization::IDownloadNotesStatus Class Reference

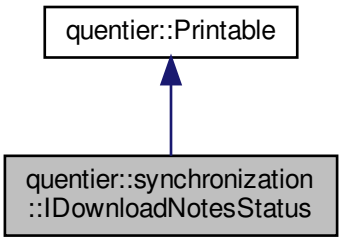
The IDownloadNotesStatus interface presents information about the status of notes downloading process.

`#include <IDownloadNotesStatus.h>`

Inheritance diagram for quentier::synchronization::IDownloadNotesStatus:



Collaboration diagram for quentier::synchronization::IDownloadNotesStatus:



Public Types

- using **QExceptionPtr** = std::shared_ptr< QException >
- using **NoteWithException** = std::pair< qevercloud::Note, QExceptionPtr >
- using **GuidWithException** = std::pair< qevercloud::Guid, QExceptionPtr >
- using **UpdateSequenceNumbersByGuid** = QHash< qevercloud::Guid, qint32 >

Public Member Functions

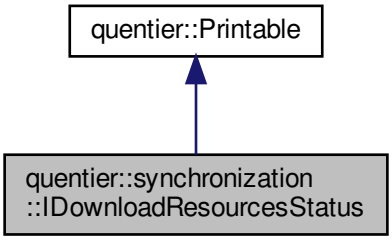
- virtual quint64 **totalNewNotes** () const =0
- virtual quint64 **totalUpdatedNotes** () const =0
- virtual quint64 **totalExpungedNotes** () const =0
- virtual QList< NoteWithException > **notesWhichFailedToDownload** () const =0
- virtual QList< NoteWithException > **notesWhichFailedToProcess** () const =0
- virtual QList< GuidWithException > **noteGuidsWhichFailedToExpunge** () const =0
- virtual UpdateSequenceNumbersByGuid **processedNoteGuidsAndUsns** () const =0
- virtual UpdateSequenceNumbersByGuid **cancelledNoteGuidsAndUsns** () const =0
- virtual QList< qevercloud::Guid > **expungedNoteGuids** () const =0
- virtual StopSynchronizationError **stopSynchronizationError** () const =0

5.23.1 Detailed Description

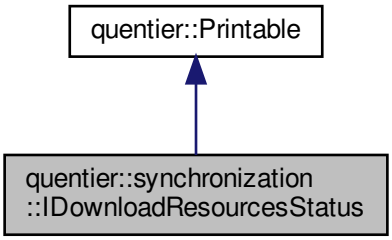
The IDownloadNotesStatus interface presents information about the status of notes downloading process.

5.24 quentier::synchronization::IDownloadResourcesStatus Class Reference

Inheritance diagram for quentier::synchronization::IDownloadResourcesStatus:



Collaboration diagram for quentier::synchronization::IDownloadResourcesStatus:



Public Types

- using **QExceptionPtr** = std::shared_ptr< QException >
- using **ResourceWithException** = std::pair< qevercloud::Resource, QExceptionPtr >
- using **UpdateSequenceNumbersByGuid** = QHash< qevercloud::Guid, qint32 >

Public Member Functions

- virtual quint64 **totalNewResources** () const =0
- virtual quint64 **totalUpdatedResources** () const =0
- virtual QList< ResourceWithException > **resourcesWhichFailedToDownload** () const =0
- virtual QList< ResourceWithException > **resourcesWhichFailedToProcess** () const =0
- virtual UpdateSequenceNumbersByGuid **processedResourceGuidsAndUsns** () const =0
- virtual UpdateSequenceNumbersByGuid **cancelledResourceGuidsAndUsns** () const =0
- virtual StopSynchronizationError **stopSynchronizationError** () const =0

5.25 quentier::enml::IENMLTagsConverter Class Reference

The IENMLTagsConverter interfaces provides methods which convert Evernote-specific markup tags such as en-crypt, en-media etc. into their counterparts which should be used in the HTML representation of note content.

`#include <IENMLTagsConverter.h>`

Public Member Functions

- virtual QString convertEnToDo (bool checked, quint32 index) const =0
- virtual QString convertEncryptedText (const QString &encryptedText, const QString &hint, const QString &cipher, std::size_t keyLength, quint32 index) const =0
- virtual QString convertDecryptedText (const QString &decryptedText, const QString &encryptedText, const QString &hint, const QString &cipher, std::size_t keyLength, quint32 index) const =0
- virtual Result< QString, ErrorString > convertResource (const qevercloud::Resource &resource) const =0

5.25.1 Detailed Description

The IENMLTagsConverter interfaces provides methods which convert Evernote-specific markup tags such as en-crypt, en-media etc. into their counterparts which should be used in the HTML representation of note content.

5.25.2 Member Function Documentation

5.25.2.1 convertDecryptedText()

```
virtual QString quentier::enml::IENMLTagsConverter::convertDecryptedText (
            const QString & decryptedText,
            const QString & encryptedText,
            const QString & hint,
            const QString & cipher,
            std::size_t keyLength,
            quint32 index ) const  [pure virtual]
```

Converts already decrypted en-crypt tag into its HTML counterpart

Parameters

decryptedText	decrypted text from en-crypt tag
encryptedText	encrypted text contained within en-crypt tag
hint	hint to be displayed when user tries to decrypt the text
cipher	cipher used to ecrypt the text
keyLength	length of the key used to encrypt the text
index	index of particular en-crypt tag within the note content so that different en-crypt tags can be differentiated

Returns

HTML representation of decrypted en-crypt tag

5.25.2.2 convertEncryptedText()

```
virtual QString quentier::enml::IENMLTagsConverter::convertEncryptedText (
            const QString & encryptedText,
            const QString & hint,
            const QString & cipher,
            std::size_t keyLength,
            quint32 index ) const  [pure virtual]
```

Converts en-crypt tag into its HTML counterpart

Parameters

encryptedText	encrypted text contained within en-crypt tag
hint	hint to be displayed when user tries to decrypt the text
cipher	cipher used to ecrypt the text
keyLength	length of the key used to encrypt the text
index	index of particular en-crypt tag within the note content so that different en-crypt tags can be differentiated

Returns

HTML representation of en-crypt tag

5.25.2.3 convertEnToDo()

```
virtual QString quentier::enml::IENMLTagsConverter::convertEnToDo (
            bool checked,
            quint32 index ) const  [pure virtual]
```

Converts en-todo tag into its HTML counterpart

Parameters

checked	indicates whether this todo is checked or not
index	index of particular en-todo tag within the note content so that different todo tags can be differentiated

Returns

HTML representation of en-todo tag

5.25.2.4 convertResource()

```
virtual Result< QString, ErrorString > quentier::enml::IENMLTagsConverter::convertResource (
            const qevercloud::Resource & resource ) const  [pure virtual]
```

Converts en-media tag representing a resource into its HTML counterpart

Parameters

resource	resource corresponding to en-media tag

Returns

Result with valid HTML representing the resource/en-media tag in case of success or error string in case of failure

5.26 quentier::synchronization::ISyncConflictResolver::ConflictResolution::IgnoreMine Struct Reference

The IgnoreMine conflict resolution means "use theirs version and ignore mine version as it doesn't really conflict with theirs version".

`#include <ISyncConflictResolver.h>`

5.26.1 Detailed Description

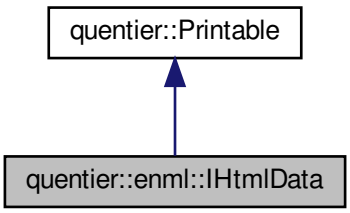
The IgnoreMine conflict resolution means "use theirs version and ignore mine version as it doesn't really conflict with theirs version".

5.27 quentier::enml::IHtmlData Struct Reference

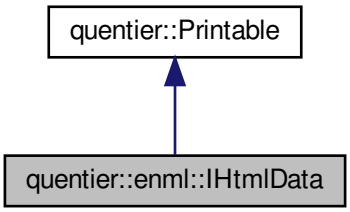
The IHtmlData represents the result of ENML to HTML conversion: HTML itself plus some metadata.

`#include <IHtmlData.h>`

Inheritance diagram for quentier::enml::IHtmlData:



Collaboration diagram for quentier::enml::IHtmlData:



Public Member Functions

- virtual QString html () const =0
- virtual quint32 numEnToDoNodes () const =0
- virtual quint32 numHyperlinkNodes () const =0
- virtual quint32 numEnCryptNodes () const =0
- virtual quint32 numEnDecryptedNodes () const =0
- QTextStream & print (QTextStream &strm) const override

5.27.1 Detailed Description

The IHtmlData represents the result of ENML to HTML conversion: HTML itself plus some metadata.

5.27.2 Member Function Documentation

5.27.2.1 html()

```
virtual QString quentier::enml::IHtmlData::html ( ) const  [pure virtual]
```

HTML representation of note content

5.27.2.2 numEnCryptNodes()

```
virtual quint32 quentier::enml::IHtmlData::numEnCryptNodes ( ) const  [pure virtual]
```

Number of en-crypt nodes within note HTML

5.27.2.3 numEnDecryptedNodes()

```
virtual quint32 quentier::enml::IHtmlData::numEnDecryptedNodes ( ) const  [pure virtual]
```

Number of decrypted en-crypt nodes within note HTML

5.27.2.4 numEnToDoNodes()

```
virtual quint32 quentier::enml::IHtmlData::numEnToDoNodes ( ) const  [pure virtual]
```

Number of ToDo nodes within note HTML

5.27.2.5 numHyperlinkNodes()

```
virtual quint32 quentier::enml::IHtmlData::numHyperlinkNodes ( ) const  [pure virtual]
```

Number of hyperlink nodes within note HTML

5.27.2.6 print()

```
QTextStream & quentier::enml::IHtmlData::print (
            QTextStream & strm ) const  [override], [virtual]
```

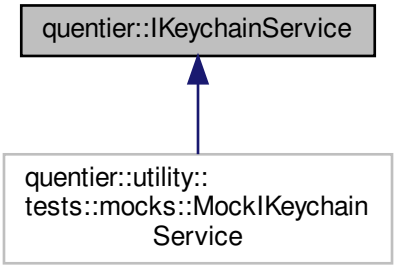
Implements quentier::Printable.

5.28 quentier::IKeychainService Class Reference

The IKeychainService interface provides the ability to interact with the storage of sensitive data - read, write and delete it.

`#include <IKeychainService.h>`

Inheritance diagram for quentier::IKeychainService:



Classes

- class Exception

  The IKeychainService::Exception class is the base class for exceptions returned inside QFutures from methods of IKeychainService.

Public Types

- enum class ErrorCode {
  NoError , EntryNotFound , CouldNotDeleteEntry , AccessDeniedByUser ,
  AccessDenied , NoBackendAvailable , NotImplemented , OtherError }

Public Member Functions

- virtual QFuture< void > writePassword (QString service, QString key, QString password)=0
- virtual QFuture< QString > readPassword (QString service, QString key) const =0
- virtual QFuture< void > deletePassword (QString service, QString key)=0

Friends

- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ErrorCode errorCode)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ErrorCode errorCode)

5.28.1 Detailed Description

The IKeychainService interface provides the ability to interact with the storage of sensitive data - read, write and delete it.

5.28.2 Member Enumeration Documentation

5.28.2.1 ErrorCode

```
enum class quentier::IKeychainService::ErrorCode  [strong]
```

Error codes for results of operations with the keychain service

Enumerator

NoError	No error occurred, operation was successful
EntryNotFound	For the given key no data was found
CouldNotDeleteEntry	Could not delete existing secret data
AccessDeniedByUser	User denied access to keychain
AccessDenied	Access denied for some reason
NoBackendAvailable	No platform-specific keychain service available
NotImplemented	Not implemented on platform
OtherError	Something else went wrong, the error description specifies what

5.28.3 Member Function Documentation

5.28.3.1 deletePassword()

```
virtual QFuture< void > quentier::IKeychainService::deletePassword (
            QString service,
            QString key )  [pure virtual]
```

deletePassword potentially asynchronously deletes password from the keychain.

Parameters

service	Name of service within the keychain
key	Key under which the password is stored

Returns

Future which becomes finished when the operation is comlete. If the operation fails, the future would contain an exception.

5.28.3.2 readPassword()

```
virtual QFuture< QString > quentier::IKeychainService::readPassword (
            QString service,
            QString key ) const  [pure virtual]
```

readPassword method potentially asynchronously reads password from the keychain.

Parameters

service	Name of service within the keychain
key	Key under which the password is stored

Returns

Future which becomes finished when the operation is complete. The value inside the future would be the read password. If the operation fails, the future would contain an exception.

5.28.3.3 writePassword()

```
virtual QFuture< void > quentier::IKeychainService::writePassword (
            QString service,
            QString key,
            QString password )  [pure virtual]
```

writePassword method potentially asynchronously writes password to the keychain.

Parameters

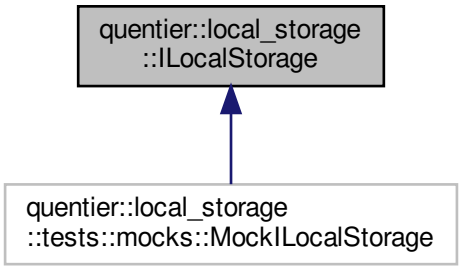
service	Name of service within the keychain
key	Key to store the password under
password	Password to store in the keychain

Returns

Future which becomes finished when the operation is comlete. If the operation fails, the future would contain an exception.

5.29 quentier::local_storage::ILocalStorage Class Reference

Inheritance diagram for quentier::local_storage::ILocalStorage:



Classes

- struct ListGuidsFilters
- struct ListLinkedNotebooksOptions
- struct ListNotebooksOptions
- struct ListNotesOptions
- struct ListObjectsFilters
- struct ListOptionsBase
- struct ListSavedSearchesOptions
- struct ListTagsOptions

Public Types

- enum class **StartupOption** { **ClearDatabase** = 1 << 1 , **OverrideLock** = 1 << 2 }
- enum class **ListObjectsFilter** { **Include** , **Exclude** }
- enum class **OrderDirection** { **Ascending** , **Descending** }
- enum class **ListNotebooksOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByNotebookName** , **ByCreationTimestamp** , **ByModificationTimestamp** }
- enum class **ListLinkedNotebooksOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByShareName** , **By↩ Username** }
- enum class **ListTagsOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByName** }
- enum class **ListNotesOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByTitle** , **ByCreationTimestamp** , **ByModificationTimestamp** , **ByDeletionTimestamp** , **ByAuthor** , **BySource** , **BySourceApplication** , **ByReminderTime** , **ByPlaceName** }
- enum class **ListSavedSearchesOrder** { **NoOrder** , **ByUpdateSequenceNumber** , **ByName** , **ByFormat** }

- enum class Affiliation { **Any** , **User** , **AnyLinkedNotebook** , **ParticularLinkedNotebooks** }
- enum class TagNotesRelation { Any , WithNotes , WithoutNotes }
- enum class **NoteCountOption** { **IncludeNonDeletedNotes** = 1 << 1 , **IncludeDeletedNotes** = 1 << 2 }
- enum class **UpdateNoteOption** { **UpdateResourceMetadata** = 1 << 1 , **UpdateResourceBinaryData** = 1 << 2 , **UpdateTags** = 1 << 3 }
- enum class **FetchNoteOption** { **WithResourceMetadata** = 1 << 1 , **WithResourceBinaryData** = 1 << 2 }
- enum class **FetchResourceOption** { **WithBinaryData** = 1 << 1 }
- enum class **HighestUsnOption** { **WithinUserOwnContent** , **WithinUserOwnContentAndLinked↩ Notebooks** }

Public Member Functions

- **Q_DECLARE_FLAGS** (StartupOptions, StartupOption)
- virtual QFuture< bool > **isVersionTooHigh** () const =0
- virtual QFuture< bool > **requiresUpgrade** () const =0
- virtual QFuture< QList< IPatchPtr > > **requiredPatches** () const =0
- virtual QFuture< qint32 > **version** () const =0
- virtual QFuture< qint32 > **highestSupportedVersion** () const =0
- virtual QFuture< quint32 > **userCount** () const =0
- virtual QFuture< void > **putUser** (qevercloud::User user)=0
- virtual QFuture< std::optional< qevercloud::User > > **findUserById** (qevercloud::UserID userId) const =0
- virtual QFuture< void > **expungeUserById** (qevercloud::UserID userId)=0
- virtual QFuture< quint32 > **notebookCount** () const =0
- virtual QFuture< void > **putNotebook** (qevercloud::Notebook notebook)=0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findNotebookByLocalId** (QString notebook↩ LocalId) const =0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findNotebookByGuid** (qevercloud::Guid guid) const =0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findNotebookByName** (QString notebook↩ Name, std::optional< qevercloud::Guid > linkedNotebookGuid=std::nullopt) const =0
- virtual QFuture< std::optional< qevercloud::Notebook > > **findDefaultNotebook** () const =0
- virtual QFuture< void > **expungeNotebookByLocalId** (QString notebookLocalId)=0
- virtual QFuture< void > **expungeNotebookByGuid** (qevercloud::Guid notebookGuid)=0
- virtual QFuture< void > **expungeNotebookByName** (QString name, std::optional< qevercloud::Guid > linkedNotebookGuid=std::nullopt)=0
- virtual QFuture< QList< qevercloud::Notebook > > **listNotebooks** (ListNotebooksOptions options={}) const =0
- virtual QFuture< QList< qevercloud::SharedNotebook > > **listSharedNotebooks** (qevercloud::Guid notebookGuid={}) const =0
- virtual QFuture< QSet< qevercloud::Guid > > **listNotebookGuids** (ListGuidsFilters filters, std::optional< qevercloud::Guid > linkedNotebookGuid={}) const =0
- virtual QFuture< quint32 > **linkedNotebookCount** () const =0
- virtual QFuture< void > **putLinkedNotebook** (qevercloud::LinkedNotebook linkedNotebook)=0
- virtual QFuture< std::optional< qevercloud::LinkedNotebook > > **findLinkedNotebookByGuid** (qevercloud::Guid guid) const =0
- virtual QFuture< void > **expungeLinkedNotebookByGuid** (qevercloud::Guid guid)=0
- virtual QFuture< QList< qevercloud::LinkedNotebook > > **listLinkedNotebooks** (ListLinkedNotebooksOptions options={}) const =0
- virtual QFuture< quint32 > **noteCount** (NoteCountOptions options=NoteCountOptions(NoteCountOption↩ ::IncludeNonDeletedNotes)) const =0
- virtual QFuture< quint32 > **noteCountPerNotebookLocalId** (QString notebookLocalId, NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)) const =0
- virtual QFuture< quint32 > **noteCountPerTagLocalId** (QString tagLocalId, NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)) const =0

- virtual QFuture< QHash< QString, quint32 > > **noteCountsPerTags** (ListTagsOptions listTagsOptions={}, NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNonDeletedNotes)) const =0
- virtual QFuture< quint32 > **noteCountPerNotebookAndTagLocalIds** (QStringList notebookLocalIds, QStringList tagLocalIds, NoteCountOptions options=NoteCountOptions(NoteCountOption::IncludeNon↩DeletedNotes)) const =0
- virtual QFuture< void > **putNote** (qevercloud::Note note)=0
- virtual QFuture< void > **updateNote** (qevercloud::Note note, UpdateNoteOptions options)=0
- virtual QFuture< std::optional< qevercloud::Note > > **findNoteByLocalId** (QString noteLocalId, Fetch↩NoteOptions options) const =0
- virtual QFuture< std::optional< qevercloud::Note > > **findNoteByGuid** (qevercloud::Guid noteGuid, FetchNoteOptions options) const =0
- virtual QFuture< QList< qevercloud::Note > > **listNotes** (FetchNoteOptions fetchOptions, ListNotesOptions listOptions={}) const =0
- virtual QFuture< QList< qevercloud::Note > > **listNotesPerNotebookLocalId** (QString notebookLocalId, FetchNoteOptions fetchOptions, ListNotesOptions listOptions={}) const =0
- virtual QFuture< QList< qevercloud::Note > > **listNotesPerTagLocalId** (QString tagLocalId, FetchNote↩Options fetchOptions, ListNotesOptions listOptions={}) const =0
- virtual QFuture< QList< qevercloud::Note > > **listNotesPerNotebookAndTagLocalIds** (QStringList notebookLocalIds, QStringList tagLocalIds, FetchNoteOptions fetchOptions, ListNotesOptions listOptions={}) const =0
- virtual QFuture< QList< qevercloud::Note > > **listNotesByLocalIds** (QStringList noteLocalIds, Fetch↩NoteOptions fetchOptions, ListNotesOptions listOptions={}) const =0
- virtual QFuture< QSet< qevercloud::Guid > > **listNoteGuids** (ListGuidsFilters filters, std::optional< qevercloud::Guid > linkedNotebookGuid={}) const =0
- virtual QFuture< QList< qevercloud::Note > > **queryNotes** (NoteSearchQuery query, FetchNoteOptions fetchOptions) const =0
- virtual QFuture< QStringList > **queryNoteLocalIds** (NoteSearchQuery query) const =0
- virtual QFuture< void > **expungeNoteByLocalId** (QString noteLocalId)=0
- virtual QFuture< void > **expungeNoteByGuid** (qevercloud::Guid noteGuid)=0
- virtual QFuture< quint32 > **tagCount** () const =0
- virtual QFuture< void > **putTag** (qevercloud::Tag tag)=0
- virtual QFuture< std::optional< qevercloud::Tag > > **findTagByLocalId** (QString tagLocalId) const =0
- virtual QFuture< std::optional< qevercloud::Tag > > **findTagByGuid** (qevercloud::Guid tagGuid) const =0
- virtual QFuture< std::optional< qevercloud::Tag > > **findTagByName** (QString tagName, std::optional< qevercloud::Guid > linkedNotebookGuid=std::nullopt) const =0
- virtual QFuture< QList< qevercloud::Tag > > **listTags** (ListTagsOptions options={}) const =0
- virtual QFuture< QList< qevercloud::Tag > > **listTagsPerNoteLocalId** (QString noteLocalId, ListTagsOptions options={}) const =0
- virtual QFuture< QSet< qevercloud::Guid > > **listTagGuids** (ListGuidsFilters filters, std::optional< qevercloud::Guid > linkedNotebookGuid={}) const =0
- virtual QFuture< void > **expungeTagByLocalId** (QString tagLocalId)=0
- virtual QFuture< void > **expungeTagByGuid** (qevercloud::Guid tagGuid)=0
- virtual QFuture< void > **expungeTagByName** (QString name, std::optional< qevercloud::Guid > linked↩NotebookGuid=std::nullopt)=0
- virtual QFuture< quint32 > **resourceCount** (NoteCountOptions options=NoteCountOptions(NoteCount↩Option::IncludeNonDeletedNotes)) const =0
- virtual QFuture< quint32 > **resourceCountPerNoteLocalId** (QString noteLocalId) const =0
- virtual QFuture< void > **putResource** (qevercloud::Resource resource)=0
- virtual QFuture< std::optional< qevercloud::Resource > > **findResourceByLocalId** (QString resource↩LocalId, FetchResourceOptions options={}) const =0
- virtual QFuture< std::optional< qevercloud::Resource > > **findResourceByGuid** (qevercloud::Guid resourceGuid, FetchResourceOptions options={}) const =0
- virtual QFuture< void > **expungeResourceByLocalId** (QString resourceLocalId)=0
- virtual QFuture< void > **expungeResourceByGuid** (qevercloud::Guid resourceGuid)=0
- virtual QFuture< quint32 > **savedSearchCount** () const =0
- virtual QFuture< void > **putSavedSearch** (qevercloud::SavedSearch search)=0

- virtual QFuture< std::optional< qevercloud::SavedSearch > > **findSavedSearchByLocalId** (QString savedSearchLocalId) const =0
- virtual QFuture< std::optional< qevercloud::SavedSearch > > **findSavedSearchByGuid** (qevercloud::Guid guid) const =0
- virtual QFuture< std::optional< qevercloud::SavedSearch > > **findSavedSearchByName** (QString name) const =0
- virtual QFuture< QList< qevercloud::SavedSearch > > **listSavedSearches** (ListSavedSearchesOptions options={}) const =0
- virtual QFuture< QSet< qevercloud::Guid > > **listSavedSearchGuids** (ListGuidsFilters filters) const =0
- virtual QFuture< void > **expungeSavedSearchByLocalId** (QString savedSearchLocalId)=0
- virtual QFuture< void > **expungeSavedSearchByGuid** (qevercloud::Guid guid)=0
- virtual QFuture< qint32 > **highestUpdateSequenceNumber** (HighestUsnOption option) const =0
- virtual QFuture< qint32 > **highestUpdateSequenceNumber** (qevercloud::Guid linkedNotebookGuid) const =0
- virtual ILocalStorageNotifier * notifier () const =0

Friends

- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, StartupOption option)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, StartupOption option)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, StartupOptions options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, StartupOptions options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListObjectsFilter filter)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListObjectsFilter filter)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const ListObjectsFilters &filters)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const ListObjectsFilters &filters)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const ListGuidsFilters &filters)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const ListGuidsFilters &filters)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, OrderDirection orderDirection)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, OrderDirection orderDirection)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListNotebooksOrder order)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListNotebooksOrder order)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListLinkedNotebooksOrder order)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListLinkedNotebooksOrder order)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListTagsOrder order)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListTagsOrder order)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListNotesOrder order)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListNotesOrder order)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, ListSavedSearchesOrder order)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, ListSavedSearchesOrder order)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, Affiliation affiliation)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, Affiliation affiliation)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const ListNotebooksOptions &options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const ListNotebooksOptions &options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const ListLinkedNotebooksOptions &options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const ListLinkedNotebooksOptions &options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const ListSavedSearchesOptions &options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const ListSavedSearchesOptions &options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const ListNotesOptions &options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const ListNotesOptions &options)

- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, const ListTagsOptions &options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, const ListTagsOptions &options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, NoteCountOption option)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, NoteCountOption option)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, NoteCountOptions options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, NoteCountOptions options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, UpdateNoteOption option)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, UpdateNoteOption option)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, UpdateNoteOptions options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, UpdateNoteOptions options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, FetchNoteOption option)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, FetchNoteOption option)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, FetchNoteOptions options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, FetchNoteOptions options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, FetchResourceOption option)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, FetchResourceOption option)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, FetchResourceOptions options)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, FetchResourceOptions options)
- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, HighestUsnOption option)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, HighestUsnOption option)

5.29.1 Member Enumeration Documentation

5.29.1.1 Affiliation

```
enum class quentier::local_storage::ILocalStorage::Affiliation  [strong]
```

Denotes whether some data item belongs to user's own account, any of linked notebooks or particular linked notebooks

5.29.1.2 TagNotesRelation

```
enum class quentier::local_storage::ILocalStorage::TagNotesRelation  [strong]
```

Denotes the relation between tag and notes - whether any note us using the given tag

Enumerator

Any	The tag might be used by some notes or it might not be.
WithNotes	The tag is used by some notes.
WithoutNotes	The tag is not used by any note.

5.29.2 Member Function Documentation

5.29.2.1 notifier()

```
virtual ILocalStorageNotifier * quentier::local_storage::ILocalStorage::notifier ( ) const
[pure virtual]
```

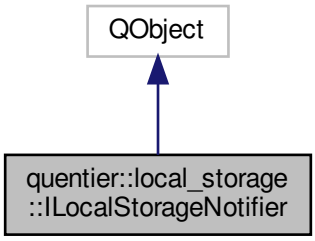
Notifications about the events occurring with the local storage are done via signals emitted by ILocalStorageNotifier.

Returns

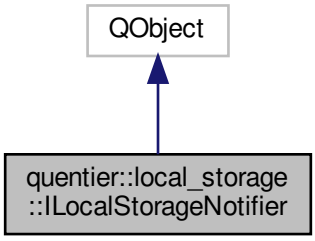
the pointer to ILocalStorageNotifier object which would be valid for at least as long as ILocalStorage object that returned it is alive.

5.30 quentier::local_storage::ILocalStorageNotifier Class Reference

Inheritance diagram for quentier::local_storage::ILocalStorageNotifier:



Collaboration diagram for quentier::local_storage::ILocalStorageNotifier:



Signals

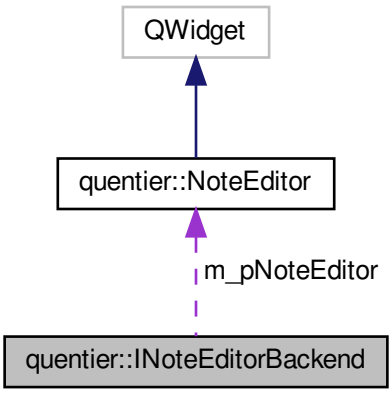
- void **userPut** (qevercloud::User user)
- void **userExpunged** (qevercloud::UserID userId)
- void **notebookPut** (qevercloud::Notebook notebook)
- void **notebookExpunged** (QString notebookLocalId)
- void **linkedNotebookPut** (qevercloud::LinkedNotebook linkedNotebook)
- void **linkedNotebookExpunged** (qevercloud::Guid linkedNotebookGuid)
- void **notePut** (qevercloud::Note note)
- void **noteUpdated** (qevercloud::Note note, ILocalStorage::UpdateNoteOptions options)
- void **noteExpunged** (QString noteLocalId)
- void **tagPut** (qevercloud::Tag tag)
- void **tagExpunged** (QString tagLocalId, QStringList expungedChildTagLocalIds)
- void **resourcePut** (qevercloud::Resource resource)
- void **resourceMetadataPut** (qevercloud::Resource resource)
- void **resourceExpunged** (QString resourceLocalId)
- void **savedSearchPut** (qevercloud::SavedSearch savedSearch)
- void **savedSearchExpunged** (QString savedSearchLocalId)

Protected Member Functions

- **ILocalStorageNotifier** (QObject *parent=nullptr)

5.31 quentier::INoteEditorBackend Class Reference

Collaboration diagram for quentier::INoteEditorBackend:



Public Types

- enum class **Rotation** { **Clockwise** , **Counterclockwise** }

Public Member Functions

- virtual void **initialize** (local_storage::ILocalStoragePtr localStorage, SpellChecker &spellChecker, const Account &account, QThread *pBackgroundJobsThread)=0
- virtual QObject * **object** ()=0
- virtual QWidget * **widget** ()=0
- virtual void **setAccount** (const Account &account)=0
- virtual void **setUndoStack** (QUndoStack *pUndoStack)=0
- virtual void **setInitialPageHtml** (const QString &html)=0
- virtual void **setNoteNotFoundPageHtml** (const QString &html)=0
- virtual void **setNoteDeletedPageHtml** (const QString &html)=0
- virtual void **setNoteLoadingPageHtml** (const QString &html)=0
- virtual bool **isNoteLoaded** () const =0
- virtual qint64 **idleTime** () const =0
- virtual void **convertToNote** ()=0
- virtual void **saveNoteToLocalStorage** ()=0
- virtual void **setNoteTitle** (const QString &noteTitle)=0
- virtual void **setTagIds** (const QStringList &tagLocalUids, const QStringList &tagGuids)=0
- virtual void **undo** ()=0
- virtual void **redo** ()=0
- virtual void **cut** ()=0
- virtual void **copy** ()=0
- virtual void **paste** ()=0
- virtual void **pasteUnformatted** ()=0
- virtual void **selectAll** ()=0
- virtual void **formatSelectionAsSourceCode** ()=0
- virtual void **fontMenu** ()=0
- virtual void **textBold** ()=0
- virtual void **textItalic** ()=0
- virtual void **textUnderline** ()=0
- virtual void **textStrikethrough** ()=0
- virtual void **textHighlight** ()=0
- virtual void **alignLeft** ()=0
- virtual void **alignCenter** ()=0
- virtual void **alignRight** ()=0
- virtual void **alignFull** ()=0
- virtual QString **selectedText** () const =0
- virtual bool **hasSelection** () const =0
- virtual void **findNext** (const QString &text, bool matchCase) const =0
- virtual void **findPrevious** (const QString &text, bool matchCase) const =0
- virtual void **replace** (const QString &textToReplace, const QString &replacementText, bool matchCase)=0
- virtual void **replaceAll** (const QString &textToReplace, const QString &replacementText, bool matchCase)=0
- virtual void **insertToDoCheckbox** ()=0
- virtual void **insertInAppNoteLink** (const QString &userId, const QString &shardId, const QString &noteGuid, const QString &linkText)=0
- virtual void **setSpellcheck** (bool enabled)=0
- virtual bool **spellCheckEnabled** () const =0
- virtual void **setFont** (const QFont &font)=0
- virtual void **setFontHeight** (int height)=0
- virtual void **setFontColor** (const QColor &color)=0
- virtual void **setBackgroundColor** (const QColor &color)=0
- virtual QPalette **defaultPalette** () const =0
- virtual void **setDefaultPalette** (const QPalette &pal)=0
- virtual const QFont * **defaultFont** () const =0
- virtual void **setDefaultFont** (const QFont &font)=0

- virtual void **insertHorizontalLine** ()=0
- virtual void **increaseFontSize** ()=0
- virtual void **decreaseFontSize** ()=0
- virtual void **increaseIndentation** ()=0
- virtual void **decreaseIndentation** ()=0
- virtual void **insertBulletedList** ()=0
- virtual void **insertNumberedList** ()=0
- virtual void **insertTableDialog** ()=0
- virtual void **insertFixedWidthTable** (int rows, int columns, int widthInPixels)=0
- virtual void **insertRelativeWidthTable** (int rows, int columns, double relativeWidth)=0
- virtual void **insertTableRow** ()=0
- virtual void **insertTableColumn** ()=0
- virtual void **removeTableRow** ()=0
- virtual void **removeTableColumn** ()=0
- virtual void **addAttachmentDialog** ()=0
- virtual void **saveAttachmentDialog** (const QByteArray &resourceHash)=0
- virtual void **saveAttachmentUnderCursor** ()=0
- virtual void **openAttachment** (const QByteArray &resourceHash)=0
- virtual void **openAttachmentUnderCursor** ()=0
- virtual void **copyAttachment** (const QByteArray &resourceHash)=0
- virtual void **copyAttachmentUnderCursor** ()=0
- virtual void **removeAttachment** (const QByteArray &resourceHash)=0
- virtual void **removeAttachmentUnderCursor** ()=0
- virtual void **renameAttachment** (const QByteArray &resourceHash)=0
- virtual void **renameAttachmentUnderCursor** ()=0
- virtual void **rotateImageAttachment** (const QByteArray &resourceHash, Rotation rotationDirection)=0
- virtual void **rotateImageAttachmentUnderCursor** (Rotation rotationDirection)=0
- virtual void **encryptSelectedText** ()=0
- virtual void **decryptEncryptedTextUnderCursor** ()=0
- virtual void **decryptEncryptedText** (QString encryptedText, QString cipher, QString keyLength, QString hint, QString enCryptIndex)=0
- virtual void **hideDecryptedTextUnderCursor** ()=0
- virtual void **hideDecryptedText** (QString encryptedText, QString decryptedText, QString cipher, QString keyLength, QString hint, QString enDecryptedIndex)=0
- virtual void **editHyperlinkDialog** ()=0
- virtual void **copyHyperlink** ()=0
- virtual void **removeHyperlink** ()=0
- virtual void **onNoteLoadCancelled** ()=0
- virtual bool **print** (QPrinter &printer, ErrorString &errorDescription)=0
- virtual bool **exportToPdf** (const QString &absoluteFilePath, ErrorString &errorDescription)=0
- virtual bool **exportToEnex** (const QStringList &tagNames, QString &enex, ErrorString &errorDescription)=0
- virtual QString **currentNoteLocalId** () const =0
- virtual void **setCurrentNoteLocalId** (const QString &noteLocalUid)=0
- virtual void **clear** ()=0
- virtual bool **isModified** () const =0
- virtual bool **isEditorPageModified** () const =0
- virtual void **setFocusToEditor** ()=0

Protected Member Functions

- **INoteEditorBackend** (NoteEditor *parent)

Protected Attributes

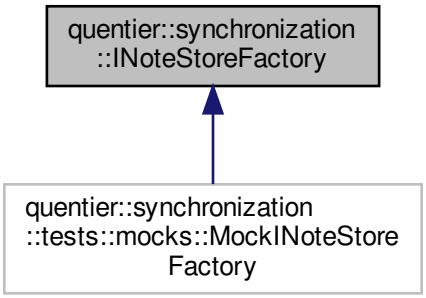
- NoteEditor * **m_pNoteEditor**

Friends

- QUENTIER_EXPORT QTextStream & **operator<<** (QTextStream &strm, Rotation rotation)
- QUENTIER_EXPORT QDebug & **operator<<** (QDebug &dbg, Rotation rotation)

5.32 quentier::synchronization::INoteStoreFactory Class Reference

Inheritance diagram for quentier::synchronization::INoteStoreFactory:

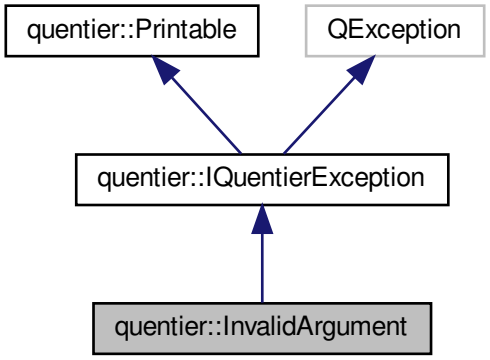


Public Member Functions

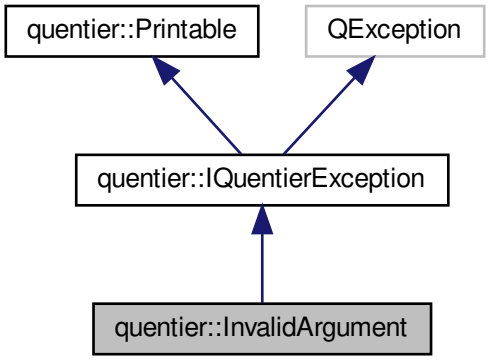
- virtual qevercloud::INoteStorePtr **createNoteStore** (QString noteStoreUrl={}, std::optional< qevercloud::↩ Guid > linkedNotebookGuid={}, qevercloud::IRequestContextPtr ctx={}, qevercloud::IRetryPolicyPtr retry↩ Policy={})=0

5.33 quentier::InvalidArgument Class Reference

Inheritance diagram for quentier::InvalidArgument:



Collaboration diagram for quentier::InvalidArgument:



Public Member Functions

- **InvalidArgument** (ErrorString message)
- InvalidArgument * **clone** () const override
- void **raise** () const override

Protected Member Functions

- QString exceptionDisplayName () const override

5.33.1 Member Function Documentation

5.33.1.1 exceptionDisplayName()

```
QString quentier::InvalidArgument::exceptionDisplayName ( ) const  [override], [protected],
[virtual]
```

Implements quentier::IQuentierException.

5.34 quentier::ResourceRecognitionIndexItem::IObjectItem Struct Reference

Public Member Functions

- virtual QString **objectType** () const =0
- virtual int **weight** () const =0

5.35 quentier::local_storage::IPatch Class Reference

The IPatch interface represents patches of the local storage. Each such patch somehow changes the layout of local storage persistence so that only compliant & corresponding versions of libquentier can be used to work with it.

```
#include <IPatch.h>
```

Public Member Functions

- virtual int fromVersion () const noexcept=0
- virtual int toVersion () const noexcept=0
- virtual QString patchShortDescription () const =0
- virtual QString patchLongDescription () const =0
- virtual QFuture< void > backupLocalStorage ()=0
- virtual QFuture< void > restoreLocalStorageFromBackup ()=0
- virtual QFuture< void > removeLocalStorageBackup ()=0
- virtual QFuture< void > apply ()=0

5.35.1 Detailed Description

The IPatch interface represents patches of the local storage. Each such patch somehow changes the layout of local storage persistence so that only compliant & corresponding versions of libquentier can be used to work with it.

5.35.2 Member Function Documentation

5.35.2.1 apply()

```
virtual QFuture< void > quentier::local_storage::IPatch::apply ( )  [pure virtual]
```

Apply the patch to local storage

Returns

Future which can be awaited for patch application. Contains exception if patch application fails.

5.35.2.2 backupLocalStorage()

```
virtual QFuture< void > quentier::local_storage::IPatch::backupLocalStorage ( )  [pure virtual]
```

Backup either the entire local storage or its parts affected by the particular patch, should be called before applying the patch (but can be skipped if not desired).

Returns

Future which can be awaited for the backup completion. Contains exception if backup fails.

5.35.2.3 fromVersion()

```
virtual int quentier::local_storage::IPatch::fromVersion ( ) const  [pure virtual], [noexcept]
```

Returns

Version of local storage to which the patch needs to be applied

5.35.2.4 patchLongDescription()

```
virtual QString quentier::local_storage::IPatch::patchLongDescription ( ) const  [pure virtual]
```

Returns

Long i.e. detailed description of the patch

5.35.2.5 patchShortDescription()

```
virtual QString quentier::local_storage::IPatch::patchShortDescription ( ) const  [pure virtual]
```

Returns

Short description of the patch

5.35.2.6 removeLocalStorageBackup()

```
virtual QFuture< void > quentier::local_storage::IPatch::removeLocalStorageBackup ( )  [pure
virtual]
```

Remove the previously made backup of local storage, presumably after successful application of the patch so the backup is no longer needed. It won't work if no backup was made before applying a patch, obviously.

Returns

Future which can be awaited for local storage backup removal. Contains exception if backup removal fails.

5.35.2.7 restoreLocalStorageFromBackup()

```
virtual QFuture< void > quentier::local_storage::IPatch::restoreLocalStorageFromBackup ( )
[pure virtual]
```

Restore local storage from previously made backup, presumably after the failed attempt to apply a patch. Won't work if no backup was made before applying a patch, obviously.

Returns

Future which can be awaited for the backup restoration completion. Contains exception if backup restoration fails.

5.35.2.8 toVersion()

```
virtual int quentier::local_storage::IPatch::toVersion ( ) const  [pure virtual], [noexcept]
```

Returns

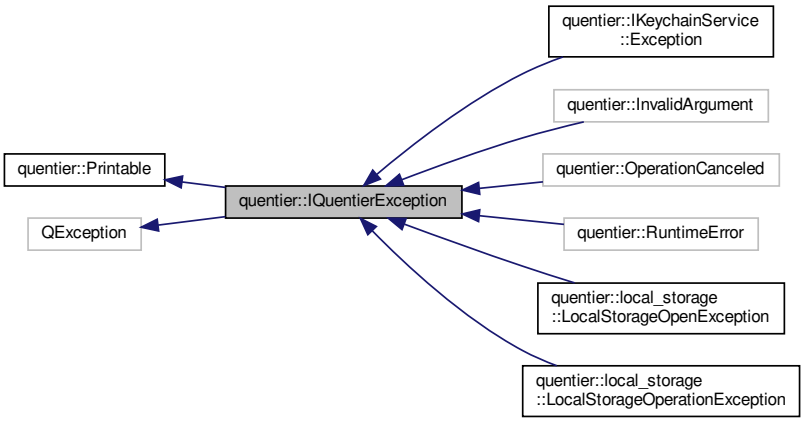
Version of local storage to which the patch would upgrade the local storage

5.36 quentier::IQuentierException Class Reference

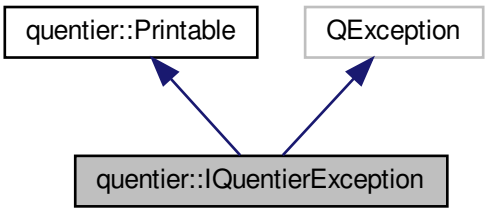
The IQuentierException class represents the interface for exceptions specific to libquentier and applications based on it.

`#include <IQuentierException.h>`

Inheritance diagram for quentier::IQuentierException:



Collaboration diagram for quentier::IQuentierException:



Public Member Functions

- ErrorString **errorMessage** () const
- QString **localizedErrorMessage** () const
- QString **nonLocalizedErrorMessage** () const
- const char * **what** () const noexcept override
- QTextStream & print (QTextStream &strm) const override

Protected Member Functions

- **IQuentierException** (ErrorString message)
- **IQuentierException** (const IQuentierException &other)
- IQuentierException & **operator=** (const IQuentierException &other)
- virtual QString **exceptionDisplayName** () const =0

5.36.1 Detailed Description

The IQuentierException class represents the interface for exceptions specific to libquentier and applications based on it.

In addition to standard exception features inherited from std::exception, IQuentierException based exceptions can provide both localized and non-localized error messages.

5.36.2 Member Function Documentation

5.36.2.1 print()

```
QTextStream & quentier::IQuentierException::print (
            QTextStream & strm ) const  [override], [virtual]
```

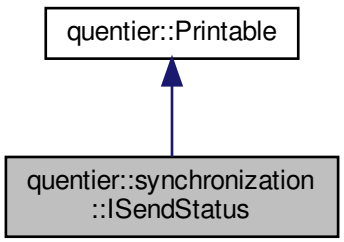
Implements quentier::Printable.

5.37 quentier::synchronization::ISendStatus Class Reference

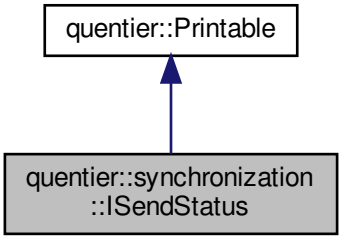
The ISendStatus interface represents the information about the attempt to send information either from user's own account or from some linked notebook to Evernote.

```
#include <ISendStatus.h>
```

Inheritance diagram for quentier::synchronization::ISendStatus:



Collaboration diagram for quentier::synchronization::ISendStatus:



Public Types

- using **QExceptionPtr** = std::shared_ptr< QException >
- using **NoteWithException** = std::pair< qevercloud::Note, QExceptionPtr >
- using **NotebookWithException** = std::pair< qevercloud::Notebook, QExceptionPtr >
- using **SavedSearchWithException** = std::pair< qevercloud::SavedSearch, QExceptionPtr >
- using **TagWithException** = std::pair< qevercloud::Tag, QExceptionPtr >

Public Member Functions

- virtual quint64 totalAttemptedToSendNotes () const =0
- virtual quint64 totalAttemptedToSendNotebooks () const =0
- virtual quint64 totalAttemptedToSendSavedSearches () const =0
- virtual quint64 totalAttemptedToSendTags () const =0
- virtual quint64 totalSuccessfullySentNotes () const =0
- virtual QList< NoteWithException > failedToSendNotes () const =0
- virtual quint64 totalSuccessfullySentNotebooks () const =0
- virtual QList< NotebookWithException > failedToSendNotebooks () const =0
- virtual quint64 totalSuccessfullySentSavedSearches () const =0
- virtual QList< SavedSearchWithException > failedToSendSavedSearches () const =0
- virtual quint64 totalSuccessfullySentTags () const =0
- virtual QList< TagWithException > failedToSendTags () const =0
- virtual StopSynchronizationError stopSynchronizationError () const =0
- virtual bool needToRepeatIncrementalSync () const =0

5.37.1 Detailed Description

The ISendStatus interface represents the information about the attempt to send information either from user's own account or from some linked notebook to Evernote.

5.37.2 Member Function Documentation

5.37.2.1 failedToSendNotebooks()

```
virtual QList< NotebookWithException > quentier::synchronization::ISendStatus::failedToSend↩
Notebooks ( ) const  [pure virtual]
```

Returns

list with notebooks and exceptions representing failures to send these notebooks to Evernote

5.37.2.2 failedToSendNotes()

```
virtual QList< NoteWithException > quentier::synchronization::ISendStatus::failedToSendNotes (
) const  [pure virtual]
```

Returns

list with notes and exceptions representing failures to send these notes to Evernote

5.37.2.3 failedToSendSavedSearches()

```
virtual QList< SavedSearchWithException > quentier::synchronization::ISendStatus::failedTo↩
SendSavedSearches ( ) const  [pure virtual]
```

Returns

list with saved searches and exceptions representing failures to send these saved searches to Evernote

5.37.2.4 failedToSendTags()

```
virtual QList< TagWithException > quentier::synchronization::ISendStatus::failedToSendTags ( )
const  [pure virtual]
```

Returns

list with tags and exceptions representing failures to send these tags to Evernote

5.37.2.5 needToRepeatIncrementalSync()

```
virtual bool quentier::synchronization::ISendStatus::needToRepeatIncrementalSync ( ) const
[pure virtual]
```

If during the send step of synchronization it was found out that Evernote service's state of account has been updated since the last download step, returns true meaning that incremental download step should be repeated. Otherwise returns false.

5.37.2.6 stopSynchronizationError()

```
virtual StopSynchronizationError quentier::synchronization::ISendStatus::stopSynchronization↩
Error ( ) const  [pure virtual]
```

Returns

error which might have occurred during sending the data to Evernote which has prevented further attempts to send anything to Evernote or std::monostate if no such error has occurred

5.37.2.7 totalAttemptedToSendNotebooks()

```
virtual quint64 quentier::synchronization::ISendStatus::totalAttemptedToSendNotebooks ( )
const  [pure virtual]
```

Returns

total number of notebooks attempted to be sent to Evernote

5.37.2.8 totalAttemptedToSendNotes()

```
virtual quint64 quentier::synchronization::ISendStatus::totalAttemptedToSendNotes ( ) const
[pure virtual]
```

Returns

total number of notes attempted to be sent to Evernote

5.37.2.9 totalAttemptedToSendSavedSearches()

```
virtual quint64 quentier::synchronization::ISendStatus::totalAttemptedToSendSavedSearches ( )
const  [pure virtual]
```

Returns

total number of saved searches attempted to be sent to Evernote

5.37.2.10 totalAttemptedToSendTags()

```
virtual quint64 quentier::synchronization::ISendStatus::totalAttemptedToSendTags ( ) const
[pure virtual]
```

Returns

total number of tags attempted to be sent to Evernote

5.37.2.11 totalSuccessfullySentNotebooks()

```
virtual quint64 quentier::synchronization::ISendStatus::totalSuccessfullySentNotebooks ( )
const  [pure virtual]
```

Returns

number of notebooks which were successfully sent to Evernote

5.37.2.12 totalSuccessfullySentNotes()

```
virtual quint64 quentier::synchronization::ISendStatus::totalSuccessfullySentNotes ( ) const
[pure virtual]
```

Returns

number of notes which were successfully sent to Evernote

5.37.2.13 totalSuccessfullySentSavedSearches()

```
virtual quint64 quentier::synchronization::ISendStatus::totalSuccessfullySentSavedSearches ( )
const  [pure virtual]
```

Returns

number of saved searches which were successfully sent to Evernote

5.37.2.14 totalSuccessfullySentTags()

```
virtual quint64 quentier::synchronization::ISendStatus::totalSuccessfullySentTags ( ) const
[pure virtual]
```

Returns

number of tags which were successfully sent to Evernote

5.38 quentier::ResourceRecognitionIndexItem::IShapeItem Struct Reference

Public Member Functions

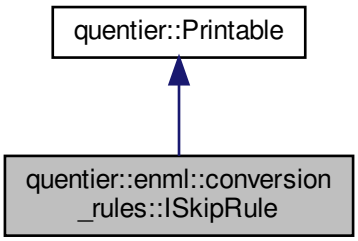
- virtual QString **shape** () const =0
- virtual int **weight** () const =0

5.39 quentier::enml::conversion_rules::ISkipRule Class Reference

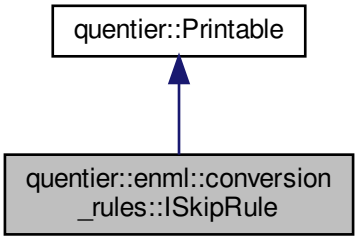
The ISkipRule interface describes a conversion rule with regards to which some ENML/HTML element/attribute should be skipped during the conversion.

```
#include <ISkipRule.h>
```

Inheritance diagram for quentier::enml::conversion_rules::ISkipRule:



Collaboration diagram for quentier::enml::conversion_rules::ISkipRule:



Public Types

- enum class Target { Element , AttibuteName , AttributeValue }

Public Member Functions

- virtual Target target () const =0
- virtual QString value () const =0
- virtual MatchMode matchMode () const =0
- virtual bool includeContents () const =0
- virtual Qt::CaseSensitivity caseSensitivity () const =0
- QTextStream & print (QTextStream &strm) const override

Friends

- QUENTIER_EXPORT QTextStream & **operator**<< (QTextStream &strm, Target target)
- QUENTIER_EXPORT QDebug & **operator**<< (QDebug &dbg, Target target)

5.39.1 Detailed Description

The ISkipRule interface describes a conversion rule with regards to which some ENML/HTML element/attribute should be skipped during the conversion.

ENML format prohibits the use of certain HTML tags and attributes. This interface facilitates skipping these tags and attributes in the process of conversion from HTML to ENML

5.39.2 Member Enumeration Documentation

5.39.2.1 Target

`enum class quentier::enml::conversion_rules::ISkipRule::Target [strong]`

Target to be affected by the skip rule

Enumerator

Element	HTML element
AttibuteName	HTML attribute with specified name
AttributeValue	HTML attribute with specified value

5.39.3 Member Function Documentation

5.39.3.1 caseSensitivity()

```
virtual Qt::CaseSensitivity quentier::enml::conversion_rules::ISkipRule::caseSensitivity ( )
const  [pure virtual]
```

Case sensitivity for target name/value check

5.39.3.2 includeContents()

```
virtual bool quentier::enml::conversion_rules::ISkipRule::includeContents ( ) const  [pure
virtual]
```

Specifies whether the element contents should be included without the element itself if it needs to be skipped or not

5.39.3.3 matchMode()

```
virtual MatchMode quentier::enml::conversion_rules::ISkipRule::matchMode ( ) const  [pure
virtual]
```

Match mode for name or value of the target

5.39.3.4 print()

```
QTextStream & quentier::enml::conversion_rules::ISkipRule::print (
            QTextStream & strm ) const  [override], [virtual]
```

Implements quentier::Printable.

5.39.3.5 target()

```
virtual Target quentier::enml::conversion_rules::ISkipRule::target ( ) const  [pure virtual]
```

Target to be affected by the skip rule

5.39.3.6 value()

```
virtual QString quentier::enml::conversion_rules::ISkipRule::value ( ) const  [pure virtual]
```

Name or value of the target

5.40 quentier::enml::conversion_rules::ISkipRuleBuilder Class Reference

Public Member Functions

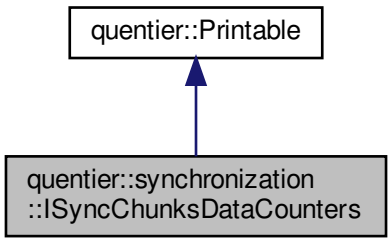
- virtual ISkipRuleBuilder & **setTarget** (ISkipRule::Target target)=0
- virtual ISkipRuleBuilder & **setValue** (QString value)=0
- virtual ISkipRuleBuilder & **setMatchMode** (MatchMode matchMode)=0
- virtual ISkipRuleBuilder & **setIncludeContents** (bool includeContents)=0
- virtual ISkipRuleBuilder & **setCaseSensitivity** (Qt::CaseSensitivity caseSensitivity)=0
- virtual ISkipRulePtr **build** ()=0

5.41 quentier::synchronization::ISyncChunksDataCounters Struct Reference

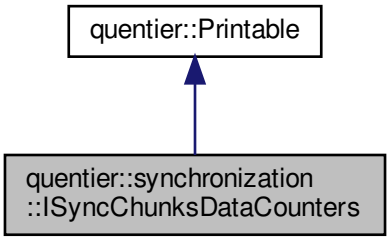
The ISyncChunksDataCounters interface provides integer counters representing the current progress on processing the data from downloaded sync chunks.

```
#include <ISyncChunksDataCounters.h>
```

Inheritance diagram for quentier::synchronization::ISyncChunksDataCounters:



Collaboration diagram for quentier::synchronization::ISyncChunksDataCounters:



Public Member Functions

- virtual quint64 totalSavedSearches () const noexcept=0
- virtual quint64 totalExpungedSavedSearches () const noexcept=0
- virtual quint64 addedSavedSearches () const noexcept=0
- virtual quint64 updatedSavedSearches () const noexcept=0
- virtual quint64 expungedSavedSearches () const noexcept=0
- virtual quint64 totalTags () const noexcept=0
- virtual quint64 totalExpungedTags () const noexcept=0
- virtual quint64 addedTags () const noexcept=0
- virtual quint64 updatedTags () const noexcept=0
- virtual quint64 expungedTags () const noexcept=0
- virtual quint64 totalLinkedNotebooks () const noexcept=0
- virtual quint64 totalExpungedLinkedNotebooks () const noexcept=0
- virtual quint64 addedLinkedNotebooks () const noexcept=0
- virtual quint64 updatedLinkedNotebooks () const noexcept=0
- virtual quint64 expungedLinkedNotebooks () const noexcept=0
- virtual quint64 totalNotebooks () const noexcept=0
- virtual quint64 totalExpungedNotebooks () const noexcept=0
- virtual quint64 addedNotebooks () const noexcept=0
- virtual quint64 updatedNotebooks () const noexcept=0
- virtual quint64 expungedNotebooks () const noexcept=0

5.41.1 Detailed Description

The ISyncChunksDataCounters interface provides integer counters representing the current progress on processing the data from downloaded sync chunks.

5.41.2 Member Function Documentation

5.41.2.1 addedLinkedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::addedLinkedNotebooks ( )
const  [pure virtual], [noexcept]
```

Number of linked notebooks from sync chunks added to the local storage so far

5.41.2.2 addedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::addedNotebooks ( ) const
[pure virtual], [noexcept]
```

Number of notebooks from sync chunks added to the local storage so far

5.41.2.3 addedSavedSearches()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::addedSavedSearches ( )
const  [pure virtual], [noexcept]
```

Number of saved searches from sync chunks added to the local storage so far

5.41.2.4 addedTags()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::addedTags ( ) const  [pure
virtual], [noexcept]
```

Number of tags from sync chunks added to the local storage so far

5.41.2.5 expungedLinkedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::expungedLinkedNotebooks (
) const  [pure virtual], [noexcept]
```

Number of linked notebooks from sync chunks expunged from the local storage so far

5.41.2.6 expungedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::expungedNotebooks ( )
const  [pure virtual], [noexcept]
```

Number of notebooks from sync chunks expunged from the local storage so far

5.41.2.7 expungedSavedSearches()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::expungedSavedSearches ( )
const  [pure virtual], [noexcept]
```

Number of saved searches from sync chunks expunged from the local storage so far

5.41.2.8 expungedTags()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::expungedTags ( ) const
[pure virtual], [noexcept]
```

Number of tags from sync chunks expunged from the local storage so far

5.41.2.9 totalExpungedLinkedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalExpungedLinked↩
Notebooks ( ) const  [pure virtual], [noexcept]
```

Total number of expunged saved searches in downloaded sync chunks

5.41.2.10 totalExpungedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalExpungedNotebooks ( )
const  [pure virtual], [noexcept]
```

Total number of expunged notebooks in downloaded sync chunks

5.41.2.11 totalExpungedSavedSearches()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalExpungedSavedSearches
( ) const  [pure virtual], [noexcept]
```

Total number of expunged saved searches in downloaded sync chunks

5.41.2.12 totalExpungedTags()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalExpungedTags ( )
const  [pure virtual], [noexcept]
```

Total number of expunged tags in downloaded sync chunks

5.41.2.13 totalLinkedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalLinkedNotebooks ( )
const  [pure virtual], [noexcept]
```

Total number of new or updated linked notebooks in downloaded sync chunks

5.41.2.14 totalNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalNotebooks ( ) const
[pure virtual], [noexcept]
```

Total number of new or updated notebooks in downloaded sync chunks

5.41.2.15 totalSavedSearches()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalSavedSearches ( )
const  [pure virtual], [noexcept]
```

Total number of new or updated saved searches in downloaded sync chunks

5.41.2.16 totalTags()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::totalTags ( ) const  [pure
virtual], [noexcept]
```

Total number of new or updated tags in downloaded sync chunks

5.41.2.17 updatedLinkedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::updatedLinkedNotebooks ( )
const  [pure virtual], [noexcept]
```

Number of linked notebooks from sync chunks updated in the local storage so far

5.41.2.18 updatedNotebooks()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::updatedNotebooks ( ) const
[pure virtual], [noexcept]
```

Number of notebooks from sync chunks updated in the local storage so far

5.41.2.19 updatedSavedSearches()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::updatedSavedSearches ( )
const  [pure virtual], [noexcept]
```

Number of saved searches from sync chunks updated in the local storage so far

5.41.2.20 updatedTags()

```
virtual quint64 quentier::synchronization::ISyncChunksDataCounters::updatedTags ( ) const
[pure virtual], [noexcept]
```

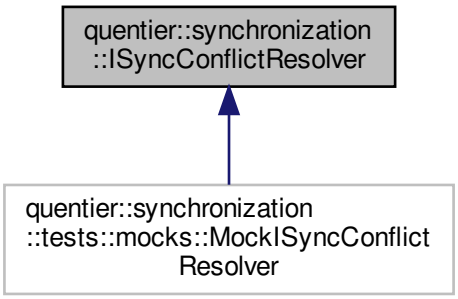
Number of tags from sync chunks updated in the local storage so far

5.42 quentier::synchronization::ISyncConflictResolver Class Reference

The ISyncConflictResolver interface provides methods used to resolve conflicts between local and remote versions of the same data item.

`#include <ISyncConflictResolver.h>`

Inheritance diagram for quentier::synchronization::ISyncConflictResolver:



Classes

- struct ConflictResolution

  The ConflictResolution struct is a namespace inside which several other structs determining actual conflict resolutions.

Public Types

- using **NotebookConflictResolution** = std::variant< ConflictResolution::UseTheirs, ConflictResolution::UseMine, ConflictResolution::IgnoreMine, ConflictResolution::MoveMine< qevercloud::Notebook > >
- using **NoteConflictResolution** = std::variant< ConflictResolution::UseTheirs, ConflictResolution::UseMine, ConflictResolution::IgnoreMine, ConflictResolution::MoveMine< qevercloud::Note > >
- using **SavedSearchConflictResolution** = std::variant< ConflictResolution::UseTheirs, ConflictResolution::UseMine, ConflictResolution::IgnoreMine, ConflictResolution::MoveMine< qevercloud::SavedSearch > >
- using **TagConflictResolution** = std::variant< ConflictResolution::IgnoreMine, ConflictResolution::UseTheirs, ConflictResolution::UseMine, ConflictResolution::MoveMine< qevercloud::Tag > >

Public Member Functions

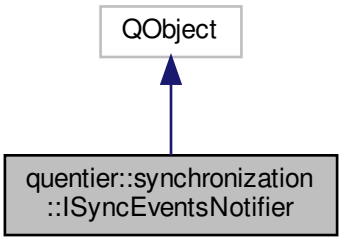
- virtual QFuture< NotebookConflictResolution > **resolveNotebookConflict** (qevercloud::Notebook theirs, qevercloud::Notebook mine)=0
- virtual QFuture< NoteConflictResolution > **resolveNoteConflict** (qevercloud::Note theirs, qevercloud::Note mine)=0
- virtual QFuture< SavedSearchConflictResolution > **resolveSavedSearchConflict** (qevercloud::Saved↩ Search theirs, qevercloud::SavedSearch mine)=0
- virtual QFuture< TagConflictResolution > **resolveTagConflict** (qevercloud::Tag theirs, qevercloud::Tag mine)=0

5.42.1 Detailed Description

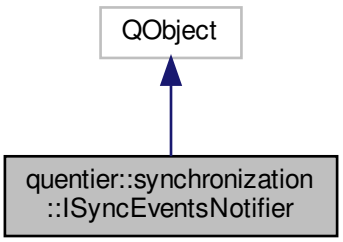
The ISyncConflictResolver interface provides methods used to resolve conflicts between local and remote versions of the same data item.

5.43 quentier::synchronization::ISyncEventsNotifier Class Reference

Inheritance diagram for quentier::synchronization::ISyncEventsNotifier:



Collaboration diagram for quentier::synchronization::ISyncEventsNotifier:



Signals

- void syncChunksDownloadProgress (qint32 highestDownloadedUsn, qint32 highestServerUsn, qint32 last↩ PreviousUsn)
- void syncChunksDownloaded ()
- void syncChunksDataProcessingProgress (ISyncChunksDataCountersPtr counters)
- void startLinkedNotebooksDataDownloading (const QList< qevercloud::LinkedNotebook > &linked↩ Notebooks)
- void linkedNotebookSyncChunksDownloadProgress (qint32 highestDownloadedUsn, qint32 highestServer↩ Usn, qint32 lastPreviousUsn, const qevercloud::LinkedNotebook &linkedNotebook)

- void linkedNotebookSyncChunksDownloaded (const qevercloud::LinkedNotebook &linkedNotebook)
- void linkedNotebookSyncChunksDataProcessingProgress (ISyncChunksDataCountersPtr counters, const qevercloud::LinkedNotebook &linkedNotebook)
- void notesDownloadProgress (quint32 notesDownloaded, quint32 totalNotesToDownload)
- void linkedNotebookNotesDownloadProgress (quint32 notesDownloaded, quint32 totalNotesToDownload, const qevercloud::LinkedNotebook &linkedNotebook)
- void resourcesDownloadProgress (quint32 resourcesDownloaded, quint32 totalResourcesToDownload)
- void linkedNotebookResourcesDownloadProgress (quint32 resourcesDownloaded, quint32 totalResources↩ ToDownload, const qevercloud::LinkedNotebook &linkedNotebook)
- void downloadFinished (bool dataDownloaded)
- void userOwnSendStatusUpdate (ISendStatusPtr sendStatus)
- void linkedNotebookSendStatusUpdate (const qevercloud::Guid &linkedNotebookGuid, ISendStatusPtr sendStatus)

Protected Member Functions

- **ISyncEventsNotifier** (QObject ∗parent=nullptr)

5.43.1 Member Function Documentation

5.43.1.1 downloadFinished

```
void quentier::synchronization::ISyncEventsNotifier::downloadFinished (
            bool dataDownloaded )  [signal]
```

This signal is emitted when the initial download step is finished.

Parameters

dataDownloaded	True if some data was actually downloaded during this step i.e. there were some updates on Evernote servers compared to local state, false otherwise.

5.43.1.2 linkedNotebookNotesDownloadProgress

```
void quentier::synchronization::ISyncEventsNotifier::linkedNotebookNotesDownloadProgress (
            quint32 notesDownloaded,
            quint32 totalNotesToDownload,
            const qevercloud::LinkedNotebook & linkedNotebook )  [signal]
```

This signal is emitted on each successful download of full note data from some linked notebook.

Parameters

notesDownloaded	The number of notes downloaded by the moment
totalNotesToDownload	The total number of notes that need to be downloaded
linkedNotebook	The linked notebook which notes download progress is being reported

5.43.1.3 linkedNotebookResourcesDownloadProgress

```
void quentier::synchronization::ISyncEventsNotifier::linkedNotebookResourcesDownloadProgress (
            quint32 resourcesDownloaded,
            quint32 totalResourcesToDownload,
            const qevercloud::LinkedNotebook & linkedNotebook )  [signal]
```

This signal is emitted on each successful download of full resource data from linked notebooks during the incremental sync (as individual resources are downloaded along with their notes during full sync).

Parameters

resourcesDownloaded	The number of resources downloaded by the moment
totalResourcesToDownload	The total number of resources that need to be downloaded
linkedNotebook	The linked notebook which resources download progress is being reported

5.43.1.4 linkedNotebookSendStatusUpdate

```
void quentier::synchronization::ISyncEventsNotifier::linkedNotebookSendStatusUpdate (
            const qevercloud::Guid & linkedNotebookGuid,
            ISendStatusPtr sendStatus )  [signal]
```

This signal is emitted on each successful or unsuccessful attempt to send some new or locally modified data item from some linked notebook to Evernote.

Parameters

linkedNotebookGuid	Guid of the linked notebook for which the send status was updated
sendStatus	The updated send status

5.43.1.5 linkedNotebookSyncChunksDataProcessingProgress

```
void quentier::synchronization::ISyncEventsNotifier::linkedNotebookSyncChunksDataProcessing↩
Progress (
            ISyncChunksDataCountersPtr counters,
            const qevercloud::LinkedNotebook & linkedNotebook )  [signal]
```

This signal is emitted during some linked notebook's downloaded sync chunks contents processing and denotes the progress on that step.

Parameters

counters	Updated sync chunks data counters
linkedNotebook	The linked notebook which sync chunks data processing progress is being reported

5.43.1.6 linkedNotebookSyncChunksDownloaded

```
void quentier::synchronization::ISyncEventsNotifier::linkedNotebookSyncChunksDownloaded (
            const qevercloud::LinkedNotebook & linkedNotebook )  [signal]
```

This signal is emitted when the sync chunks for data from some linked notebook are downloaded during "remote to local" synchronization step

Parameters

linkedNotebook	The linked notebook which sync chunks were downloaded

5.43.1.7 linkedNotebookSyncChunksDownloadProgress

```
void quentier::synchronization::ISyncEventsNotifier::linkedNotebookSyncChunksDownloadProgress
(
            qint32 highestDownloadedUsn,
            qint32 highestServerUsn,
            qint32 lastPreviousUsn,
            const qevercloud::LinkedNotebook & linkedNotebook )  [signal]
```

This signal is emitted during linked notebooks sync chunks downloading and denotes the progress of that step, individually for each linked notebook. The percentage of completeness can be computed roughly as (highest↩ DownloadedUsn - lastPreviousUsn) / (highestServerUsn - lastPreviousUsn) ∗ 100%.

Parameters

highestDownloadedUsn	The highest update sequence number within data items from linked notebook sync chunks downloaded so far
highestServerUsn	The current highest update sequence number within the linked notebook
lastPreviousUsn	The last update sequence number from previous sync of the given linked notebook; if current sync is the first one, this value is zero
linkedNotebook	The linked notebook which sync chunks download progress is reported

5.43.1.8 notesDownloadProgress

```
void quentier::synchronization::ISyncEventsNotifier::notesDownloadProgress (
            quint32 notesDownloaded,
            quint32 totalNotesToDownload )  [signal]
```

This signal is emitted on each successful download of full note data from user's own account.

Parameters

notesDownloaded	The number of notes downloaded by the moment
totalNotesToDownload	The total number of notes that need to be downloaded

5.43.1.9 resourcesDownloadProgress

```
void quentier::synchronization::ISyncEventsNotifier::resourcesDownloadProgress (
            quint32 resourcesDownloaded,
            quint32 totalResourcesToDownload )  [signal]
```

This signal is emitted on each successful doenload of full resource data from user's own account during the incremental sync (as individual resources are downloaded along with their notes during full sync).

Parameters

resourcesDownloaded	The number of resources downloaded by the moment
totalResourcesToDownload	The total number of resources that need to be downloaded

5.43.1.10 startLinkedNotebooksDataDownloading

```
void quentier::synchronization::ISyncEventsNotifier::startLinkedNotebooksDataDownloading (
            const QList< qevercloud::LinkedNotebook > & linkedNotebooks )  [signal]
```

This signal is emitted before the downloading of data corresponding to linked notebooks starts.

Parameters

linkedNotebooks	Linked notebooks the data from which will start being downloaded after the execution of this callback

5.43.1.11 syncChunksDataProcessingProgress

```
void quentier::synchronization::ISyncEventsNotifier::syncChunksDataProcessingProgress (
            ISyncChunksDataCountersPtr counters )  [signal]
```

This signal is emitted during user own account's downloaded sync chunks contents processing and denotes the progress on that step.

5.43.1.12 syncChunksDownloaded

```
void quentier::synchronization::ISyncEventsNotifier::syncChunksDownloaded ( )  [signal]
```

This signal is emitted when the sync chunks for data from user's own account are downloaded during the download synchronization step.

5.43.1.13 syncChunksDownloadProgress

```
void quentier::synchronization::ISyncEventsNotifier::syncChunksDownloadProgress (
            qint32 highestDownloadedUsn,
            qint32 highestServerUsn,
            qint32 lastPreviousUsn ) [signal]
```

This signal is emitted during user own account's sync chunks downloading and denotes the progress of that step. The percentage of completeness can be computed roughly as (highestDownloadedUsn - lastPreviousUsn) / (highestServerUsn - lastPreviousUsn) * 100%.

Parameters

highestDownloadedUsn	The highest update sequence number within data items from sync chunks downloaded so far
highestServerUsn	The current highest update sequence number within the account
lastPreviousUsn	The last update sequence number from previous sync; if current sync is the first one, this value is zero

5.43.1.14 userOwnSendStatusUpdate

```
void quentier::synchronization::ISyncEventsNotifier::userOwnSendStatusUpdate (
            ISendStatusPtr sendStatus ) [signal]
```

This signal is emitted on each successful or unsuccessful attempt to send some new or locally modified data item from user's own account to Evernote.

Parameters

sendStatus	The updated send status

5.44 quentier::synchronization::ISynchronizer Class Reference

Public Types

- using **SyncResult** = std::pair< QFuture< ISyncResultPtr >, ISyncEventsNotifier * >

Public Member Functions

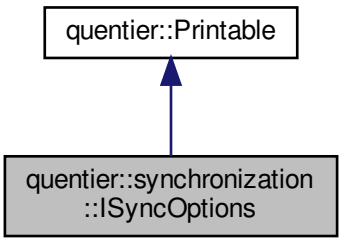
- virtual QFuture< std::pair< Account, IAuthenticationInfoPtr > > **authenticateNewAccount** ()=0
- virtual QFuture< IAuthenticationInfoPtr > **authenticateAccount** (Account account)=0
- virtual SyncResult **synchronizeAccount** (Account account, local_storage::ILocalStoragePtr localStorage, utility::cancelers::ICancelerPtr canceler, ISyncOptionsPtr options=nullptr, ISyncConflictResolverPtr syncConflictResolver=nullptr)=0
- virtual void **revokeAuthentication** (qevercloud::UserID userId)=0

5.45 quentier::synchronization::ISyncOptions Class Reference

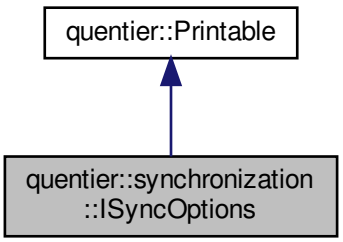
Options for synchronization process.

`#include <ISyncOptions.h>`

Inheritance diagram for quentier::synchronization::ISyncOptions:



Collaboration diagram for quentier::synchronization::ISyncOptions:



Public Member Functions

- virtual bool downloadNoteThumbnails () const =0
- virtual std::optional< QDir > inkNoteImagesStorageDir () const =0
- virtual qevercloud::IRequestContextPtr requestContext () const =0
- virtual qevercloud::IRetryPolicyPtr retryPolicy () const =0
- virtual std::optional< quint32 > maxConcurrentNoteDownloads () const =0
- virtual std::optional< quint32 > maxConcurrentResourceDownloads () const =0

5.45.1 Detailed Description

Options for synchronization process.

5.45.2 Member Function Documentation

5.45.2.1 downloadNoteThumbnails()

```
virtual bool quentier::synchronization::ISyncOptions::downloadNoteThumbnails ( ) const  [pure
virtual]
```

Flag to enable or disable downloading of note thumbnails during the sync. Note thumbnails are stored inside the local storage along with other note data.

5.45.2.2 inkNoteImagesStorageDir()

```
virtual std::optional< QDir > quentier::synchronization::ISyncOptions::inkNoteImagesStorageDir
( ) const  [pure virtual]
```

Directory to store the downloaded ink note images. If this method returns std::nullopt, ink note images would not be downloaded during the sync.

Ink notes images data is stored inside note's resources but the format of the data is not documented, which makes it quite hard to implement note editor able to fully handle ink notes. An easier option is to visualize a static image corresponding to the last revision of the ink note. Such images need to be downloaded separately during the sync if they are required.

Ink note images are stored right in this directory without any subdirectories, file names correspond to pattern <resource guid>.png.

5.45.2.3 maxConcurrentNoteDownloads()

```
virtual std::optional< quint32 > quentier::synchronization::ISyncOptions::maxConcurrentNote↩
Downloads ( ) const  [pure virtual]
```

Maximal number of concurrent note downloads. Allowing unlimited concurrent note downloads can lead to errors due to reaching the platform's limit on the number of open files. If std::nullopt is returned, the default limit is used.

5.45.2.4 maxConcurrentResourceDownloads()

```
virtual std::optional< quint32 > quentier::synchronization::ISyncOptions::maxConcurrent↩
ResourceDownloads ( ) const  [pure virtual]
```

Maximal number of concurrent resource downloads. Allowing unlimited concurrent resource downloads can lead to errors due to reaching the platform's limit on the number of open files. If std::nullopt is returned, the default limit is used.

5.45.2.5 requestContext()

```
virtual qevercloud::IRequestContextPtr quentier::synchronization::ISyncOptions::requestContext
( ) const  [pure virtual]
```

Request context with settings which should be used during the sync. If nullptr then request context with default settings would be used.

5.45.2.6 retryPolicy()

```
virtual qevercloud::IRetryPolicyPtr quentier::synchronization::ISyncOptions::retryPolicy ( )
const  [pure virtual]
```

Retry policy which should be used during the sync. If nullptr then the default retry policy would be used.

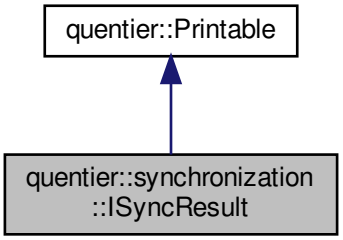
5.46 quentier::synchronization::ISyncOptionsBuilder Class Reference

Public Member Functions

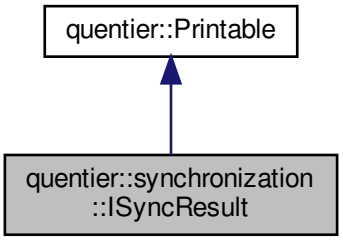
- virtual ISyncOptionsBuilder & **setDownloadNoteThumbnails** (bool value)=0
- virtual ISyncOptionsBuilder & **setInkNoteImagesStorageDir** (std::optional< QDir > dir)=0
- virtual ISyncOptionsBuilder & **setRequestContext** (qevercloud::IRequestContextPtr ctx)=0
- virtual ISyncOptionsBuilder & **setRetryPolicy** (qevercloud::IRetryPolicyPtr retryPolicy)=0
- virtual ISyncOptionsBuilder & **setMaxConcurrentNoteDownloads** (std::optional< quint32 > max↩ ConcurrentNoteDownloads)=0
- virtual ISyncOptionsBuilder & **setMaxConcurrentResourceDownloads** (std::optional< quint32 > max↩ ConcurrentResourceDownloads)=0
- virtual ISyncOptionsPtr **build** ()=0

5.47 quentier::synchronization::ISyncResult Class Reference

Inheritance diagram for quentier::synchronization::ISyncResult:



Collaboration diagram for quentier::synchronization::ISyncResult:



Public Member Functions

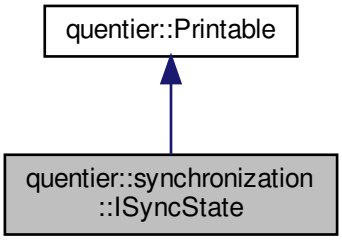
- virtual ISyncStatePtr **syncState** () const =0
- virtual ISyncChunksDataCountersPtr **userAccountSyncChunksDataCounters** () const =0
- virtual QHash< qevercloud::Guid, ISyncChunksDataCountersPtr > **linkedNotebookSyncChunksData↩ Counters** () const =0
- virtual bool **userAccountSyncChunksDownloaded** () const =0
- virtual QSet< qevercloud::Guid > **linkedNotebookGuidsWithSyncChunksDownloaded** () const =0
- virtual IDownloadNotesStatusPtr **userAccountDownloadNotesStatus** () const =0
- virtual QHash< qevercloud::Guid, IDownloadNotesStatusPtr > **linkedNotebookDownloadNotesStatuses** () const =0
- virtual IDownloadResourcesStatusPtr **userAccountDownloadResourcesStatus** () const =0
- virtual QHash< qevercloud::Guid, IDownloadResourcesStatusPtr > **linkedNotebookDownload↩ ResourcesStatuses** () const =0
- virtual ISendStatusPtr **userAccountSendStatus** () const =0
- virtual QHash< qevercloud::Guid, ISendStatusPtr > **linkedNotebookSendStatuses** () const =0
- virtual StopSynchronizationError **stopSynchronizationError** () const =0

5.48 quentier::synchronization::ISyncState Class Reference

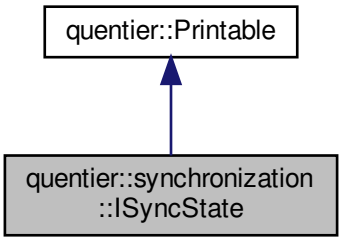
The ISyncState interface provides accessory methods to determine the sync state for the account.

`#include <ISyncState.h>`

Inheritance diagram for quentier::synchronization::ISyncState:



Collaboration diagram for quentier::synchronization::ISyncState:



Public Member Functions

- virtual qint32 **userDataUpdateCount** () const =0
- virtual qevercloud::Timestamp **userDataLastSyncTime** () const =0
- virtual QHash< qevercloud::Guid, qint32 > **linkedNotebookUpdateCounts** () const =0
- virtual QHash< qevercloud::Guid, qevercloud::Timestamp > **linkedNotebookLastSyncTimes** () const =0

5.48.1 Detailed Description

The ISyncState interface provides accessory methods to determine the sync state for the account.

5.49 quentier::synchronization::ISyncStateBuilder Class Reference

Public Member Functions

- virtual ISyncStateBuilder & **setUserDataUpdateCount** (qint32 updateCount)=0

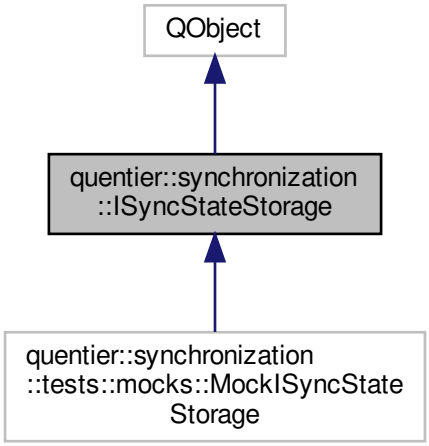
- virtual ISyncStateBuilder & **setUserDataLastSyncTime** (qevercloud::Timestamp lastSyncTime)=0
- virtual ISyncStateBuilder & **setLinkedNotebookUpdateCounts** (QHash< qevercloud::Guid, qint32 > updateCounts)=0
- virtual ISyncStateBuilder & **setLinkedNotebookLastSyncTimes** (QHash< qevercloud::Guid, qevercloud::Timestamp > lastSyncTimes)=0
- virtual ISyncStatePtr **build** ()=0

5.50 quentier::synchronization::ISyncStateStorage Class Reference

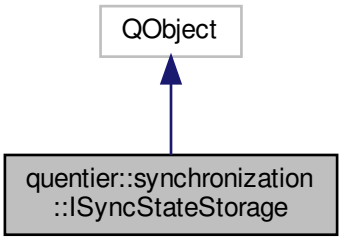
The ISyncStateStorage interface represents the interface of a class which stores sync state for given accounts persistently and provides access to previously stores sync states.

`#include <ISyncStateStorage.h>`

Inheritance diagram for quentier::synchronization::ISyncStateStorage:



Collaboration diagram for quentier::synchronization::ISyncStateStorage:



Signals

- void notifySyncStateUpdated (Account account, ISyncStatePtr syncState)

Public Member Functions

- virtual ISyncStatePtr **getSyncState** (const Account &account)=0
- virtual void **setSyncState** (const Account &account, ISyncStatePtr syncState)=0

Protected Member Functions

- **ISyncStateStorage** (QObject *parent=nullptr)

5.50.1 Detailed Description

The ISyncStateStorage interface represents the interface of a class which stores sync state for given accounts persistently and provides access to previously stores sync states.

5.50.2 Member Function Documentation

5.50.2.1 notifySyncStateUpdated

```
void quentier::synchronization::ISyncStateStorage::notifySyncStateUpdated (
            Account account,
            ISyncStatePtr syncState )  [signal]
```

Classes implementing ISyncStateStorage interface are expected to emit notifySyncStateUpdated signal each time when sync state for the corresponding account is updated

5.51 quentier::ResourceRecognitionIndexItem::ITextItem Struct Reference

Public Member Functions

- virtual QString **text** () const =0
- virtual int **weight** () const =0

5.52 quentier::synchronization::IUserStoreFactory Class Reference

Public Member Functions

- virtual qevercloud::IUserStorePtr **createUserStore** (QString userStoreUrl={}, qevercloud::IRequest↩ ContextPtr ctx={}, qevercloud::IRetryPolicyPtr retryPolicy={})=0

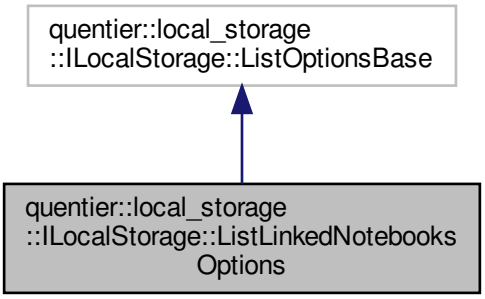
5.53 quentier::local_storage::ILocalStorage::ListGuidsFilters Struct Reference

Public Attributes

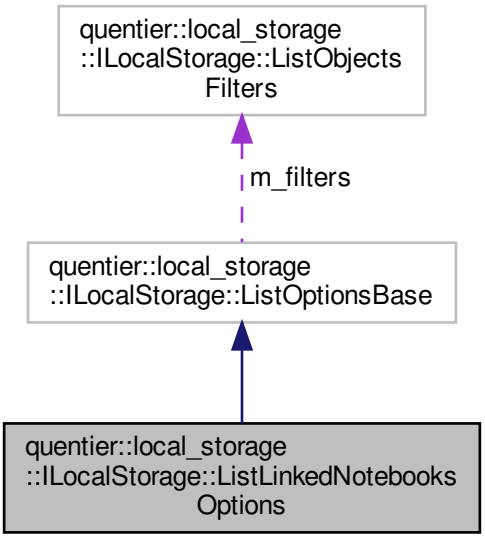
- std::optional< ListObjectsFilter > **m_locallyModifiedFilter**
- std::optional< ListObjectsFilter > **m_locallyFavoritedFilter**

5.54 quentier::local_storage::ILocalStorage::ListLinkedNotebooks↩ Options Struct Reference

Inheritance diagram for quentier::local_storage::ILocalStorage::ListLinkedNotebooksOptions:



Collaboration diagram for quentier::local_storage::ILocalStorage::ListLinkedNotebooksOptions:

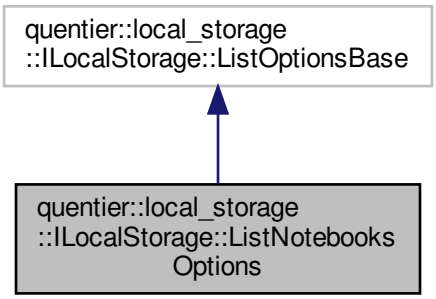


Public Attributes

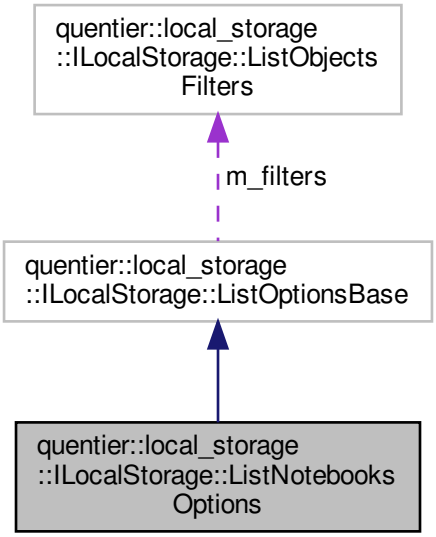
- ListLinkedNotebooksOrder **m_order** = ListLinkedNotebooksOrder::NoOrder

5.55 quentier::local_storage::ILocalStorage::ListNotebooksOptions Struct Reference

Inheritance diagram for quentier::local_storage::ILocalStorage::ListNotebooksOptions:



Collaboration diagram for quentier::local_storage::ILocalStorage::ListNotebooksOptions:

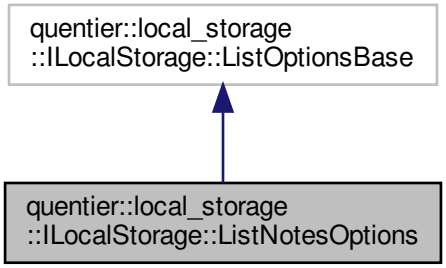


Public Attributes

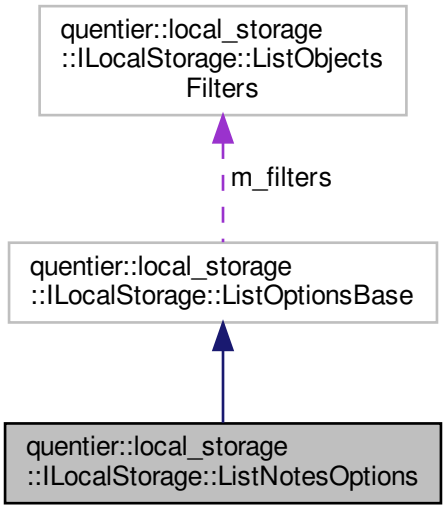
- ListNotebooksOrder **m_order** = ListNotebooksOrder::NoOrder
- Affiliation **m_affiliation** = Affiliation::Any
- QList< qevercloud::Guid > **m_linkedNotebookGuids**

5.56 quentier::local_storage::ILocalStorage::ListNotesOptions Struct Reference

Inheritance diagram for quentier::local_storage::ILocalStorage::ListNotesOptions:



Collaboration diagram for quentier::local_storage::ILocalStorage::ListNotesOptions:



Public Attributes

- ListNotesOrder **m_order** = ListNotesOrder::NoOrder

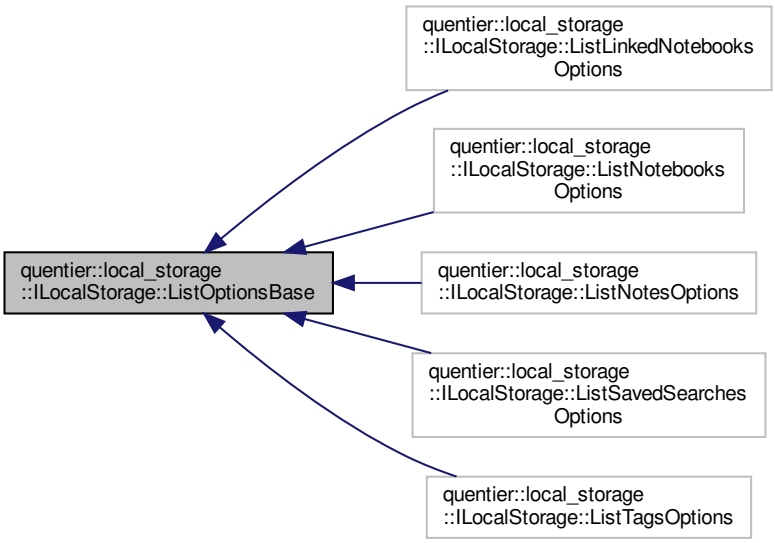
5.57 quentier::local_storage::ILocalStorage::ListObjectsFilters Struct Reference

Public Attributes

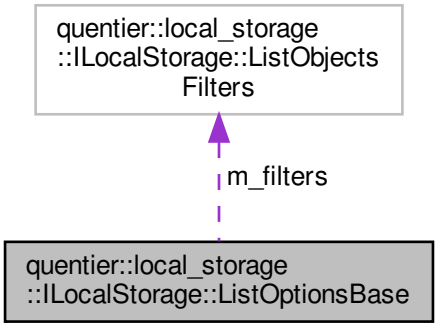
- std::optional< ListObjectsFilter > **m_locallyModifiedFilter**
- std::optional< ListObjectsFilter > **m_withGuidFilter**
- std::optional< ListObjectsFilter > **m_localOnlyFilter**
- std::optional< ListObjectsFilter > **m_locallyFavoritedFilter**

5.58 quentier::local_storage::ILocalStorage::ListOptionsBase Struct Reference

Inheritance diagram for quentier::local_storage::ILocalStorage::ListOptionsBase:



Collaboration diagram for quentier::local_storage::ILocalStorage::ListOptionsBase:

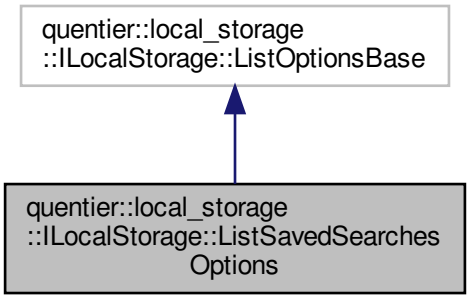


Public Attributes

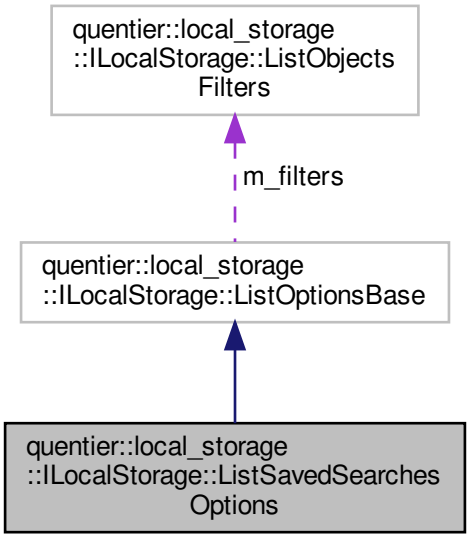
- ListObjectsFilters **m_filters** = {}
- quint64 **m_limit** = 0UL
- quint64 **m_offset** = 0UL
- OrderDirection **m_direction** = OrderDirection::Ascending

5.59 quentier::local_storage::ILocalStorage::ListSavedSearchesOptions Struct Reference

Inheritance diagram for quentier::local_storage::ILocalStorage::ListSavedSearchesOptions:



Collaboration diagram for quentier::local_storage::ILocalStorage::ListSavedSearchesOptions:

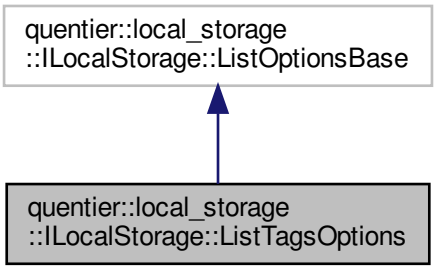


Public Attributes

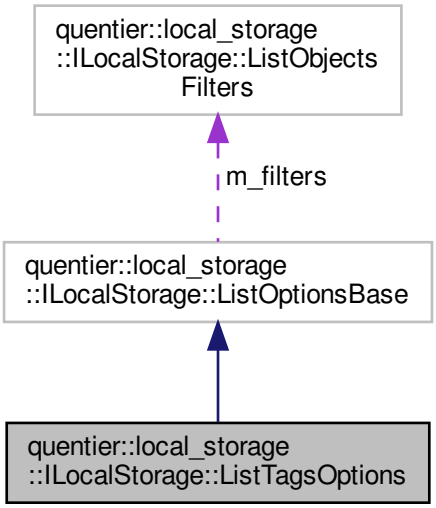
- ListSavedSearchesOrder **m_order** = ListSavedSearchesOrder::NoOrder

5.60 quentier::local_storage::ILocalStorage::ListTagsOptions Struct Reference

Inheritance diagram for quentier::local_storage::ILocalStorage::ListTagsOptions:



Collaboration diagram for quentier::local_storage::ILocalStorage::ListTagsOptions:



Public Attributes

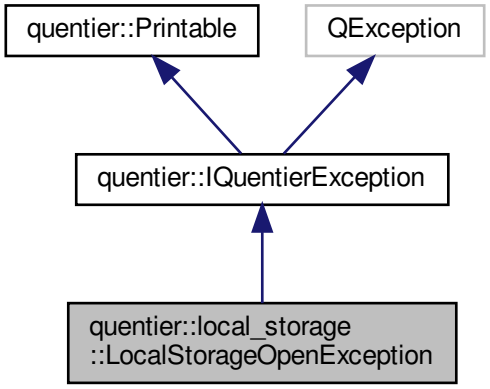
- ListTagsOrder **m_order** = ListTagsOrder::NoOrder
- Affiliation **m_affiliation** = Affiliation::Any
- QList< qevercloud::Guid > **m_linkedNotebookGuids**
- TagNotesRelation **m_tagNotesRelation** = TagNotesRelation::Any

5.61 quentier::local_storage::LocalStorageOpenException Class Reference

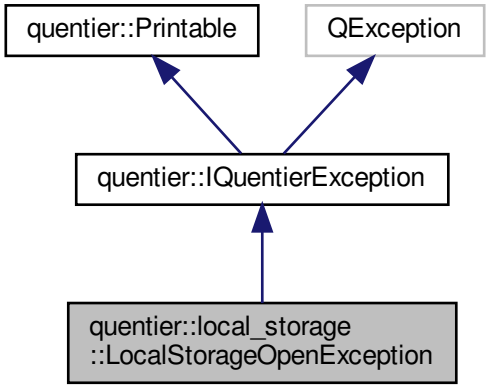
The LocalStorageOpenException is thrown on failure to open the local storage database.

`#include <LocalStorageOpenException.h>`

Inheritance diagram for quentier::local_storage::LocalStorageOpenException:



Collaboration diagram for quentier::local_storage::LocalStorageOpenException:



Public Member Functions

- **LocalStorageOpenException** (const ErrorString &message)
- LocalStorageOpenException * **clone** () const override
- void **raise** () const override

Protected Member Functions

- QString exceptionDisplayName () const override

5.61.1 Detailed Description

The LocalStorageOpenException is thrown on failure to open the local storage database.

5.61.2 Member Function Documentation

5.61.2.1 exceptionDisplayName()

```
QString quentier::local_storage::LocalStorageOpenException::exceptionDisplayName ( ) const
[override], [protected], [virtual]
```

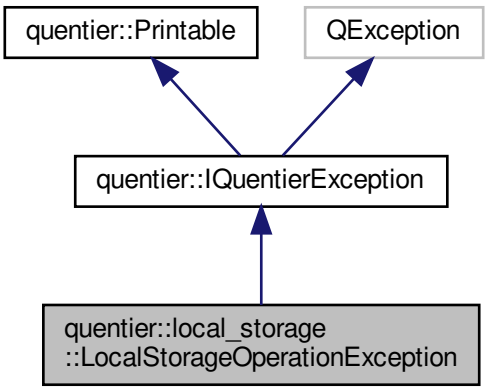
Implements quentier::IQuentierException.

5.62 quentier::local_storage::LocalStorageOperationException Class Reference

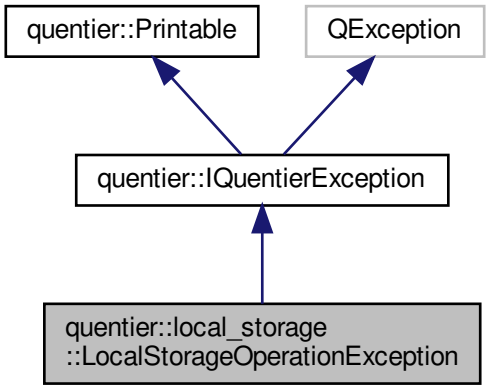
The LocalStorageOperationException is thrown when the local storage encounters some internal error during the attemt to process some operation.

```
#include <LocalStorageOperationException.h>
```

Inheritance diagram for quentier::local_storage::LocalStorageOperationException:



Collaboration diagram for quentier::local_storage::LocalStorageOperationException:



Public Member Functions

- **LocalStorageOperationException** (ErrorString message)
- LocalStorageOperationException ∗ **clone** () const override
- void **raise** () const override

Protected Member Functions

- QString exceptionDisplayName () const override

5.62.1 Detailed Description

The LocalStorageOperationException is thrown when the local storage encounters some internal error during the attemt to process some operation.

5.62.2 Member Function Documentation

5.62.2.1 exceptionDisplayName()

```
QString quentier::local_storage::LocalStorageOperationException::exceptionDisplayName ( )
const  [override], [protected], [virtual]
```

Implements quentier::IQuentierException.

5.63 quentier::LRUCache< Key, Value, Allocator > Class Template Reference

Public Types

- using **key_type** = Key
- using **mapped_type** = Value
- using **allocator_type** = Allocator
- using **value_type** = std::pair< key_type, mapped_type >
- using **container_type** = std::list< value_type, allocator_type >
- using **size_type** = typename container_type::size_type
- using **difference_type** = typename container_type::difference_type
- using **iterator** = typename container_type::iterator
- using **const_iterator** = typename container_type::const_iterator
- using **reverse_iterator** = std::reverse_iterator< iterator >
- using **const_reverse_iterator** = std::reverse_iterator< const_iterator >
- using **reference** = value_type &
- using **const_reference** = const value_type &
- using **pointer** = typename std::allocator_traits< allocator_type >::pointer
- using **const_pointer** = typename std::allocator_traits< allocator_type >::const_pointer

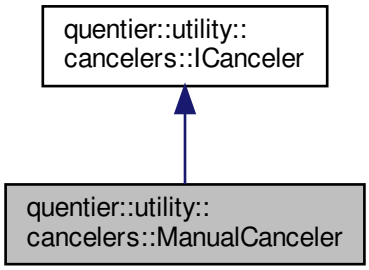
Public Member Functions

- **LRUCache** (const size_t maxSize=100)
- iterator **begin** () noexcept
- const_iterator **begin** () const noexcept
- reverse_iterator **rbegin** () noexcept
- const_reverse_iterator **rbegin** () const noexcept
- iterator **end** () noexcept
- const_iterator **end** () const noexcept
- reverse_iterator **rend** () noexcept
- const_reverse_iterator **rend** () const noexcept
- bool **empty** () const noexcept
- size_t **size** () const noexcept
- size_t **max_size** () const noexcept
- void **clear** ()
- void **put** (const key_type &key, const mapped_type &value)
- const mapped_type * **get** (const key_type &key) const noexcept
- bool **exists** (const key_type &key) const noexcept
- bool **remove** (const key_type &key) noexcept
- void **setMaxSize** (const size_t maxSize)

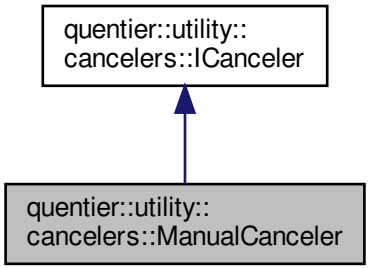
5.64 quentier::utility::cancelers::ManualCanceler Class Reference

`#include <ManualCanceler.h>`

Inheritance diagram for quentier::utility::cancelers::ManualCanceler:



Collaboration diagram for quentier::utility::cancelers::ManualCanceler:



Public Member Functions

- **ManualCanceler** (ManualCanceler &&other) noexcept
- ManualCanceler & **operator=** (ManualCanceler &&other) noexcept
- void cancel () noexcept
- bool isCanceled () const noexcept override

5.64.1 Detailed Description

ICanceler which allows one to manually call cancel method to cancel some task

5.64.2 Member Function Documentation

5.64.2.1 cancel()

```
void quentier::utility::cancelers::ManualCanceler::cancel ( )  [noexcept]
```

Manually cancel a task

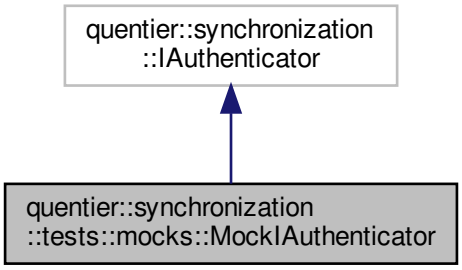
5.64.2.2 isCanceled()

```
bool quentier::utility::cancelers::ManualCanceler::isCanceled ( ) const  [override], [virtual],
[noexcept]
```

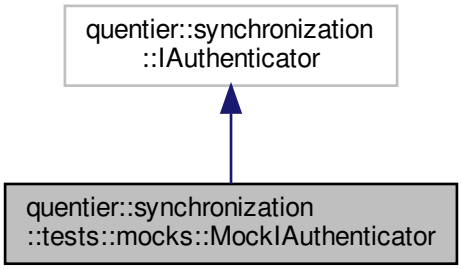
Implements quentier::utility::cancelers::ICanceler.

5.65 quentier::synchronization::tests::mocks::MockIAuthenticator Class Reference

Inheritance diagram for quentier::synchronization::tests::mocks::MockIAuthenticator:



Collaboration diagram for quentier::synchronization::tests::mocks::MockIAuthenticator:

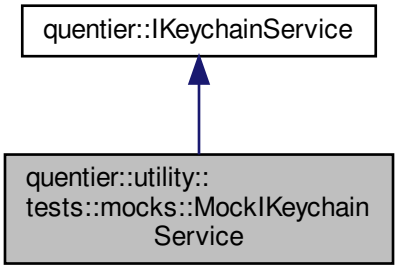


Public Member Functions

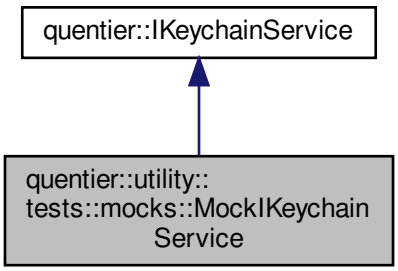
- **MOCK_METHOD** (QFuture< IAuthenticationInfoPtr >, authenticateNewAccount,(),(override))
- **MOCK_METHOD** (QFuture< IAuthenticationInfoPtr >, authenticateAccount,(Account account),(override))

5.66 quentier::utility::tests::mocks::MockIKeychainService Class Reference

Inheritance diagram for quentier::utility::tests::mocks::MockIKeychainService:



Collaboration diagram for quentier::utility::tests::mocks::MockIKeychainService:



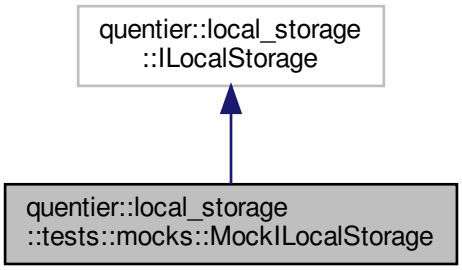
Public Member Functions

- **MOCK_METHOD** (QFuture< void >, writePassword,(QString service, QString key, QString password),(override))
- **MOCK_METHOD** (QFuture< QString >, readPassword,(QString service, QString key),(const, override))
- **MOCK_METHOD** (QFuture< void >, deletePassword,(QString service, QString key),(override))

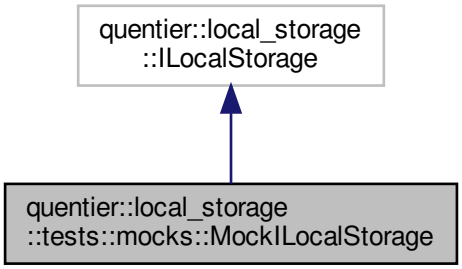
Additional Inherited Members

5.67 quentier::local_storage::tests::mocks::MockILocalStorage Class Reference

Inheritance diagram for quentier::local_storage::tests::mocks::MockILocalStorage:



Collaboration diagram for quentier::local_storage::tests::mocks::MockILocalStorage:



Public Member Functions

- **MOCK_METHOD** (QFuture< bool >, isVersionTooHigh,(),(const, override))
- **MOCK_METHOD** (QFuture< bool >, requiresUpgrade,(),(const, override))
- **MOCK_METHOD** (QFuture< QList< IPatchPtr > >, requiredPatches,(),(const, override))
- **MOCK_METHOD** (QFuture< qint32 >, version,(),(const, override))
- **MOCK_METHOD** (QFuture< qint32 >, highestSupportedVersion,(),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, userCount,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, putUser,(qevercloud::User user),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::User > >, findUserById,(qevercloud::UserID userId),(const, override))

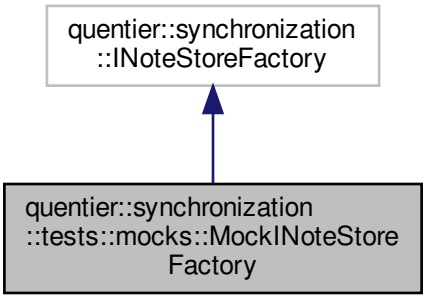
- **MOCK_METHOD** (QFuture< void >, expungeUserById,(qevercloud::UserID userId),(override))
- **MOCK_METHOD** (QFuture< quint32 >, notebookCount,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, putNotebook,(qevercloud::Notebook notebook),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Notebook > >, findNotebookByLocalId,(QString localId),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Notebook > >, findNotebookByGuid,(qevercloud::Guid guid),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Notebook > >, findNotebookByName,(QString name, std::optional< qevercloud::Guid > linkedNotebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Notebook > >, findDefaultNotebook,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeNotebookByLocalId,(QString localId),(override))
- **MOCK_METHOD** (QFuture< void >, expungeNotebookByGuid,(qevercloud::Guid guid),(override))
- **MOCK_METHOD** (QFuture< void >, expungeNotebookByName,(QString name, std::optional< qevercloud::Guid > linkedNotebookGuid),(override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Notebook > >, listNotebooks,(ListNotebooksOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::SharedNotebook > >, listSharedNotebooks,(qevercloud::Guid notebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< QSet< qevercloud::Guid > >, listNotebookGuids,(ListGuidsFilters filters, std::optional< qevercloud::Guid > linkedNotebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, linkedNotebookCount,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, putLinkedNotebook,(qevercloud::LinkedNotebook linkedNotebook),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::LinkedNotebook > >, findLinkedNotebookByGuid,(qevercloud::Guid guid),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeLinkedNotebookByGuid,(qevercloud::Guid guid),(override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::LinkedNotebook > >, listLinkedNotebooks,(ListLinkedNotebooksOptions options),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, noteCount,(NoteCountOptions options),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, noteCountPerNotebookLocalId,(QString notebookLocalId, NoteCountOptions options),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, noteCountPerTagLocalId,(QString tagLocalId, NoteCountOptions options),(const, override))
- **MOCK_METHOD** ((QFuture< QHash< QString, quint32 > >), noteCountsPerTags,(ListTagsOptions listTagsOptions, NoteCountOptions options),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, noteCountPerNotebookAndTagLocalIds,(QStringList notebookLocalIds, QStringList tagLocalIds, NoteCountOptions options),(const, override))
- **MOCK_METHOD** (QFuture< void >, putNote,(qevercloud::Note note),(override))
- **MOCK_METHOD** (QFuture< void >, updateNote,(qevercloud::Note note, UpdateNoteOptions options),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Note > >, findNoteByLocalId,(QString localId, FetchNoteOptions options),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Note > >, findNoteByGuid,(qevercloud::Guid guid, FetchNoteOptions options),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeNoteByLocalId,(QString localId),(override))
- **MOCK_METHOD** (QFuture< void >, expungeNoteByGuid,(qevercloud::Guid guid),(override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, listNotes,(FetchNoteOptions fetchOptions, ListNotesOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, listNotesPerNotebookLocalId,(QString notebookLocalId, FetchNoteOptions fetchOptions, ListNotesOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, listNotesPerTagLocalId,(QString tagLocalId, FetchNoteOptions fetchOptions, ListNotesOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, listNotesPerNotebookAndTagLocalIds,(QStringList notebookLocalIds, QStringList tagLocalIds, FetchNoteOptions fetchOptions, ListNotesOptions options),(const, override))

- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, listNotesByLocalIds,(QStringList noteLocalIds, FetchNoteOptions fetchOptions, ListNotesOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QSet< qevercloud::Guid > >, listNoteGuids,(ListGuidsFilters filters, std::optional< qevercloud::Guid > linkedNotebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Note > >, queryNotes,(NoteSearchQuery query, FetchNoteOptions fetchOptions),(const, override))
- **MOCK_METHOD** (QFuture< QStringList >, queryNoteLocalIds,(NoteSearchQuery query),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, tagCount,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, putTag,(qevercloud::Tag tag),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Tag > >, findTagByLocalId,(QString tagLocalId),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Tag > >, findTagByGuid,(qevercloud::Guid tagGuid),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Tag > >, findTagByName,(QString tagName, std::optional< QString > linkedNotebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Tag > >, listTags,(ListTagsOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::Tag > >, listTagsPerNoteLocalId,(QString noteLocalId, ListTagsOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QSet< qevercloud::Guid > >, listTagGuids,(ListGuidsFilters filters, std::optional< qevercloud::Guid > linkedNotebookGuid),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeTagByLocalId,(QString tagLocalId),(override))
- **MOCK_METHOD** (QFuture< void >, expungeTagByGuid,(qevercloud::Guid tagGuid),(override))
- **MOCK_METHOD** (QFuture< void >, expungeTagByName,(QString name, std::optional< qevercloud::Guid > linkedNotebookGuid),(override))
- **MOCK_METHOD** (QFuture< quint32 >, resourceCount,(NoteCountOptions options),(const, override))
- **MOCK_METHOD** (QFuture< quint32 >, resourceCountPerNoteLocalId,(QString noteLocalId),(const, override))
- **MOCK_METHOD** (QFuture< void >, putResource,(qevercloud::Resource resource),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Resource > >, findResourceByLocalId,(QString resourceLocalId, FetchResourceOptions options),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::Resource > >, findResourceByGuid,(qevercloud::Guid resourceGuid, FetchResourceOptions options),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeResourceByLocalId,(QString resourceLocalId),(override))
- **MOCK_METHOD** (QFuture< void >, expungeResourceByGuid,(qevercloud::Guid resourceGuid),(override))
- **MOCK_METHOD** (QFuture< quint32 >, savedSearchCount,(),(const, override))
- **MOCK_METHOD** (QFuture< void >, putSavedSearch,(qevercloud::SavedSearch search),(override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::SavedSearch > >, findSavedSearchByLocalId,(QString localId),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::SavedSearch > >, findSavedSearchByGuid,(qevercloud::Guid guid),(const, override))
- **MOCK_METHOD** (QFuture< std::optional< qevercloud::SavedSearch > >, findSavedSearchByName,(QString name),(const, override))
- **MOCK_METHOD** (QFuture< QList< qevercloud::SavedSearch > >, listSavedSearches,(ListSavedSearchesOptions options),(const, override))
- **MOCK_METHOD** (QFuture< QSet< qevercloud::Guid > >, listSavedSearchGuids,(ListGuidsFilters filters),(const, override))
- **MOCK_METHOD** (QFuture< void >, expungeSavedSearchByLocalId,(QString localId),(override))
- **MOCK_METHOD** (QFuture< void >, expungeSavedSearchByGuid,(qevercloud::Guid guid),(override))
- **MOCK_METHOD** (QFuture< qint32 >, highestUpdateSequenceNumber,(HighestUsnOption option),(const, override))
- **MOCK_METHOD** (QFuture< qint32 >, highestUpdateSequenceNumber,(qevercloud::Guid linkedNotebookGuid),(const, override))
- **MOCK_METHOD** (ILocalStorageNotifier *, notifier,(),(const, override))

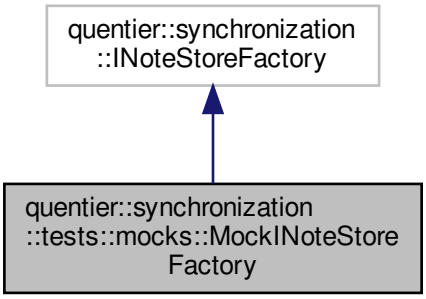
Additional Inherited Members

5.68 quentier::synchronization::tests::mocks::MockINoteStoreFactory Class Reference

Inheritance diagram for quentier::synchronization::tests::mocks::MockINoteStoreFactory:



Collaboration diagram for quentier::synchronization::tests::mocks::MockINoteStoreFactory:

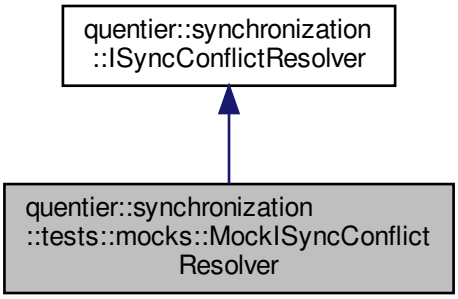


Public Member Functions

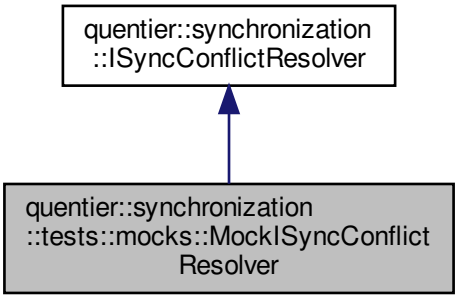
- **MOCK_METHOD** (::qevercloud::INoteStorePtr, createNoteStore,(QString noteStoreUrl, std::optional<↩ ::qevercloud::Guid > linkedNotebookGuid, ::qevercloud::IRequestContextPtr ctx, ::qevercloud::IRetryPolicy↩ Ptr retryPolicy),(override))

5.69 quentier::synchronization::tests::mocks::MockISyncConflict↩ Resolver Class Reference

Inheritance diagram for quentier::synchronization::tests::mocks::MockISyncConflictResolver:



Collaboration diagram for quentier::synchronization::tests::mocks::MockISyncConflictResolver:



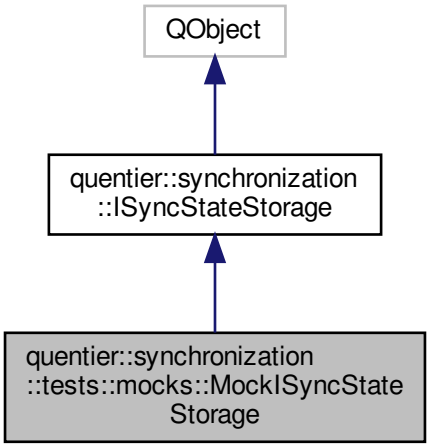
Public Member Functions

- **MOCK_METHOD** (QFuture< NotebookConflictResolution >, resolveNotebookConflict,(::qevercloud::↩ Notebook theirs, ::qevercloud::Notebook mine),(override))
- **MOCK_METHOD** (QFuture< NoteConflictResolution >, resolveNoteConflict,(::qevercloud::Note theirs, ↩ ::qevercloud::Note mine),(override))
- **MOCK_METHOD** (QFuture< SavedSearchConflictResolution >, resolveSavedSearchConflict,(↩ ::qevercloud::SavedSearch theirs, ::qevercloud::SavedSearch mine),(override))
- **MOCK_METHOD** (QFuture< TagConflictResolution >, resolveTagConflict,(::qevercloud::Tag theirs, ↩ ::qevercloud::Tag mine),(override))

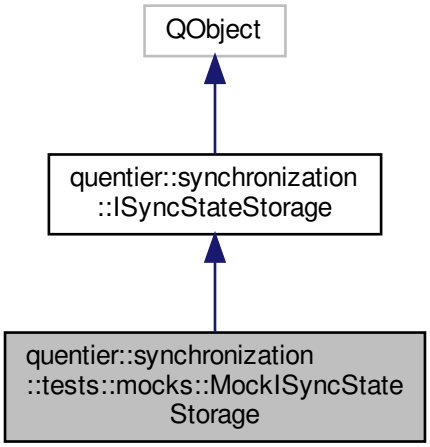
Additional Inherited Members

5.70 quentier::synchronization::tests::mocks::MockISyncStateStorage Class Reference

Inheritance diagram for quentier::synchronization::tests::mocks::MockISyncStateStorage:



Collaboration diagram for quentier::synchronization::tests::mocks::MockISyncStateStorage:



Public Member Functions

- **MOCK_METHOD** (ISyncStatePtr, getSyncState,(const Account &account),(override))
- **MOCK_METHOD** (void, setSyncState,(const Account &account, ISyncStatePtr syncState),(override))

Additional Inherited Members

5.71 quentier::synchronization::ISyncConflictResolver::ConflictResolution::MoveMine< T > Struct Template Reference

The MoveMine conflict resolution means "before using theirs version change mine version as specified". Note: the data item inside this conflict resolution might refer to something different than mine version passed to the conflict resolution function. It can be that way because the actual conflict might be with another local data item instead of the passed one.

`#include <ISyncConflictResolver.h>`

Public Types

- using **value_type** = T

Public Attributes

- T mine

5.71.1 Detailed Description

template<class T>
struct quentier::synchronization::ISyncConflictResolver::ConflictResolution::MoveMine< T >

The MoveMine conflict resolution means "before using theirs version change mine version as specified". Note: the data item inside this conflict resolution might refer to something different than mine version passed to the conflict resolution function. It can be that way because the actual conflict might be with another local data item instead of the passed one.

5.71.2 Member Data Documentation

5.71.2.1 mine

```
template<class T >
T quentier::synchronization::ISyncConflictResolver::ConflictResolution::MoveMine< T >::mine
```

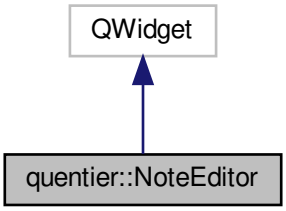
The changed value of mine data item.

5.72 quentier::NoteEditor Class Reference

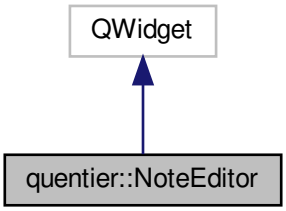
The NoteEditor class is a widget encapsulating all the functionality necessary for showing and editing notes.

`#include <NoteEditor.h>`

Inheritance diagram for quentier::NoteEditor:



Collaboration diagram for quentier::NoteEditor:



Public Slots

- void convertToNote ()
- void saveNoteToLocalStorage ()
- void setNoteTitle (const QString &noteTitle)
- void setTagIds (const QStringList &tagLocalIds, const QStringList &tagGuids)
- void **undo** ()
- void **redo** ()
- void **cut** ()
- void **copy** ()
- void **paste** ()
- void **pasteUnformatted** ()
- void **selectAll** ()
- void **formatSelectionAsSourceCode** ()

- void **fontMenu** ()
- void **textBold** ()
- void **textItalic** ()
- void **textUnderline** ()
- void **textStrikethrough** ()
- void **textHighlight** ()
- void **alignLeft** ()
- void **alignCenter** ()
- void **alignRight** ()
- void **alignFull** ()
- void **findNext** (const QString &text, bool matchCase) const
- void **findPrevious** (const QString &text, bool matchCase) const
- void **replace** (const QString &textToReplace, const QString &replacementText, bool matchCase)
- void **replaceAll** (const QString &textToReplace, const QString &replacementText, bool matchCase)
- void **insertToDoCheckbox** ()
- void **insertInAppNoteLink** (const QString &userId, const QString &shardId, const QString &noteGuid, const QString &linkText)
- void **setSpellcheck** (bool enabled)
- void **setFont** (const QFont &font)
- void **setFontHeight** (int height)
- void **setFontColor** (const QColor &color)
- void **setBackgroundColor** (const QColor &color)
- void setDefaultPalette (const QPalette &pal)
- void setDefaultFont (const QFont &font)
- void **insertHorizontalLine** ()
- void **increaseFontSize** ()
- void **decreaseFontSize** ()
- void **increaseIndentation** ()
- void **decreaseIndentation** ()
- void **insertBulletedList** ()
- void **insertNumberedList** ()
- void **insertTableDialog** ()
- void **insertFixedWidthTable** (int rows, int columns, int widthInPixels)
- void **insertRelativeWidthTable** (int rows, int columns, double relativeWidth)
- void **insertTableRow** ()
- void **insertTableColumn** ()
- void **removeTableRow** ()
- void **removeTableColumn** ()
- void **addAttachmentDialog** ()
- void **saveAttachmentDialog** (const QByteArray &resourceHash)
- void **saveAttachmentUnderCursor** ()
- void **openAttachment** (const QByteArray &resourceHash)
- void **openAttachmentUnderCursor** ()
- void **copyAttachment** (const QByteArray &resourceHash)
- void **copyAttachmentUnderCursor** ()
- void **encryptSelectedText** ()
- void **decryptEncryptedTextUnderCursor** ()
- void **editHyperlinkDialog** ()
- void **copyHyperlink** ()
- void **removeHyperlink** ()
- void **onNoteLoadCancelled** ()

Signals

- void **contentChanged** ()

  contentChanged signal is emitted when the note's content (text) gets modified via manual editing (i.e. not any action like paste or cut)
- void **noteAndNotebookFoundInLocalStorage** (qevercloud::Note note, qevercloud::Notebook notebook)

  noteAndNotebookFoundInLocalStorage signal is emitted when note and its corresponding notebook were found within the local storage right before the note editor starts to load the note into the editor
- void **noteNotFound** (QString noteLocalId)

  noteNotFound signal is emitted when the note could not be found within the local storage by the provided local id
- void **noteDeleted** (QString noteLocalId)

  noteDeleted signal is emitted when the note displayed within the note editor is deleted. The note editor stops displaying the note in this case shortly after emitting this signal
- void **noteModified** ()

  noteModified signal is emitted when the note's content within the editor gets modified via some way - either via manual editing or via some action (like paste or cut)
- void **notifyError** (ErrorString error)

  notifyError signal is emitted when NoteEditor encounters some problem worth letting the user to know about
- void **inAppNoteLinkClicked** (QString userId, QString shardId, QString noteGuid)

  inAppNoteLinkClicked signal is emitted when the in-app note link is clicked within the note editor
- void inAppNoteLinkPasteRequested (QString url, QString userId, QString shardId, QString noteGuid)
- void **convertedToNote** (qevercloud::Note note)
- void **cantConvertToNote** (ErrorString error)
- void **noteEditorHtmlUpdated** (QString html)
- void **currentNoteChanged** (qevercloud::Note note)
- void **spellCheckerNotReady** ()
- void **spellCheckerReady** ()
- void **noteLoaded** ()
- void **noteSavedToLocalStorage** (QString noteLocalId)

  noteSavedToLocalStorage signal is emitted when the note has been saved within the local storage. NoteEditor doesn't do this on its own unless it's explicitly asked to do this via invoking its saveNoteToLocalStorage slot
- void **failedToSaveNoteToLocalStorage** (ErrorString errorDescription, QString noteLocalId)

  failedToSaveNoteToLocalStorage signal is emitted in case of failure to save the note to local storage
- void **textBoldState** (bool state)
- void **textItalicState** (bool state)
- void **textUnderlineState** (bool state)
- void **textStrikethroughState** (bool state)
- void **textAlignLeftState** (bool state)
- void **textAlignCenterState** (bool state)
- void **textAlignRightState** (bool state)
- void **textAlignFullState** (bool state)
- void **textInsideOrderedListState** (bool state)
- void **textInsideUnorderedListState** (bool state)
- void **textInsideTableState** (bool state)
- void **textFontFamilyChanged** (QString fontFamily)
- void **textFontSizeChanged** (int fontSize)
- void **insertTableDialogRequested** ()

Public Member Functions

- **NoteEditor** (QWidget *parent=nullptr, Qt::WindowFlags flags={})
- void initialize (local_storage::ILocalStoragePtr localStorage, SpellChecker &spellChecker, const Account &account, QThread *pBackgroundJobsThread=nullptr)
- INoteEditorBackend * backend () noexcept
- void setBackend (INoteEditorBackend *backend)
- void setAccount (const Account &account)
- const QUndoStack * undoStack () const noexcept
- void setUndoStack (QUndoStack *pUndoStack)
- void setInitialPageHtml (const QString &html)
- void setNoteNotFoundPageHtml (const QString &html)
- void setNoteDeletedPageHtml (const QString &html)
- void setNoteLoadingPageHtml (const QString &html)
- QString currentNoteLocalId () const
- void setCurrentNoteLocalId (const QString &noteLocalId)
- void clear ()
- bool isModified () const noexcept
- bool isEditorPageModified () const noexcept
- bool isNoteLoaded () const noexcept
- qint64 idleTime () const noexcept
- void setFocus ()
- QString **selectedText** () const noexcept
- bool **hasSelection** () const noexcept
- bool **spellCheckEnabled** () const noexcept
- bool **print** (QPrinter &printer, ErrorString &errorDescription)
- bool **exportToPdf** (const QString &absoluteFilePath, ErrorString &errorDescription)
- bool **exportToEnex** (const QStringList &tagNames, QString &enex, ErrorString &errorDescription)
- QPalette defaultPalette () const
- const QFont * defaultFont () const

Protected Member Functions

- void **dragMoveEvent** (QDragMoveEvent *pEvent) override
- void **dropEvent** (QDropEvent *pEvent) override

5.72.1 Detailed Description

The NoteEditor class is a widget encapsulating all the functionality necessary for showing and editing notes.

5.72.2 Member Function Documentation

5.72.2.1 backend()

```
INoteEditorBackend * quentier::NoteEditor::backend ( )  [noexcept]
```

Returns

the pointer to the note editor's backend

5.72.2.2 clear()

```
void quentier::NoteEditor::clear ( )
```

Clear the contents of the note editor

5.72.2.3 convertToNote

```
void quentier::NoteEditor::convertToNote ( )  [slot]
```

Invoke this slot to launch the asynchronous procedure of converting the current contents of the note editor to note; the convertedToNote signal would be emitted in response when the conversion is done

5.72.2.4 currentNoteLocalId()

```
QString quentier::NoteEditor::currentNoteLocalId ( ) const
```

Get the local id of the note currently set to the note editor

5.72.2.5 defaultFont()

```
const QFont * quentier::NoteEditor::defaultFont ( ) const
```

Returns

pointer to the default font used by the note editor; if no such font was set to the editor previously, returns null pointer

5.72.2.6 defaultPalette()

```
QPalette quentier::NoteEditor::defaultPalette ( ) const
```

Returns

palette containing default colors used by the editor; the palette is composed of colors from note editor widget's native palette but some of them might be overridden by colors from the palette specified previously via set↩ DefaultPalette method: those colors from the specified palette which were valid

5.72.2.7 idleTime()

```
qint64 quentier::NoteEditor::idleTime ( ) const  [noexcept]
```

Returns

the number of milliseconds since the last user's interaction with the note editor or -1 if there was no interaction or if no note is loaded at the moment

5.72.2.8 inAppNoteLinkPasteRequested

```
void quentier::NoteEditor::inAppNoteLinkPasteRequested (
            QString url,
            QString userId,
            QString shardId,
            QString noteGuid )  [signal]
```

inAppNoteLinkPasteRequested signal is emitted when the note editor detects the attempt to paste the in-app note link into the note editor; the link would not be inserted right away, instead this signal would be emitted. Whatever party managing the note editor is expected to connect some slot to this signal and provide the optionally amended link information to the note editor by sending the signal connected to its insertInAppNoteLink slot - this slot accepts both the URL of the link and the link text and performs the actual link insertion into the note. If the link text is empty, the URL itself is used as the link text.

5.72.2.9 initialize()

```
void quentier::NoteEditor::initialize (
            local_storage::ILocalStoragePtr localStorage,
            SpellChecker & spellChecker,
            const Account & account,
            QThread * pBackgroundJobsThread = nullptr )
```

NoteEditor requires LocalStorageManagerAsync, SpellChecker and Account for its work but due to the particularities of Qt's .ui files processing these can't be passed right inside the constructor, hence here's a special initialization method

Parameters

localStorage	Local storage
spellChecker	Spell checker to be used by note editor
account	Current account
pBackgroundJobsThread	Pointer to the thread to be used for scheduling of background jobs of NoteEditor; if null, NoteEditor's background jobs would take place in GUI thread

5.72.2.10 isEditorPageModified()

```
bool quentier::NoteEditor::isEditorPageModified ( ) const  [noexcept]
```

Returns

true if there's content within the editor not yet converted to note, false otherwise

5.72.2.11 isModified()

```
bool quentier::NoteEditor::isModified ( ) const  [noexcept]
```

Returns

true if there's content within the editor not yet converted to note or not saved to local storage, false otherwise

5.72.2.12 isNoteLoaded()

```
bool quentier::NoteEditor::isNoteLoaded ( ) const  [noexcept]
```

Returns

true if the note last set to the editor has been fully loaded already, false otherwise

5.72.2.13 saveNoteToLocalStorage

```
void quentier::NoteEditor::saveNoteToLocalStorage ( )  [slot]
```

Invoke this slot to launch the asynchronous procedure of saving the modified current note back to the local storage. If no note is set to the editor or if the note is not modified, no action would be performed. Otherwise noteSaved↩ ToLocalStorage signal would be emitted in case of successful saving or failedToSaveNoteToLocalStorage would be emitted otherwise

5.72.2.14 setAccount()

```
void quentier::NoteEditor::setAccount (
            const Account & account )
```

Set the current account to the note editor

5.72.2.15 setBackend()

```
void quentier::NoteEditor::setBackend (
            INoteEditorBackend * backend )
```

This method can be used to set the backend to the note editor; the note editor has the default backend so this method is not obligatory to be called

5.72.2.16 setCurrentNoteLocalId()

```
void quentier::NoteEditor::setCurrentNoteLocalId (
            const QString & noteLocalId )
```

Set note local id to the note editor. The note is being searched for within the local storage, in case of no note being found noteNotFound signal is emitted. Otherwise note editor page starts loading.

Parameters

noteLocalId	The local id of note

5.72.2.17 setDefaultFont

```
void quentier::NoteEditor::setDefaultFont (
            const QFont & font )  [slot]
```

Sets the font which would be used by the editor by default

Parameters

font	The font to be used by the editor by default

5.72.2.18 setDefaultPalette

```
void quentier::NoteEditor::setDefaultPalette (
            const QPalette & pal )  [slot]
```

Sets the palette with colors to be used by the editor. New colors are applied after the note is fully loaded. If no note is set to the editor, the palette is simply remembered for the next note to be loaded into it.

Colors within the palette and their usage:

1. WindowText - used as default font color
2. Base - used as default background color
3. HighlightedText - used as font color for selected text
4. Highlight - used as background color for selected text

Parameters

pal	The palette to be set. Invalid colors from it are substituted by colors from widget's palette by the editor

5.72.2.19 setFocus()

```
void quentier::NoteEditor::setFocus ( )
```

Sets the focus to the backend note editor widget

5.72.2.20 setInitialPageHtml()

```
void quentier::NoteEditor::setInitialPageHtml (
            const QString & html )
```

Set the html to be displayed when the note is not set to the editor

5.72.2.21 setNoteDeletedPageHtml()

```
void quentier::NoteEditor::setNoteDeletedPageHtml (
            const QString & html )
```

Set the html to be displayed when the note set to the editor was deleted from the local storage (either marked as deleted or deleted permanently i.e. expunged)

5.72.2.22 setNoteLoadingPageHtml()

```
void quentier::NoteEditor::setNoteLoadingPageHtml (
            const QString & html )
```

Set the html to be displayed when the note set to the editor is being loaded into it

5.72.2.23 setNoteNotFoundPageHtml()

```
void quentier::NoteEditor::setNoteNotFoundPageHtml (
            const QString & html )
```

Set the html to be displayed when the note attempted to be set to the editor was not found within the local storage

5.72.2.24 setNoteTitle

```
void quentier::NoteEditor::setNoteTitle (
            const QString & noteTitle )  [slot]
```

Invoke this slot to set the title to the note displayed via the note editor. The note editor itself doesn't manage the note title in any way so any external code using the note editor can set the title to the note editor's note which would be considered modified if the title is new and then eventually the note would be saved to local storage

Parameters

noteTitle	The title of the note

5.72.2.25 setTagIds

```
void quentier::NoteEditor::setTagIds (
            const QStringList & tagLocalIds,
            const QStringList & tagGuids )  [slot]
```

Invoke this slot to set tag local ids and/or tag guids to the note displayed via the note editor. The note editor itself doesn't manage the note tags in any way so any external code using the note editor can set the tag ids to the note editor's internal note which would be considered modified if the tag ids are new and then eventually the note would be saved to local storage

Parameters

tagLocalIds	The list of tag local ids for the note
tagGuids	The list of tag guids for the note

5.72.2.26 setUndoStack()

```
void quentier::NoteEditor::setUndoStack (
            QUndoStack * pUndoStack )
```

Set the undo stack for the note editor to use

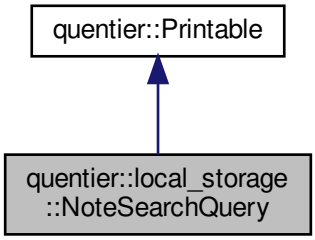
5.72.2.27 undoStack()

```
const QUndoStack * quentier::NoteEditor::undoStack ( ) const  [noexcept]
```

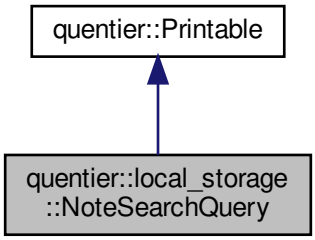
Get the undo stack serving to the note editor

5.73 quentier::local_storage::NoteSearchQuery Class Reference

Inheritance diagram for quentier::local_storage::NoteSearchQuery:



Collaboration diagram for quentier::local_storage::NoteSearchQuery:



Public Member Functions

- **NoteSearchQuery** (const NoteSearchQuery &other)
- **NoteSearchQuery** (NoteSearchQuery &&other) noexcept
- NoteSearchQuery & **operator=** (const NoteSearchQuery &other)
- NoteSearchQuery & **operator=** (NoteSearchQuery &&other) noexcept
- bool **isEmpty** () const
- void **clear** ()
- QString queryString () const
- bool **setQueryString** (const QString &queryString, ErrorString &error)
- QString notebookModifier () const
- bool **hasAnyModifier** () const
- const QStringList & **tagNames** () const
- const QStringList & **negatedTagNames** () const
- bool **hasAnyTag** () const
- bool **hasNegatedAnyTag** () const
- const QStringList & **titleNames** () const
- const QStringList & **negatedTitleNames** () const
- bool **hasAnyTitleName** () const
- bool **hasNegatedAnyTitleName** () const
- const QList< qint64 > & **creationTimestamps** () const
- const QList< qint64 > & **negatedCreationTimestamps** () const
- bool **hasAnyCreationTimestamp** () const
- bool **hasNegatedAnyCreationTimestamp** () const
- const QList< qint64 > & **modificationTimestamps** () const
- const QList< qint64 > & **negatedModificationTimestamps** () const
- bool **hasAnyModificationTimestamp** () const
- bool **hasNegatedAnyModificationTimestamp** () const
- const QStringList & **resourceMimeTypes** () const
- const QStringList & **negatedResourceMimeTypes** () const
- bool **hasAnyResourceMimeType** () const
- bool **hasNegatedAnyResourceMimeType** () const
- const QList< qint64 > & **subjectDateTimestamps** () const
- const QList< qint64 > & **negatedSubjectDateTimestamps** () const
- bool **hasAnySubjectDateTimestamp** () const
- bool **hasNegatedAnySubjectDateTimestamp** () const
- const QList< double > & **latitudes** () const
- const QList< double > & **negatedLatitudes** () const
- bool **hasAnyLatitude** () const
- bool **hasNegatedAnyLatitude** () const
- const QList< double > & **longitudes** () const
- const QList< double > & **negatedLongitudes** () const
- bool **hasAnyLongitude** () const
- bool **hasNegatedAnyLongitude** () const
- const QList< double > & **altitudes** () const
- const QList< double > & **negatedAltitudes** () const
- bool **hasAnyAltitude** () const
- bool **hasNegatedAnyAltitude** () const
- const QStringList & **authors** () const
- const QStringList & **negatedAuthors** () const
- bool **hasAnyAuthor** () const
- bool **hasNegatedAnyAuthor** () const
- const QStringList & **sources** () const
- const QStringList & **negatedSources** () const
- bool **hasAnySource** () const

- bool **hasNegatedAnySource** () const
- const QStringList & **sourceApplications** () const
- const QStringList & **negatedSourceApplications** () const
- bool **hasAnySourceApplication** () const
- bool **hasNegatedAnySourceApplication** () const
- const QStringList & **contentClasses** () const
- const QStringList & **negatedContentClasses** () const
- bool **hasAnyContentClass** () const
- bool **hasNegatedAnyContentClass** () const
- const QStringList & **placeNames** () const
- const QStringList & **negatedPlaceNames** () const
- bool **hasAnyPlaceName** () const
- bool **hasNegatedAnyPlaceName** () const
- const QStringList & **applicationData** () const
- const QStringList & **negatedApplicationData** () const
- bool **hasAnyApplicationData** () const
- bool **hasNegatedAnyApplicationData** () const
- const QList< qint64 > & **reminderOrders** () const
- const QList< qint64 > & **negatedReminderOrders** () const
- bool **hasAnyReminderOrder** () const
- bool **hasNegatedAnyReminderOrder** () const
- const QList< qint64 > & **reminderTimes** () const
- const QList< qint64 > & **negatedReminderTimes** () const
- bool **hasAnyReminderTime** () const
- bool **hasNegatedAnyReminderTime** () const
- const QList< qint64 > & **reminderDoneTimes** () const
- const QList< qint64 > & **negatedReminderDoneTimes** () const
- bool **hasAnyReminderDoneTime** () const
- bool **hasNegatedAnyReminderDoneTime** () const
- bool **hasUnfinishedToDo** () const
- bool **hasNegatedUnfinishedToDo** () const
- bool **hasFinishedToDo** () const
- bool **hasNegatedFinishedToDo** () const
- bool **hasAnyToDo** () const
- bool **hasNegatedAnyToDo** () const
- bool **hasEncryption** () const
- bool **hasNegatedEncryption** () const
- const QStringList & **contentSearchTerms** () const
- const QStringList & **negatedContentSearchTerms** () const
- bool **hasAnyContentSearchTerms** () const
- bool **isMatcheable** () const
- QTextStream & print (QTextStream &strm) const override

5.73.1 Member Function Documentation

5.73.1.1 notebookModifier()

```
QString quentier::local_storage::NoteSearchQuery::notebookModifier ( ) const
```

If query string has "notebook:<notebook name>" scope modifier, this method returns the name of the notebook, otherwise it returns empty string

5.73.1.2 print()

```
QTextStream & quentier::local_storage::NoteSearchQuery::print (
            QTextStream & strm ) const  [override], [virtual]
```

Implements quentier::Printable.

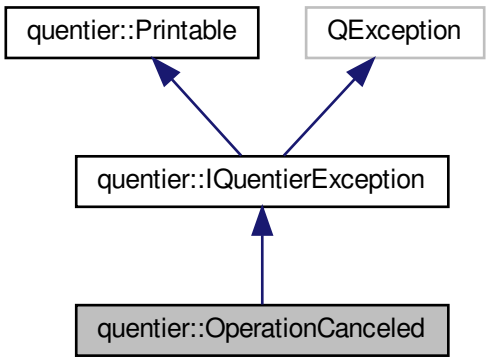
5.73.1.3 queryString()

```
QString quentier::local_storage::NoteSearchQuery::queryString ( ) const
```

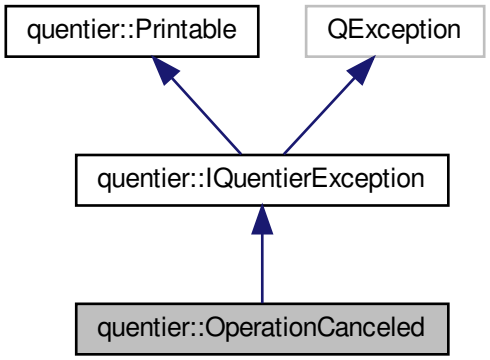
Returns the original non-parsed query string

5.74 quentier::OperationCanceled Class Reference

Inheritance diagram for quentier::OperationCanceled:



Collaboration diagram for quentier::OperationCanceled:



Public Member Functions

- OperationCanceled * **clone** () const override
- void **raise** () const override

Protected Member Functions

- QString exceptionDisplayName () const override

5.74.1 Member Function Documentation

5.74.1.1 exceptionDisplayName()

```
QString quentier::OperationCanceled::exceptionDisplayName ( ) const  [override], [protected],
[virtual]
```

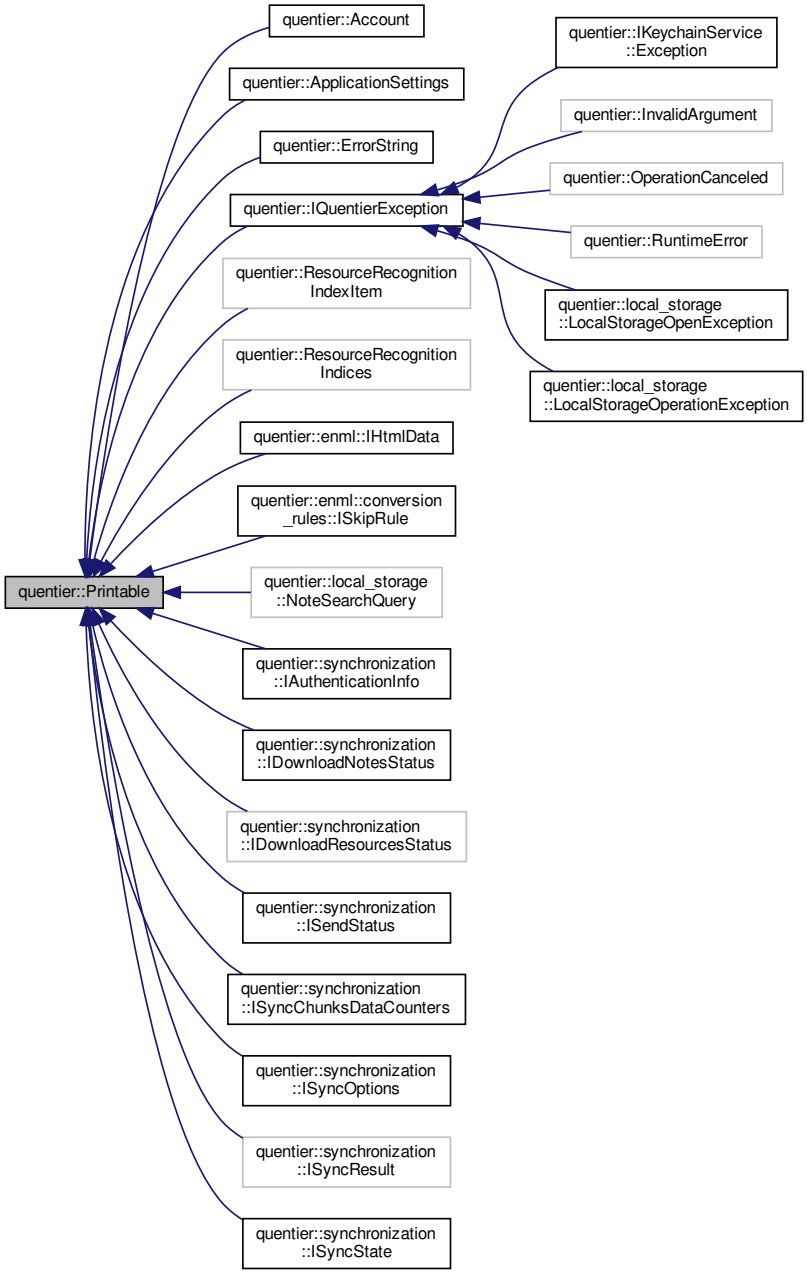
Implements quentier::IQuentierException.

5.75 quentier::Printable Class Reference

The Printable class is the interface for Quentier's internal classes which should be able to write themselves into QTextStream and/or convert to QString.

`#include <Printable.h>`

Inheritance diagram for quentier::Printable:



Public Member Functions

- virtual QTextStream & **print** (QTextStream &strm) const =0
- QString **toString** () const

Friends

- QUENTIER_EXPORT QTextStream & **operator<<** (QTextStream &strm, const Printable &printable)
- QUENTIER_EXPORT QDebug & **operator<<** (QDebug &debug, const Printable &printable)

5.75.1 Detailed Description

The Printable class is the interface for Quentier's internal classes which should be able to write themselves into QTextStream and/or convert to QString.

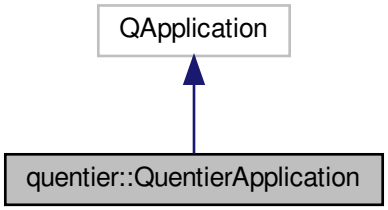
5.76 QPromise< T > Class Template Reference

Public Member Functions

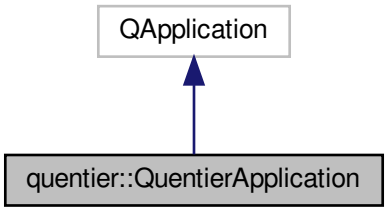
- **QPromise** (QPromise< T > &&other) noexcept
- **QPromise** (QFutureInterface< T > &other)
- QPromise & **operator=** (QPromise< T > &&other) noexcept
- QFuture< T > **future** () const
- template<typename U , typename = std::enable_if_t< std::is_same_v<U, T> || std::is_convertible_v<U, T>>>
  void **addResult** (U &&result, int index=-1)
- void **setException** (const QException &e)
- void **start** ()
- void **finish** ()
- void **suspendIfRequested** ()
- bool **isCanceled** () const
- void **setProgressRange** (int minimum, int maximum)
- void **setProgressValue** (int progressValue)
- void **setProgressValueAndText** (int progressValue, const QString &progressText)
- void **swap** (QPromise< T > &other) noexcept

5.77 quentier::QuentierApplication Class Reference

Inheritance diagram for quentier::QuentierApplication:



Collaboration diagram for quentier::QuentierApplication:



Public Member Functions

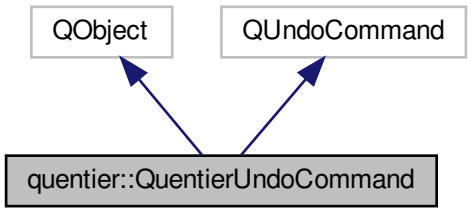
- **QuentierApplication** (int &argc, char ∗argv[ ])
- bool **notify** (QObject ∗pObject, QEvent ∗pEvent) override
- bool **event** (QEvent ∗pEvent) override

5.78 quentier::QuentierUndoCommand Class Reference

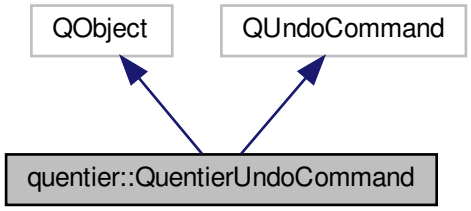
The QuentierUndoCommand class has the sole purpose of working around one quirky aspect of Qt's undo/redo framework: when you push QUndoCommand to QUndoStack, it calls "redo" method of that command. This class offers subclasses to implement their own methods for actual "undo" and "redo" commands while ignoring the attempts to "redo" anything if there were no previous "undo" call prior to that.

`#include <QuentierUndoCommand.h>`

Inheritance diagram for quentier::QuentierUndoCommand:



Collaboration diagram for quentier::QuentierUndoCommand:



Signals

- void **notifyError** (ErrorString error)

Public Member Functions

- **QuentierUndoCommand** (QUndoCommand *parent=nullptr)
- **QuentierUndoCommand** (const QString &text, QUndoCommand *parent=nullptr)
- void **undo** () final
- void **redo** () final
- bool **onceUndoExecuted** () const noexcept

Protected Member Functions

- virtual void **undoImpl** ()=0
- virtual void **redoImpl** ()=0

5.78.1 Detailed Description

The QuentierUndoCommand class has the sole purpose of working around one quirky aspect of Qt's undo/redo framework: when you push QUndoCommand to QUndoStack, it calls "redo" method of that command. This class offers subclasses to implement their own methods for actual "undo" and "redo" commands while ignoring the attempts to "redo" anything if there were no previous "undo" call prior to that.

The rationale behind the current behaviour seems to be the compliance with "command pattern behaviour" when you create the command to execute the action instead of just executing it immediately. This design is enforced by Qt's undo/redo framework, there's no option to choose not to call "redo" when pushing to the stack.

One thing which this design fails to see is the fact that the command may be already executed externally by the moment the QUndoCommand can be created. Suppose we can get the information about how to undo (and then again redo) that command. We create the corresponding QUndoCommand, set up the stuff for its undo/redo methods and push it to QUndoStack for future use... But at the same time QUndoStack calls "redo" method of the command. Really not the behaviour you'd like to have.

QuentierUndoCommand is also QObject, it is for error reporting via notifyError signal

5.79 quentier::synchronization::RateLimitReachedError Struct Reference

`#include <Errors.h>`

Public Attributes

- std::optional< qint32 > rateLimitDurationSec

5.79.1 Detailed Description

Information about "API rate limit reached" error which Evernote servers might return if too much of their API calls were made recently. In case of such error synchronization should be repeated later, after some time passes.

5.79.2 Member Data Documentation

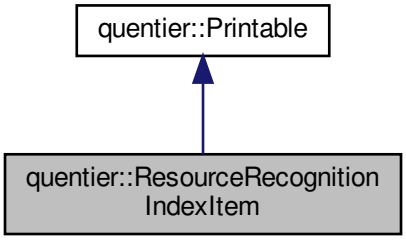
5.79.2.1 rateLimitDurationSec

`std::optional<qint32> quentier::synchronization::RateLimitReachedError::rateLimitDurationSec`

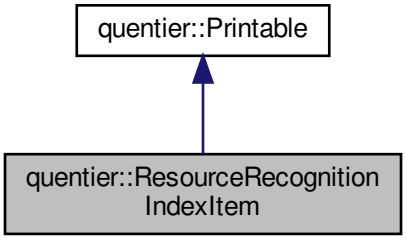
Number of seconds during which since the current moment during which any call to Evernote API would again result in "API rate limit reached" error i.e. the number of seconds to wait for before the next attempt to run synchronization

5.80 quentier::ResourceRecognitionIndexItem Class Reference

Inheritance diagram for quentier::ResourceRecognitionIndexItem:



Collaboration diagram for quentier::ResourceRecognitionIndexItem:



Classes

- struct IBarcodeItem
- struct IObjectItem
- struct IShapeItem
- struct ITextItem

Public Types

- using **ITextItemPtr** = std::shared_ptr< ITextItem >
- using **IObjectItemPtr** = std::shared_ptr< IObjectItem >
- using **IShapeItemPtr** = std::shared_ptr< IShapeItem >
- using **IBarcodeItemPtr** = std::shared_ptr< IBarcodeItem >

Public Member Functions

- **ResourceRecognitionIndexItem** (const ResourceRecognitionIndexItem &other)
- **ResourceRecognitionIndexItem** (ResourceRecognitionIndexItem &&other) noexcept
- ResourceRecognitionIndexItem & **operator=** (const ResourceRecognitionIndexItem &other)
- ResourceRecognitionIndexItem & **operator=** (ResourceRecognitionIndexItem &&other) noexcept
- bool **isValid** () const
- int **x** () const
- void **setX** (int x)
- int **y** () const
- void **setY** (int y)
- int **h** () const
- void **setH** (int h)
- int **w** () const
- void **setW** (int w)
- int **offset** () const
- void **setOffset** (int offset)
- int **duration** () const
- void **setDuration** (int duration)
- QList< int > **strokes** () const
- void **setStrokes** (QList< int > strokes)

- QList< ITextItemPtr > **textItems** () const
- void **setTextItems** (QList< ITextItemPtr > textItems)
- QList< IObjectItemPtr > **objectItems** () const
- void **setObjectItems** (QList< IObjectItemPtr > objectItems)
- QList< IShapeItemPtr > **shapeItems** () const
- void **setShapeItems** (QList< IShapeItemPtr > shapeItems)
- QList< IBarcodeItemPtr > **barcodeItems** () const
- void **setBarcodeItems** (QList< IBarcodeItemPtr > barcodeItems)
- QTextStream & print (QTextStream &strm) const override

5.80.1 Member Function Documentation

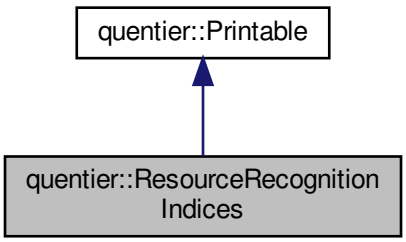
5.80.1.1 print()

```
QTextStream & quentier::ResourceRecognitionIndexItem::print (
            QTextStream & strm ) const  [override], [virtual]
```

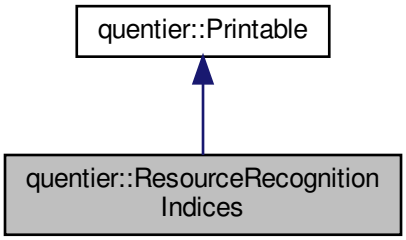
Implements quentier::Printable.

5.81 quentier::ResourceRecognitionIndices Class Reference

Inheritance diagram for quentier::ResourceRecognitionIndices:



Collaboration diagram for quentier::ResourceRecognitionIndices:



Public Member Functions

- **ResourceRecognitionIndices** (const QByteArray &rawRecognitionIndicesData)
- **ResourceRecognitionIndices** (const ResourceRecognitionIndices &other)
- **ResourceRecognitionIndices** (ResourceRecognitionIndices &&other) noexcept
- ResourceRecognitionIndices & **operator=** (const ResourceRecognitionIndices &other)
- ResourceRecognitionIndices & **operator=** (ResourceRecognitionIndices &&other) noexcept
- bool **isNull** () const
- bool **isValid** () const
- QString **objectId** () const
- QString **objectType** () const
- QString **recoType** () const
- QString **engineVersion** () const
- QString **docType** () const
- QString **lang** () const
- int **objectHeight** () const
- int **objectWidth** () const
- QVector< ResourceRecognitionIndexItem > **items** () const
- bool **setData** (const QByteArray &rawRecognitionIndicesData)
- QTextStream & print (QTextStream &strm) const override

5.81.1 Member Function Documentation

5.81.1.1 print()

```
QTextStream & quentier::ResourceRecognitionIndices::print (
            QTextStream & strm ) const  [override], [virtual]
```

Implements quentier::Printable.

5.82 quentier::Result< T, Error, typename > Class Template Reference

The Result template class represents the bare bones result monad implementation which either contains some valid value or an error.

`#include <Result.h>`

Public Member Functions

- template<typename T1 = T, typename std::enable_if_t<!std::is_void_v< std::decay_t< T1 > > > * = nullptr> **Result** (T1 t)
- **Result** (Error error)
- **Result** (const Result< T, Error > &other)
- **Result** (Result< T, Error > &&other)
- Result & **operator=** (const Result< T, Error > &other)
- Result & **operator=** (Result< T, Error > &&other)
- bool isValid () const noexcept
- **operator bool** () const noexcept
- template<typename T1 = T, typename std::enable_if_t<!std::is_void_v< std::decay_t< T1 > > > * = nullptr> T1 & **get** ()
- template<typename T1 = T, typename std::enable_if_t<!std::is_void_v< std::decay_t< T1 > > > * = nullptr> const T1 & **get** () const
- template<typename T1 = T, typename std::enable_if_t<!std::is_void_v< std::decay_t< T1 > > > * = nullptr> T1 & **operator*** ()
- template<typename T1 = T, typename std::enable_if_t<!std::is_void_v< std::decay_t< T1 > > > * = nullptr> const T1 & **operator*** () const
- const Error & **error** () const
- Error & **error** ()

5.82.1 Detailed Description

template<class T, class Error, typename = typename std::enable_if_t< !std::is_same_v<std::decay_t<T>, std::decay_t<Error>>>>
class quentier::Result< T, Error, typename >

The Result template class represents the bare bones result monad implementation which either contains some valid value or an error.

5.82.2 Member Function Documentation

5.82.2.1 isValid()

```
template<class T , class Error , typename = typename std::enable_if_t< !std::is_same_v<std::decay_t<T>, std::decay_t<Error>>>>
bool quentier::Result< T, Error, typename >::isValid ( ) const  [inline], [noexcept]
```

Returns

boolean value indicating whether the result contains a value

5.83 quentier::threading::detail::ResultTypeHelper< F, Arg, Enable > Struct Template Reference

5.84 quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFuture< Arg > > > > Struct Template Reference

Public Types

- using **ResultType** = std::invoke_result_t< std::decay_t< F >, std::decay_t< Arg > >

5.85 quentier::threading::detail::ResultTypeHelper< F, Arg, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFuture< Arg > > > > Struct Template Reference

Public Types

- using **ResultType** = std::invoke_result_t< std::decay_t< F >, QFuture< Arg > >

5.86 quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< !std::is_invocable_v< std::decay_t< F >, QFuture< void > > > > Struct Template Reference

Public Types

- using **ResultType** = std::invoke_result_t< std::decay_t< F > >

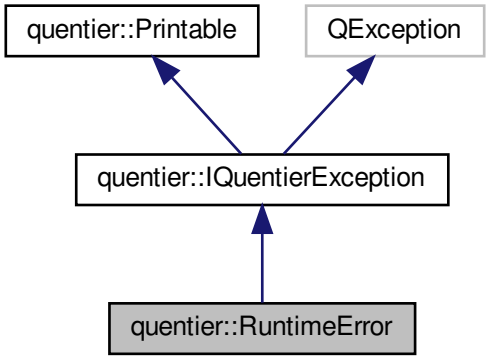
5.87 quentier::threading::detail::ResultTypeHelper< F, void, typename std::enable_if_t< std::is_invocable_v< std::decay_t< F >, QFuture< void > > > > Struct Template Reference

Public Types

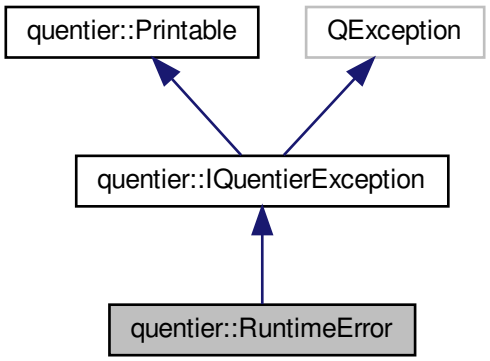
- using **ResultType** = std::invoke_result_t< std::decay_t< F >, QFuture< void > >

5.88 quentier::RuntimeError Class Reference

Inheritance diagram for quentier::RuntimeError:



Collaboration diagram for quentier::RuntimeError:



Public Member Functions

- **RuntimeError** (ErrorString message)
- RuntimeError * **clone** () const override
- void **raise** () const override

Protected Member Functions

- QString exceptionDisplayName () const override

5.88.1 Member Function Documentation

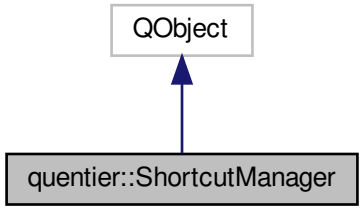
5.88.1.1 exceptionDisplayName()

```
QString quentier::RuntimeError::exceptionDisplayName ( ) const  [override], [protected], [virtual]
```

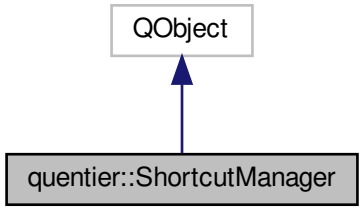
Implements quentier::IQuentierException.

5.89 quentier::ShortcutManager Class Reference

Inheritance diagram for quentier::ShortcutManager:



Collaboration diagram for quentier::ShortcutManager:



Public Types

- enum **QuentierShortcutKey** {
  NewNote = 5000 , **NewTag** , **NewNotebook** , **NewSavedSearch** ,
  AddAttachment , **SaveAttachment** , **OpenAttachment** , **CopyAttachment** ,
  CutAttachment , **RemoveAttachment** , **RenameAttachment** , **AddAccount** ,
  ExitAccount , **SwitchAccount** , **AccountInfo** , **NoteSearch** ,
  NewNoteSearch , **ShowNotes** , **ShowNotebooks** , **ShowTags** ,
  ShowSavedSearches , **ShowDeletedNotes** , **ShowStatusBar** , **ShowToolBar** ,
  PasteUnformatted , **Font** , **UpperIndex** , **LowerIndex** ,
  AlignLeft , **AlignCenter** , **AlignRight** , **AlignFull** ,
  IncreaseIndentation , **DecreaseIndentation** , **IncreaseFontSize** , **DecreaseFontSize** ,
  InsertNumberedList , **InsertBulletedList** , **Strikethrough** , **Highlight** ,
  InsertTable , **InsertRow** , **InsertColumn** , **RemoveRow** ,
  RemoveColumn , **InsertHorizontalLine** , **InsertToDoTag** , **EditHyperlink** ,
  CopyHyperlink , **RemoveHyperlink** , **Encrypt** , **Decrypt** ,
  DecryptPermanently , **BackupLocalStorage** , **RestoreLocalStorage** , **UpgradeLocalStorage** ,
  LocalStorageStatus , **SpellCheck** , **SpellCheckIgnoreWord** , **SpellCheckAddWordToUserDictionary** ,
  SaveImage , **AnnotateImage** , **ImageRotateClockwise** , **ImageRotateCounterClockwise** ,
  Synchronize , **FullSync** , **ImportFolders** , **Preferences** ,
  ReleaseNotes , **ViewLogs** , **About** , **UnknownKey** = 100000 }

Public Slots

- void **setUserShortcut** (int key, QKeySequence shortcut, const Account &account, QString context={})
- void **setNonStandardUserShortcut** (QString nonStandardKey, QKeySequence shortcut, const Account &account, QString context={})
- void **setDefaultShortcut** (int key, QKeySequence shortcut, const Account &account, QString context={})
- void **setNonStandardDefaultShortcut** (QString nonStandardKey, QKeySequence shortcut, const Account &account, QString context={})

Signals

- void **shortcutChanged** (int key, QKeySequence shortcut, const Account &account, QString context)
- void **nonStandardShortcutChanged** (QString nonStandardKey, QKeySequence shortcut, const Account &account, QString context)

Public Member Functions

- **ShortcutManager** (QObject *parent=nullptr)
- QKeySequence shortcut (int key, const Account &account, const QString &context={}) const
- QKeySequence shortcut (const QString &nonStandardKey, const Account &account, const QString &context={}) const
- QKeySequence defaultShortcut (int key, const Account &account, const QString &context={}) const
- QKeySequence defaultShortcut (const QString &nonStandardKey, const Account &account, const QString &context={}) const
- QKeySequence userShortcut (int key, const Account &account, const QString &context={}) const
- QKeySequence userShortcut (const QString &nonStandardKey, const Account &account, const QString &context={}) const

5.89.1 Member Function Documentation

5.89.1.1 defaultShortcut() [1/2]

```
QKeySequence quentier::ShortcutManager::defaultShortcut (
            const QString & nonStandardKey,
            const Account & account,
            const QString & context = {} ) const
```

Returns

Default shortcut for the non-standard key if present, otherwise empty key sequence

5.89.1.2 defaultShortcut() [2/2]

```
QKeySequence quentier::ShortcutManager::defaultShortcut (
            int key,
            const Account & account,
            const QString & context = {} ) const
```

Returns

Default shortcut for the standard key if present, otherwise empty key sequence

5.89.1.3 shortcut() [1/2]

```
QKeySequence quentier::ShortcutManager::shortcut (
            const QString & nonStandardKey,
            const Account & account,
            const QString & context = {} ) const
```

Returns

Active shortcut for the non-standard key - either the user defined shortcut (if present) or the default one (if present as well)

5.89.1.4 shortcut() [2/2]

```
QKeySequence quentier::ShortcutManager::shortcut (
            int key,
            const Account & account,
            const QString & context = {} ) const
```

Returns

Active shortcut for the standard key - either the user defined shortcut (if present) or the default one (if present as well)

5.89.1.5 userShortcut() [1/2]

```
QKeySequence quentier::ShortcutManager::userShortcut (
             const QString & nonStandardKey,
             const Account & account,
             const QString & context = {} ) const
```

Returns

User defined shortcut for the non-standard key if present, otherwise empty key sequence

5.89.1.6 userShortcut() [2/2]

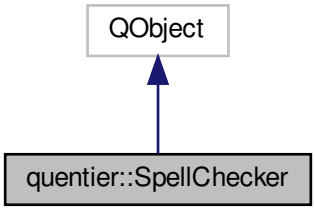
```
QKeySequence quentier::ShortcutManager::userShortcut (
             int key,
             const Account & account,
             const QString & context = {} ) const
```

Returns

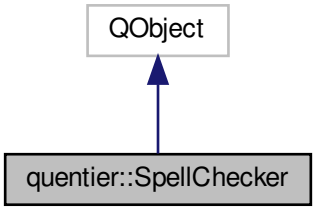
User defined shortcut for the standard key if present, otherwise empty key sequence

5.90 quentier::SpellChecker Class Reference

Inheritance diagram for quentier::SpellChecker:



Collaboration diagram for quentier::SpellChecker:



Signals

- void **ready** ()

Public Member Functions

- **SpellChecker** (FileIOProcessorAsync *fileIOProcessorAsync, Account account, QObject *parent=nullptr, const QString &userDictionaryPath={})
- QList< std::pair< QString, bool > > **listAvailableDictionaries** () const
- void **setAccount** (const Account &account)
- void **enableDictionary** (const QString &language)
- void **disableDictionary** (const QString &language)
- bool **checkSpell** (const QString &word) const
- QStringList **spellCorrectionSuggestions** (const QString &misSpelledWord) const
- void **addToUserWordlist** (const QString &word)
- void **removeFromUserWordList** (const QString &word)
- void **ignoreWord** (const QString &word)
- void **removeWord** (const QString &word)
- bool **isReady** () const noexcept

5.91 quentier::StringUtils Class Reference

Public Member Functions

- void **removePunctuation** (QString &str, const QList< QChar > &charactersToPreserve={}) const
- void **removeDiacritics** (QString &str) const
- void **removeNewlines** (QString &str) const

5.92 quentier::SysInfo Class Reference

Public Member Functions

- qint64 **pageSize** ()
- qint64 **totalMemory** ()
- qint64 **freeMemory** ()
- QString **stackTrace** ()
- QString **platformName** ()

5.93 quentier::threading::TrackedTask< LockableObject, Function > Class Template Reference

`#include <TrackedTask.h>`

Public Member Functions

- template<typename SomeLockableObject , typename SomeFunction >
  constexpr **TrackedTask** (SomeLockableObject &&someLockableObject, SomeFunction &&function)
- template<typename... Arguments, typename = std::enable_if_t< std::is_invocable_v<Function, Arguments...> || std::is_member_function_pointer_v<Function>>>
  constexpr void **operator()** (Arguments &&... arguments)
- template<typename... Arguments, typename = std::enable_if_t< std::is_invocable_v<Function, Arguments...> || std::is_member_function_pointer_v<Function>>>
  constexpr void **operator()** (Arguments &&... arguments) const

5.93.1 Detailed Description

template<typename LockableObject, typename Function>
class quentier::threading::TrackedTask< LockableObject, Function >

Wrapper class which automates checking for the state of a lockable object. With this class code like this

auto task = [selfWeak = weak_from_this()] { auto self = selfWeak.lock(); if (!self) { return; } // otherwise do something };

can be written like this:

auto task = threading::TrackedTask{weak_from_this(), &MyClass::someMethod};

5.94 quentier::UidGenerator Class Reference

Static Public Member Functions

- static QString **Generate** ()
- static QString **UidToString** (const QUuid &uid)

5.95 quentier::synchronization::ISyncConflictResolver::ConflictResolution::UseMine Struct Reference

The UseMine conflict resolution means "override theirs version with mine version".

`#include <ISyncConflictResolver.h>`

5.95.1 Detailed Description

The UseMine conflict resolution means "override theirs version with mine version".

5.96 quentier::synchronization::ISyncConflictResolver::ConflictResolution::UseTheirs Struct Reference

The UseTheirs conflict resolution means "override mine version with theirs version".

`#include <ISyncConflictResolver.h>`

5.96.1 Detailed Description

The UseTheirs conflict resolution means "override mine version with theirs version".

Chapter 6

File Documentation

6.1 ISkipRule.h

```
1 /*
2 * Copyright 2023-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/enml/conversion_rules/MatchMode.h>
22 #include <quentier/utility/Printable.h>
23
24 #include <QtGlobal>
25
26 namespace quentier::enml::conversion_rules {
27
37 class ISkipRule :  public Printable
38 {
39 public:
40     ~ISkipRule() override;
41
45     enum class Target
46     {
50         Element,
54         AttibuteName,
58         AttributeValue
59     };
60
61     friend QUENTIER_EXPORT QTextStream & operator«(
62         QTextStream & strm, Target target);
63
64     friend QUENTIER_EXPORT QDebug & operator«(QDebug & dbg, Target target);
65
69     [[nodiscard]] virtual Target target() const = 0;
70
74     [[nodiscard]] virtual QString value() const = 0;
75
79     [[nodiscard]] virtual MatchMode matchMode() const = 0;
80
85     [[nodiscard]] virtual bool includeContents() const = 0;
86
90     [[nodiscard]] virtual Qt::CaseSensitivity caseSensitivity() const = 0;
91
92 public:  // Printable
93     QTextStream & print(QTextStream & strm) const override;
94 };
95
96 } // namespace quentier::enml::conversion_rules
```

6.2 ISkipRuleBuilder.h

```
1 /*
2 * Copyright 2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/enml/conversion_rules/Fwd.h>
22 #include <quentier/enml/conversion_rules/ISkipRule.h>
23 #include <quentier/utility/Linkage.h>
24
25 namespace quentier::enml::conversion_rules {
26
27 class QUENTIER_EXPORT ISkipRuleBuilder
28 {
29 public:
30     virtual ~ISkipRuleBuilder();
31
32     virtual ISkipRuleBuilder & setTarget(ISkipRule::Target target) = 0;
33     virtual ISkipRuleBuilder & setValue(QString value) = 0;
34     virtual ISkipRuleBuilder & setMatchMode(MatchMode matchMode) = 0;
35     virtual ISkipRuleBuilder & setIncludeContents(bool includeContents) = 0;
36     virtual ISkipRuleBuilder & setCaseSensitivity(
37         Qt::CaseSensitivity caseSensitivity) = 0;
38
39     [[nodiscard]] virtual ISkipRulePtr build() = 0;
40 };
41
42 } // namespace quentier::enml::conversion_rules
```

6.3 MatchMode.h

```
1 /*
2 * Copyright 2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 class QDebug;
24 class QTextStream;
25
26 namespace quentier::enml::conversion_rules {
27
31 enum class MatchMode
32 {
36     Equals,
40     StartsWith,
44     EndsWith,
48     Contains
49 };
50
51 QUENTIER_EXPORT QTextStream & operator<<(
```

```
52     QTextStream & strm, MatchMode matchMode);
53
54 QUENTIER_EXPORT QDebug & operator<<(QDebug & dbg, MatchMode matchMode);
55
56 } // namespace quentier::enml::conversion_rules
```

6.4 HtmlUtils.h

```
1 /*
2 * Copyright 2023-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/ErrorString.h>
22 #include <quentier/types/Result.h>
23 #include <quentier/utility/Linkage.h>
24
25 #include <QFlags>
26
27 #include <memory>
28
29 namespace quentier::enml::utils {
30
31 [[nodiscard]] Result<QString, ErrorString> QUENTIER_EXPORT
32     convertHtmlToXml(const QString & html);
33
34 [[nodiscard]] Result<QString, ErrorString> QUENTIER_EXPORT
35     convertHtmlToXhtml(const QString & html);
36
37 [[nodiscard]] Result<QString, ErrorString> QUENTIER_EXPORT
38     cleanupHtml(const QString & html);
39
40 enum class EscapeStringOption
41 {
42     Simplify = 1 << 0,
43 };
44
45 Q_DECLARE_FLAGS(EscapeStringOptions, EscapeStringOption);
46
47 [[nodiscard]] QString QUENTIER_EXPORT htmlEscapeString(
48     QString str, EscapeStringOptions options = EscapeStringOptions{});
49
50 } // namespace quentier::enml::utils
```

6.5 IConverter.h

```
1 /*
2 * Copyright 2023-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
```

```
19 #pragma once
20
21 #include <quentier/enml/Fwd.h>
22 #include <quentier/enml/conversion_rules/Fwd.h>
23 #include <quentier/types/ErrorString.h>
24 #include <quentier/types/Result.h>
25 #include <quentier/utility/Linkage.h>
26
27 #include <QList>
28 #include <QStringList>
29 #include <QTextDocument>
30
31 #include <qevercloud/types/Note.h>
32
33 namespace quentier::enml {
34
39 class QUENTIER_EXPORT IConverter
40 {
41 public:
42     virtual ~IConverter();
43
52     [[nodiscard]] virtual Result<QString, ErrorString> convertHtmlToEnml(
53         const QString & html, IDecryptedTextCache & decryptedTextCache,
54         const QList<conversion_rules::ISkipRulePtr> & skipRules = {}) const = 0;
55
64     [[nodiscard]] virtual Result<void, ErrorString> convertHtmlToDoc(
65         const QString & html, QTextDocument & doc,
66         const QList<conversion_rules::ISkipRulePtr> & skipRules = {}) const = 0;
67
74     [[nodiscard]] virtual Result<QString, ErrorString> convertHtmlToXml(
75         const QString & html) const = 0;
76
83     [[nodiscard]] virtual Result<QString, ErrorString> convertHtmlToXhtml(
84         const QString & html) const = 0;
85
93     [[nodiscard]] virtual Result<IHtmlDataPtr, ErrorString> convertEnmlToHtml(
94         const QString & enml,
95         IDecryptedTextCache & decryptedTextCache) const = 0;
96
103     [[nodiscard]] virtual Result<QString, ErrorString> convertEnmlToPlainText(
104         const QString & enml) const = 0;
105
112     [[nodiscard]] virtual Result<QStringList, ErrorString>
113         convertEnmlToWordsList(const QString & enml) const = 0;
114
120     [[nodiscard]] virtual QStringList convertPlainTextToWordsList(
121         const QString & plainText) const = 0;
122
129     [[nodiscard]] virtual Result<void, ErrorString> validateEnml(
130         const QString & enml) const = 0;
131
138     [[nodiscard]] virtual Result<QString, ErrorString> validateAndFixupEnml(
139         const QString & enml) const = 0;
140
145     enum class EnexExportTags
146     {
147         Yes = 0,
148         No
149     };
150
160     [[nodiscard]] virtual Result<QString, ErrorString> exportNotesToEnex(
161         const QList<qevercloud::Note> & notes,
162         const QHash<QString, QString> & tagNamesByTagLocalIds,
163         EnexExportTags exportTagsOption,
164         const QString & version = {}) const = 0;
165
174     [[nodiscard]] virtual Result<QList<qevercloud::Note>, ErrorString>
175         importEnex(const QString & enex) const = 0;
176 };
177
178 } // namespace quentier::enml
```

6.6 IDecryptedTextCache.h

```
1 /*
2 * Copyright 2016-2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
```

```
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QString>
24
25 #include <cstddef>
26 #include <optional>
27 #include <utility>
28
29 class QDebug;
30 class QTextStream;
31
32 namespace quentier::enml {
33
34 class QUENTIER_EXPORT IDecryptedTextCache
35 {
36 public:
37     virtual ~IDecryptedTextCache();
38
39     enum class RememberForSession
40     {
41         Yes,
42         No
43     };
44
45     friend QUENTIER_EXPORT QDebug & operator«(
46         QDebug & dbg, RememberForSession rememberForSession);
47
48     friend QUENTIER_EXPORT QTextStream & operator«(
49         QTextStream & strm, RememberForSession rememberForSession);
50
51     virtual void addDecryptexTextInfo(
52         const QString & encryptedText, const QString & decryptedText,
53         const QString & passphrase, const QString & cipher,
54         std::size_t keyLength, RememberForSession rememberForSession) = 0;
55
56     [[nodiscard]] virtual std::optional<std::pair<QString, RememberForSession»
57         findDecryptedTextInfo(const QString & encryptedText) const = 0;
58
59     [[nodiscard]] virtual std::optional<QString> updateDecryptedTextInfo(
60         const QString & originalEncryptedText,
61         const QString & newDecryptedText) = 0;
62
63     virtual void removeDecryptedTextInfo(const QString & encryptedText) = 0;
64     virtual void clearNonRememberedForSessionEntries() = 0;
65 };
66
67 } // namespace quentier::enml
```

6.7 IENMLTagsConverter.h

```
1 /*
2 * Copyright 2023-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/ErrorString.h>
22 #include <quentier/types/Result.h>
```

```
23 #include <quentier/utility/Linkage.h>
24
25 #include <qevercloud/types/Fwd.h>
26
27 #include <QString>
28 #include <QtGlobal>
29
30 #include <cstddef>
31
32 namespace quentier::enml {
33
39 class QUENTIER_EXPORT IENMLTagsConverter
40 {
41 public:
42     virtual ~IENMLTagsConverter();
43
51     [[nodiscard]] virtual QString convertEnToDo(
52         bool checked, quint32 index) const = 0;
53
64     [[nodiscard]] virtual QString convertEncryptedText(
65         const QString & encryptedText, const QString & hint,
66         const QString & cipher, std::size_t keyLength, quint32 index) const = 0;
67
79     [[nodiscard]] virtual QString convertDecryptedText(
80         const QString & decryptedText, const QString & encryptedText,
81         const QString & hint, const QString & cipher, std::size_t keyLength,
82         quint32 index) const = 0;
83
90     [[nodiscard]] virtual Result<QString, ErrorString> convertResource(
91         const qevercloud::Resource & resource) const = 0;
92 };
93
94 } // namespace quentier::enml
```

6.8 IHtmlData.h

```
1 /*
2 * Copyright 2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22 #include <quentier/utility/Printable.h>
23
24 #include <QString>
25 #include <QtGlobal>
26
27 namespace quentier::enml {
28
33 struct QUENTIER_EXPORT IHtmlData :  public Printable
34 {
38     [[nodiscard]] virtual QString html() const = 0;
39
43     [[nodiscard]] virtual quint32 numEnToDoNodes() const = 0;
44
48     [[nodiscard]] virtual quint32 numHyperlinkNodes() const = 0;
49
53     [[nodiscard]] virtual quint32 numEnCryptNodes() const = 0;
54
58     [[nodiscard]] virtual quint32 numEnDecryptedNodes() const = 0;
59
60 public:  // Printable
61     QTextStream & print(QTextStream & strm) const override;
62 };
63
64 } // namespace quentier::enml
```

6.9 InvalidArgument.h

```
1 /*
2 * Copyright 2021 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/exception/IQuentierException.h>
22
23 namespace quentier {
24
25 class QUENTIER_EXPORT InvalidArgument :  public IQuentierException
26 {
27 public:
28     explicit InvalidArgument(ErrorString message);
29
30     [[nodiscard]] InvalidArgument * clone() const override;
31     void raise() const override;
32
33 protected:
34     [[nodiscard]] QString exceptionDisplayName() const override;
35 };
36
37 } // namespace quentier
```

6.10 IQuentierException.h

```
1 /*
2 * Copyright 2016-2021 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #ifndef LIB_QUENTIER_EXCEPTION_I_QUENTIER_EXCEPTION_H
20 #define LIB_QUENTIER_EXCEPTION_I_QUENTIER_EXCEPTION_H
21
22 #include <quentier/types/ErrorString.h>
23 #include <quentier/utility/Printable.h>
24
25 #include <QException>
26
27 namespace quentier {
28
37 class QUENTIER_EXPORT IQuentierException :  public Printable, public QException
38 {
39 public:
40     ~IQuentierException() noexcept override;
41
42     [[nodiscard]] ErrorString errorMessage() const;
43     [[nodiscard]] QString localizedErrorMessage() const;
44     [[nodiscard]] QString nonLocalizedErrorMessage() const;
45
46     [[nodiscard]] const char * what() const noexcept override;
47
48     QTextStream & print(QTextStream & strm) const override;
49
```

```
50 protected:
51     explicit IQuentierException(ErrorString message);
52     IQuentierException(const IQuentierException & other);
53     IQuentierException & operator=(const IQuentierException & other);
54
55     [[nodiscard]] virtual QString exceptionDisplayName() const = 0;
56
57 private:
58     ErrorString m_message;
59     char * m_whatMessage = nullptr;
60 };
61
62 } // namespace quentier
63
64 #endif // LIB_QUENTIER_EXCEPTION_I_QUENTIER_EXCEPTION_H
```

6.11 OperationCanceled.h

```
1 /*
2 * Copyright 2022 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/exception/IQuentierException.h>
22
23 namespace quentier {
24
25 class QUENTIER_EXPORT OperationCanceled :  public IQuentierException
26 {
27 public:
28     explicit OperationCanceled();
29
30     [[nodiscard]] OperationCanceled * clone() const override;
31     void raise() const override;
32
33 protected:
34     [[nodiscard]] QString exceptionDisplayName() const override;
35 };
36
37 } // namespace quentier
```

6.12 RuntimeError.h

```
1 /*
2 * Copyright 2021 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/exception/IQuentierException.h>
```

```
22
23 namespace quentier {
24
25 class QUENTIER_EXPORT RuntimeError :  public IQuentierException
26 {
27 public:
28     explicit RuntimeError(ErrorString message);
29     ~RuntimeError() noexcept override;
30
31     [[nodiscard]] RuntimeError * clone() const override;
32     void raise() const override;
33
34 protected:
35     [[nodiscard]] QString exceptionDisplayName() const override;
36 };
37
38 } // namespace quentier
```

6.13 enml/conversion_rules/Factory.h

```
1 /*
2 * Copyright 2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/enml/conversion_rules/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 namespace quentier::enml::conversion_rules {
25
26 [[nodiscard]] QUENTIER_EXPORT ISkipRuleBuilderPtr createSkipRuleBuilder();
27
28 } // namespace quentier::enml::conversion_rules
```

6.14 enml/Factory.h

```
1 /*
2 * Copyright 2023-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/enml/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 namespace quentier::enml {
25
29 [[nodiscard]] QUENTIER_EXPORT IDecryptedTextCachePtr createDecryptedTextCache();
30
34 [[nodiscard]] QUENTIER_EXPORT IENMLTagsConverterPtr createEnmlTagsConverter();
```

```
35
43 [[nodiscard]] QUENTIER_EXPORT IConverterPtr
44     createConverter(IENMLTagsConverterPtr enmlTagsConverter = nullptr);
45
46 } // namespace quentier::enml
```

6.15 local_storage/Factory.h

```
1 /*
2 * Copyright 2023-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/local_storage/Fwd.h>
22 #include <quentier/threading/Fwd.h>
23 #include <quentier/types/Fwd.h>
24 #include <quentier/utility/Linkage.h>
25
26 #include <QtGlobal>
27
28 class QDir;
29
30 namespace quentier::local_storage {
31
32 [[nodiscard]] QUENTIER_EXPORT ILocalStoragePtr createSqliteLocalStorage(
33     const Account & account, const QDir & localStorageDir,
34     threading::QThreadPtr thread = {});
35
36 } // namespace quentier::local_storage
```

6.16 synchronization/Factory.h

```
1 /*
2 * Copyright 2023-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/local_storage/Fwd.h>
22 #include <quentier/synchronization/Fwd.h>
23 #include <quentier/threading/Fwd.h>
24 #include <quentier/utility/Fwd.h>
25 #include <quentier/utility/Linkage.h>
26
27 #include <qevercloud/Fwd.h>
28
29 #include <QString>
30 #include <QUrl>
31 #include <QtGlobal>
32
```

```
33 class QWidget;
34
35 namespace quentier::synchronization {
36
37 [[nodiscard]] QUENTIER_EXPORT IAuthenticatorPtr createQEverCloudAuthenticator(
38     QString consumerKey, QString consumerSecret, QUrl serverUrl,
39     threading::QThreadPtr uiThread, QWidget * parentWidget = nullptr);
40
41 [[nodiscard]] QUENTIER_EXPORT ISynchronizerPtr createSynchronizer(
42     const QUrl & userStoreUrl, IAuthenticatorPtr authenticator,
43     ISyncStateStoragePtr syncStateStorage = nullptr,
44     IKeychainServicePtr keychainService = nullptr,
45     INoteStoreFactoryPtr noteStoreFactory = nullptr,
46     IUserStoreFactoryPtr userStoreFactory = nullptr,
47     qevercloud::IRequestContextPtr ctx = nullptr,
48     qevercloud::IRetryPolicyPtr retryPolicy = nullptr);
49
50 [[nodiscard]] QUENTIER_EXPORT ISyncConflictResolverPtr
51     createSimpleSyncConflictResolver(
52         local_storage::ILocalStoragePtr localStorage);
53
54 [[nodiscard]] QUENTIER_EXPORT ISyncStateStoragePtr
55     createSyncStateStorage(QObject * parent = nullptr);
56
57 } // namespace quentier::synchronization
```

6.17 threading/Factory.h

```
1 /*
2 * Copyright 2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/threading/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 namespace quentier::threading {
25
29 [[nodiscard]] QUENTIER_EXPORT QThreadPoolPtr globalThreadPool();
30
31 } // namespace quentier::threading
```

6.18 enml/conversion_rules/Fwd.h

```
1 /*
2 * Copyright 2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
```

```
21 #include <memory>
22
23 namespace quentier::enml::conversion_rules {
24
25 class ISkipRule;
26 using ISkipRulePtr = std::shared_ptr<ISkipRule>;
27
28 class ISkipRuleBuilder;
29 using ISkipRuleBuilderPtr = std::shared_ptr<ISkipRuleBuilder>;
30
31 } // namespace quentier::enml::conversion_rules
```

6.19 enml/Fwd.h

```
1 /*
2 * Copyright 2016-2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <memory>
22
23 namespace quentier::enml {
24
25 class IConverter;
26 using IConverterPtr = std::shared_ptr<IConverter>;
27
28 class IDecryptedTextCache;
29 using IDecryptedTextCachePtr = std::shared_ptr<IDecryptedTextCache>;
30
31 class IENMLTagsConverter;
32 using IENMLTagsConverterPtr = std::shared_ptr<IENMLTagsConverter>;
33
34 struct IHtmlData;
35 using IHtmlDataPtr = std::shared_ptr<IHtmlData>;
36
37 } // namespace quentier::enml
```

6.20 local_storage/Fwd.h

```
1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <memory>
22
23 namespace quentier::local_storage {
24
25 class ILocalStorage;
```

```
26 using ILocalStoragePtr = std::shared_ptr<ILocalStorage>;
27
28 class ILocalStorageNotifier;
29
30 class IPatch;
31 using IPatchPtr = std::shared_ptr<IPatch>;
32
33 class NoteSearchQuery;
34
35 } // namespace quentier::local_storage
```

6.21 synchronization/Fwd.h

```
1 /*
2 * Copyright 2020-2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <memory>
22
23 namespace quentier::synchronization {
24
25 class IAuthenticator;
26 using IAuthenticatorPtr = std::shared_ptr<IAuthenticator>;
27
28 class INoteStoreFactory;
29 using INoteStoreFactoryPtr = std::shared_ptr<INoteStoreFactory>;
30
31 class ISyncConflictResolver;
32 using ISyncConflictResolverPtr = std::shared_ptr<ISyncConflictResolver>;
33
34 class ISynchronizer;
35 using ISynchronizerPtr = std::shared_ptr<ISynchronizer>;
36
37 class ISyncEventsNotifier;
38
39 class ISyncOptions;
40 using ISyncOptionsPtr = std::shared_ptr<ISyncOptions>;
41
42 class ISyncStateStorage;
43 using ISyncStateStoragePtr = std::shared_ptr<ISyncStateStorage>;
44
45 class IUserStoreFactory;
46 using IUserStoreFactoryPtr = std::shared_ptr<IUserStoreFactory>;
47
48 } // namespace quentier::synchronization
```

6.22 synchronization/types/Fwd.h

```
1 /*
2 * Copyright 2022 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
```

```
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <memory>
22
23 namespace quentier::synchronization {
24
25 class IAuthenticationInfo;
26 using IAuthenticationInfoPtr = std::shared_ptr<IAuthenticationInfo>;
27
28 class IAuthenticationInfoBuilder;
29 using IAuthenticationInfoBuilderPtr =
30     std::shared_ptr<IAuthenticationInfoBuilder>;
31
32 class IDownloadNotesStatus;
33 using IDownloadNotesStatusPtr = std::shared_ptr<IDownloadNotesStatus>;
34
35 class IDownloadResourcesStatus;
36 using IDownloadResourcesStatusPtr = std::shared_ptr<IDownloadResourcesStatus>;
37
38 class ISendStatus;
39 using ISendStatusPtr = std::shared_ptr<ISendStatus>;
40
41 struct ISyncChunksDataCounters;
42 using ISyncChunksDataCountersPtr = std::shared_ptr<ISyncChunksDataCounters>;
43
44 class ISyncOptions;
45 using ISyncOptionsPtr = std::shared_ptr<ISyncOptions>;
46
47 class ISyncOptionsBuilder;
48 using ISyncOptionsBuilderPtr = std::shared_ptr<ISyncOptionsBuilder>;
49
50 class ISyncResult;
51 using ISyncResultPtr = std::shared_ptr<ISyncResult>;
52
53 class ISyncState;
54 using ISyncStatePtr = std::shared_ptr<ISyncState>;
55
56 class ISyncStateBuilder;
57 using ISyncStateBuilderPtr = std::shared_ptr<ISyncStateBuilder>;
58
59 } // namespace quentier::synchronization
```

6.23 threading/Fwd.h

```
1 /*
2 * Copyright 2022 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <memory>
22
23 class QMutex;
24 class QThread;
25 class QThreadPool;
26
27 namespace quentier::threading {
28
29 using QMutexPtr = std::shared_ptr<QMutex>;
30 using QThreadPtr = std::shared_ptr<QThread>;
31 using QThreadPoolPtr = std::shared_ptr<QThreadPool>;
32
33 } // namespace quentier::threading
```

6.24 types/Fwd.h

```
1 /*
2 * Copyright 2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 namespace quentier {
22
23 class Account;
24 class ErrorString;
25 class ResourceRecognitionIndexItem;
26 class ResourceRecognitionIndices;
27
28 } // namespace quentier
```

6.25 utility/cancelers/Fwd.h

```
1 /*
2 * Copyright 2022 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <memory>
22
23 namespace quentier::utility::cancelers {
24
25 class AnyOfCanceler;
26 using AnyOfCancelerPtr = std::shared_ptr<AnyOfCanceler>;
27
28 class ICanceler;
29 using ICancelerPtr = std::shared_ptr<ICanceler>;
30
31 class ManualCanceler;
32 using ManualCancelerPtr = std::shared_ptr<ManualCanceler>;
33
34 } // namespace quentier::utility::cancelers
```

6.26 utility/Fwd.h

```
1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
```

```
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <memory>
22
23 namespace quentier {
24
25 class IKeychainService;
26 using IKeychainServicePtr = std::shared_ptr<IKeychainService>;
27
28 } // namespace quentier
```

6.27 ILocalStorage.h

```
1 /*
2 * Copyright 2020-2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/local_storage/Fwd.h>
22 #include <quentier/local_storage/NoteSearchQuery.h>
23 #include <quentier/types/Fwd.h>
24 #include <quentier/utility/Linkage.h>
25
26 #include <qevercloud/types/LinkedNotebook.h>
27 #include <qevercloud/types/Note.h>
28 #include <qevercloud/types/Notebook.h>
29 #include <qevercloud/types/Resource.h>
30 #include <qevercloud/types/SavedSearch.h>
31 #include <qevercloud/types/SharedNotebook.h>
32 #include <qevercloud/types/Tag.h>
33 #include <qevercloud/types/User.h>
34
35 #include <QFlags>
36 #include <QFuture>
37 #include <QHash>
38 #include <QList>
39 #include <QStringList>
40 #include <QThreadPool>
41
42 #include <optional>
43 #include <utility>
44
45 class QDebug;
46 class QTextStream;
47 class QThreadPool;
48
49 namespace quentier::local_storage {
50
51 class QUENTIER_EXPORT ILocalStorage
52 {
53 public:
54     virtual ~ILocalStorage() = default;
55
56 public:
57     enum class StartupOption
58     {
59         ClearDatabase = 1 << 1,
60         OverrideLock = 1 << 2
61     };
```

```
62      Q_DECLARE_FLAGS(StartupOptions, StartupOption);
63
64      friend QUENTIER_EXPORT QTextStream & operator«(
65          QTextStream & strm, StartupOption option);
66
67      friend QUENTIER_EXPORT QDebug & operator«(
68          QDebug & dbg, StartupOption option);
69
70      friend QUENTIER_EXPORT QTextStream & operator«(
71          QTextStream & strm, StartupOptions options);
72
73      friend QUENTIER_EXPORT QDebug & operator«(
74          QDebug & dbg, StartupOptions options);
75
77
78      enum class ListObjectsFilter
79      {
80          Include,
81          Exclude
82      };
83
84      friend QUENTIER_EXPORT QTextStream & operator«(
85          QTextStream & strm, ListObjectsFilter filter);
86
87      friend QUENTIER_EXPORT QDebug & operator«(
88          QDebug & dbg, ListObjectsFilter filter);
89
91
92      struct QUENTIER_EXPORT ListObjectsFilters
93      {
94          std::optional<ListObjectsFilter> m_locallyModifiedFilter;
95          std::optional<ListObjectsFilter> m_withGuidFilter;
96          std::optional<ListObjectsFilter> m_localOnlyFilter;
97          std::optional<ListObjectsFilter> m_locallyFavoritedFilter;
98      };
99
100      friend QUENTIER_EXPORT QTextStream & operator«(
101          QTextStream & strm, const ListObjectsFilters & filters);
102
103      friend QUENTIER_EXPORT QDebug & operator«(
104          QDebug & dbg, const ListObjectsFilters & filters);
105
107
108      struct QUENTIER_EXPORT ListGuidsFilters
109      {
110          std::optional<ListObjectsFilter> m_locallyModifiedFilter;
111          std::optional<ListObjectsFilter> m_locallyFavoritedFilter;
112      };
113
114      friend QUENTIER_EXPORT QTextStream & operator«(
115          QTextStream & strm, const ListGuidsFilters & filters);
116
117      friend QUENTIER_EXPORT QDebug & operator«(
118          QDebug & dbg, const ListGuidsFilters & filters);
119
121
122      enum class OrderDirection
123      {
124          Ascending,
125          Descending
126      };
127
128      friend QUENTIER_EXPORT QTextStream & operator«(
129          QTextStream & strm, OrderDirection orderDirection);
130
131      friend QUENTIER_EXPORT QDebug & operator«(
132          QDebug & dbg, OrderDirection orderDirection);
133
135
136      enum class ListNotebooksOrder
137      {
138          NoOrder,
139          ByUpdateSequenceNumber,
140          ByNotebookName,
141          ByCreationTimestamp,
142          ByModificationTimestamp
143      };
144
145      friend QUENTIER_EXPORT QTextStream & operator«(
146          QTextStream & strm, ListNotebooksOrder order);
147
148      friend QUENTIER_EXPORT QDebug & operator«(
149          QDebug & dbg, ListNotebooksOrder order);
150
152
153      enum class ListLinkedNotebooksOrder
154      {
```

```
155         NoOrder,
156         ByUpdateSequenceNumber,
157         ByShareName,
158         ByUsername,
159     };
160
161     friend QUENTIER_EXPORT QTextStream & operator<<(
162         QTextStream & strm, ListLinkedNotebooksOrder order);
163
164     friend QUENTIER_EXPORT QDebug & operator<<(
165         QDebug & dbg, ListLinkedNotebooksOrder order);
166
168
169     enum class ListTagsOrder
170     {
171         NoOrder,
172         ByUpdateSequenceNumber,
173         ByName
174     };
175
176     friend QUENTIER_EXPORT QTextStream & operator<<(
177         QTextStream & strm, ListTagsOrder order);
178
179     friend QUENTIER_EXPORT QDebug & operator<<(
180         QDebug & dbg, ListTagsOrder order);
181
183
184     enum class ListNotesOrder
185     {
186         NoOrder,
187         ByUpdateSequenceNumber,
188         ByTitle,
189         ByCreationTimestamp,
190         ByModificationTimestamp,
191         ByDeletionTimestamp,
192         ByAuthor,
193         BySource,
194         BySourceApplication,
195         ByReminderTime,
196         ByPlaceName
197     };
198
199     friend QUENTIER_EXPORT QTextStream & operator<<(
200         QTextStream & strm, ListNotesOrder order);
201
202     friend QUENTIER_EXPORT QDebug & operator<<(
203         QDebug & dbg, ListNotesOrder order);
204
206
207     enum class ListSavedSearchesOrder
208     {
209         NoOrder,
210         ByUpdateSequenceNumber,
211         ByName,
212         ByFormat
213     };
214
215     friend QUENTIER_EXPORT QTextStream & operator<<(
216         QTextStream & strm, ListSavedSearchesOrder order);
217
218     friend QUENTIER_EXPORT QDebug & operator<<(
219         QDebug & dbg, ListSavedSearchesOrder order);
220
222
225     enum class Affiliation
226     {
227         Any,
228         User,
229         AnyLinkedNotebook,
230         ParticularLinkedNotebooks
231     };
232
233     friend QUENTIER_EXPORT QTextStream & operator<<(
234         QTextStream & strm, Affiliation affiliation);
235
236     friend QUENTIER_EXPORT QDebug & operator<<(
237         QDebug & dbg, Affiliation affiliation);
238
240
241     struct QUENTIER_EXPORT ListOptionsBase
242     {
243         ListOptionsBase() noexcept {}; // NOLINT
244
245         ListObjectsFilters m_filters = {};
246         quint64 m_limit = 0UL;
247         quint64 m_offset = 0UL;
248         OrderDirection m_direction = OrderDirection::Ascending;
```

```
249     };
250
251     struct QUENTIER_EXPORT ListNotebooksOptions :  public ListOptionsBase
252     {
253         ListNotebooksOptions() noexcept {}; // NOLINT
254
255         ListNotebooksOrder m_order = ListNotebooksOrder::NoOrder;
256         Affiliation m_affiliation = Affiliation::Any;
257         QList<qevercloud::Guid> m_linkedNotebookGuids;
258     };
259
260     friend QUENTIER_EXPORT QTextStream & operator<<(
261         QTextStream & strm, const ListNotebooksOptions & options);
262
263     friend QUENTIER_EXPORT QDebug & operator<<(
264         QDebug & dbg, const ListNotebooksOptions & options);
265
266     struct QUENTIER_EXPORT ListLinkedNotebooksOptions :  public ListOptionsBase
267     {
268         ListLinkedNotebooksOptions() noexcept {}; // NOLINT
269
270         ListLinkedNotebooksOrder m_order = ListLinkedNotebooksOrder::NoOrder;
271     };
272
273     friend QUENTIER_EXPORT QTextStream & operator<<(
274         QTextStream & strm, const ListLinkedNotebooksOptions & options);
275
276     friend QUENTIER_EXPORT QDebug & operator<<(
277         QDebug & dbg, const ListLinkedNotebooksOptions & options);
278
279     struct QUENTIER_EXPORT ListSavedSearchesOptions :  public ListOptionsBase
280     {
281         ListSavedSearchesOptions() noexcept {}; // NOLINT
282
283         ListSavedSearchesOrder m_order = ListSavedSearchesOrder::NoOrder;
284     };
285
286     friend QUENTIER_EXPORT QTextStream & operator<<(
287         QTextStream & strm, const ListSavedSearchesOptions & options);
288
289     friend QUENTIER_EXPORT QDebug & operator<<(
290         QDebug & dbg, const ListSavedSearchesOptions & options);
291
292     struct QUENTIER_EXPORT ListNotesOptions :  public ListOptionsBase
293     {
294         ListNotesOptions() noexcept {}; // NOLINT
295
296         ListNotesOrder m_order = ListNotesOrder::NoOrder;
297     };
298
299     friend QUENTIER_EXPORT QTextStream & operator<<(
300         QTextStream & strm, const ListNotesOptions & options);
301
302     friend QUENTIER_EXPORT QDebug & operator<<(
303         QDebug & dbg, const ListNotesOptions & options);
304
307     enum class TagNotesRelation
308     {
310         Any,
312         WithNotes,
314         WithoutNotes
315     };
316
317     struct QUENTIER_EXPORT ListTagsOptions :  public ListOptionsBase
318     {
319         ListTagsOptions() noexcept {}; // NOLINT
320
321         ListTagsOrder m_order = ListTagsOrder::NoOrder;
322         Affiliation m_affiliation = Affiliation::Any;
323         QList<qevercloud::Guid> m_linkedNotebookGuids;
324         TagNotesRelation m_tagNotesRelation = TagNotesRelation::Any;
325     };
326
327     friend QUENTIER_EXPORT QTextStream & operator<<(
328         QTextStream & strm, const ListTagsOptions & options);
329
330     friend QUENTIER_EXPORT QDebug & operator<<(
331         QDebug & dbg, const ListTagsOptions & options);
332
334
335     enum class NoteCountOption
336     {
337         IncludeNonDeletedNotes = 1 << 1,
338         IncludeDeletedNotes = 1 << 2
339     };
340     Q_DECLARE_FLAGS(NoteCountOptions, NoteCountOption)
341
```

```
342     friend QUENTIER_EXPORT QTextStream & operator<<(
343         QTextStream & strm, NoteCountOption option);
344
345     friend QUENTIER_EXPORT QDebug & operator<<(
346         QDebug & dbg, NoteCountOption option);
347
348     friend QUENTIER_EXPORT QTextStream & operator<<(
349         QTextStream & strm, NoteCountOptions options);
350
351     friend QUENTIER_EXPORT QDebug & operator<<(
352         QDebug & dbg, NoteCountOptions options);
353
355
356     enum class UpdateNoteOption
357     {
358         UpdateResourceMetadata = 1 << 1,
359         UpdateResourceBinaryData = 1 << 2,
360         UpdateTags = 1 << 3
361     };
362     Q_DECLARE_FLAGS(UpdateNoteOptions, UpdateNoteOption)
363
364     friend QUENTIER_EXPORT QTextStream & operator<<(
365         QTextStream & strm, UpdateNoteOption option);
366
367     friend QUENTIER_EXPORT QDebug & operator<<(
368         QDebug & dbg, UpdateNoteOption option);
369
370     friend QUENTIER_EXPORT QTextStream & operator<<(
371         QTextStream & strm, UpdateNoteOptions options);
372
373     friend QUENTIER_EXPORT QDebug & operator<<(
374         QDebug & dbg, UpdateNoteOptions options);
375
377
378     enum class FetchNoteOption
379     {
380         WithResourceMetadata = 1 << 1,
381         WithResourceBinaryData = 1 << 2
382     };
383     Q_DECLARE_FLAGS(FetchNoteOptions, FetchNoteOption)
384
385     friend QUENTIER_EXPORT QTextStream & operator<<(
386         QTextStream & strm, FetchNoteOption option);
387
388     friend QUENTIER_EXPORT QDebug & operator<<(
389         QDebug & dbg, FetchNoteOption option);
390
391     friend QUENTIER_EXPORT QTextStream & operator<<(
392         QTextStream & strm, FetchNoteOptions options);
393
394     friend QUENTIER_EXPORT QDebug & operator<<(
395         QDebug & dbg, FetchNoteOptions options);
396
398
399     enum class FetchResourceOption
400     {
401         WithBinaryData = 1 << 1
402     };
403     Q_DECLARE_FLAGS(FetchResourceOptions, FetchResourceOption)
404
405     friend QUENTIER_EXPORT QTextStream & operator<<(
406         QTextStream & strm, FetchResourceOption option);
407
408     friend QUENTIER_EXPORT QDebug & operator<<(
409         QDebug & dbg, FetchResourceOption option);
410
411     friend QUENTIER_EXPORT QTextStream & operator<<(
412         QTextStream & strm, FetchResourceOptions options);
413
414     friend QUENTIER_EXPORT QDebug & operator<<(
415         QDebug & dbg, FetchResourceOptions options);
416
418
419     enum class HighestUsnOption
420     {
421         WithinUserOwnContent,
422         WithinUserOwnContentAndLinkedNotebooks
423     };
424
425     friend QUENTIER_EXPORT QTextStream & operator<<(
426         QTextStream & strm, HighestUsnOption option);
427
428     friend QUENTIER_EXPORT QDebug & operator<<(
429         QDebug & dbg, HighestUsnOption option);
430
431 public:
432     // Versions/upgrade API
```

```
433     [[nodiscard]] virtual QFuture<bool> isVersionTooHigh() const = 0;
434     [[nodiscard]] virtual QFuture<bool> requiresUpgrade() const = 0;
435     [[nodiscard]] virtual QFuture<QList<IPatchPtr» requiredPatches() const = 0;
436     [[nodiscard]] virtual QFuture<qint32> version() const = 0;
437     [[nodiscard]] virtual QFuture<qint32> highestSupportedVersion() const = 0;
438
439     // Users API
440     [[nodiscard]] virtual QFuture<quint32> userCount() const = 0;
441     [[nodiscard]] virtual QFuture<void> putUser(qevercloud::User user) = 0;
442
443     [[nodiscard]] virtual QFuture<std::optional<qevercloud::User» findUserById(
444         qevercloud::UserID userId) const = 0;
445
446     [[nodiscard]] virtual QFuture<void> expungeUserById(
447         qevercloud::UserID userId) = 0;
448
449     // Notebooks API
450     [[nodiscard]] virtual QFuture<quint32> notebookCount() const = 0;
451
452     [[nodiscard]] virtual QFuture<void> putNotebook(
453         qevercloud::Notebook notebook) = 0;
454
455     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Notebook»
456         findNotebookByLocalId(QString notebookLocalId) const = 0;
457
458     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Notebook»
459         findNotebookByGuid(qevercloud::Guid guid) const = 0;
460
461     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Notebook»
462         findNotebookByName(
463             QString notebookName,
464             std::optional<qevercloud::Guid> linkedNotebookGuid =
465                 std::nullopt) const = 0;
466
467     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Notebook»
468         findDefaultNotebook() const = 0;
469
470     [[nodiscard]] virtual QFuture<void> expungeNotebookByLocalId(
471         QString notebookLocalId) = 0;
472
473     [[nodiscard]] virtual QFuture<void> expungeNotebookByGuid(
474         qevercloud::Guid notebookGuid) = 0;
475
476     [[nodiscard]] virtual QFuture<void> expungeNotebookByName(
477         QString name,
478         std::optional<qevercloud::Guid> linkedNotebookGuid = std::nullopt) = 0;
479
480     [[nodiscard]] virtual QFuture<QList<qevercloud::Notebook» listNotebooks(
481         ListNotebooksOptions options = {}) const = 0;
482
483     [[nodiscard]] virtual QFuture<QList<qevercloud::SharedNotebook»
484         listSharedNotebooks(qevercloud::Guid notebookGuid = {}) const = 0;
485
486     [[nodiscard]] virtual QFuture<QSet<qevercloud::Guid» listNotebookGuids(
487         ListGuidsFilters filters,
488         std::optional<qevercloud::Guid> linkedNotebookGuid = {}) const = 0;
489
490     // Linked notebooks API
491     [[nodiscard]] virtual QFuture<quint32> linkedNotebookCount() const = 0;
492
493     [[nodiscard]] virtual QFuture<void> putLinkedNotebook(
494         qevercloud::LinkedNotebook linkedNotebook) = 0;
495
496     [[nodiscard]] virtual QFuture<std::optional<qevercloud::LinkedNotebook»
497         findLinkedNotebookByGuid(qevercloud::Guid guid) const = 0;
498
499     [[nodiscard]] virtual QFuture<void> expungeLinkedNotebookByGuid(
500         qevercloud::Guid guid) = 0;
501
502     [[nodiscard]] virtual QFuture<QList<qevercloud::LinkedNotebook»
503         listLinkedNotebooks(ListLinkedNotebooksOptions options = {}) const = 0;
504
505     // Notes API
506     [[nodiscard]] virtual QFuture<quint32> noteCount(
507         NoteCountOptions options = NoteCountOptions(
508             NoteCountOption::IncludeNonDeletedNotes)) const = 0;
509
510     [[nodiscard]] virtual QFuture<quint32> noteCountPerNotebookLocalId(
511         QString notebookLocalId,
512         NoteCountOptions options = NoteCountOptions(
513             NoteCountOption::IncludeNonDeletedNotes)) const = 0;
514
515     [[nodiscard]] virtual QFuture<quint32> noteCountPerTagLocalId(
516         QString tagLocalId,
517         NoteCountOptions options = NoteCountOptions(
518             NoteCountOption::IncludeNonDeletedNotes)) const = 0;
519
```

```
520     [[nodiscard]] virtual QFuture<QHash<QString, quint32» noteCountsPerTags(
521         ListTagsOptions listTagsOptions = {},
522         NoteCountOptions options = NoteCountOptions(
523             NoteCountOption::IncludeNonDeletedNotes)) const = 0;
524
525     [[nodiscard]] virtual QFuture<quint32> noteCountPerNotebookAndTagLocalIds(
526         QStringList notebookLocalIds, QStringList tagLocalIds,
527         NoteCountOptions options = NoteCountOptions(
528             NoteCountOption::IncludeNonDeletedNotes)) const = 0;
529
530     [[nodiscard]] virtual QFuture<void> putNote(qevercloud::Note note) = 0;
531
532     [[nodiscard]] virtual QFuture<void> updateNote(
533         qevercloud::Note note, UpdateNoteOptions options) = 0;
534
535     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Note»
536         findNoteByLocalId(
537             QString noteLocalId, FetchNoteOptions options) const = 0;
538
539     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Note»
540         findNoteByGuid(
541             qevercloud::Guid noteGuid, FetchNoteOptions options) const = 0;
542
543     [[nodiscard]] virtual QFuture<QList<qevercloud::Note» listNotes(
544         FetchNoteOptions fetchOptions,
545         ListNotesOptions listOptions = {}) const = 0;
546
547     [[nodiscard]] virtual QFuture<QList<qevercloud::Note»
548         listNotesPerNotebookLocalId(
549             QString notebookLocalId, FetchNoteOptions fetchOptions,
550             ListNotesOptions listOptions = {}) const = 0;
551
552     [[nodiscard]] virtual QFuture<QList<qevercloud::Note»
553         listNotesPerTagLocalId(
554             QString tagLocalId, FetchNoteOptions fetchOptions,
555             ListNotesOptions listOptions = {}) const = 0;
556
557     [[nodiscard]] virtual QFuture<QList<qevercloud::Note»
558         listNotesPerNotebookAndTagLocalIds(
559             QStringList notebookLocalIds, QStringList tagLocalIds,
560             FetchNoteOptions fetchOptions,
561             ListNotesOptions listOptions = {}) const = 0;
562
563     [[nodiscard]] virtual QFuture<QList<qevercloud::Note» listNotesByLocalIds(
564         QStringList noteLocalIds, FetchNoteOptions fetchOptions,
565         ListNotesOptions listOptions = {}) const = 0;
566
567     [[nodiscard]] virtual QFuture<QSet<qevercloud::Guid» listNoteGuids(
568         ListGuidsFilters filters,
569         std::optional<qevercloud::Guid> linkedNotebookGuid = {}) const = 0;
570
571     [[nodiscard]] virtual QFuture<QList<qevercloud::Note» queryNotes(
572         NoteSearchQuery query, FetchNoteOptions fetchOptions) const = 0;
573
574     [[nodiscard]] virtual QFuture<QStringList> queryNoteLocalIds(
575         NoteSearchQuery query) const = 0;
576
577     [[nodiscard]] virtual QFuture<void> expungeNoteByLocalId(
578         QString noteLocalId) = 0;
579
580     [[nodiscard]] virtual QFuture<void> expungeNoteByGuid(
581         qevercloud::Guid noteGuid) = 0;
582
583     // Tags API
584     [[nodiscard]] virtual QFuture<quint32> tagCount() const = 0;
585     [[nodiscard]] virtual QFuture<void> putTag(qevercloud::Tag tag) = 0;
586
587     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Tag»
588         findTagByLocalId(QString tagLocalId) const = 0;
589
590     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Tag» findTagByGuid(
591         qevercloud::Guid tagGuid) const = 0;
592
593     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Tag» findTagByName(
594         QString tagName,
595         std::optional<qevercloud::Guid> linkedNotebookGuid =
596             std::nullopt) const = 0;
597
598     [[nodiscard]] virtual QFuture<QList<qevercloud::Tag» listTags(
599         ListTagsOptions options = {}) const = 0;
600
601     [[nodiscard]] virtual QFuture<QList<qevercloud::Tag»
602         listTagsPerNoteLocalId(
603             QString noteLocalId, ListTagsOptions options = {}) const = 0;
604
605     [[nodiscard]] virtual QFuture<QSet<qevercloud::Guid» listTagGuids(
606         ListGuidsFilters filters,
```

```
607         std::optional<qevercloud::Guid> linkedNotebookGuid = {}) const = 0;
608
609     [[nodiscard]] virtual QFuture<void> expungeTagByLocalId(
610         QString tagLocalId) = 0;
611
612     [[nodiscard]] virtual QFuture<void> expungeTagByGuid(
613         qevercloud::Guid tagGuid) = 0;
614
615     [[nodiscard]] virtual QFuture<void> expungeTagByName(
616         QString name,
617         std::optional<qevercloud::Guid> linkedNotebookGuid = std::nullopt) = 0;
618
619     // Resources API
620     [[nodiscard]] virtual QFuture<quint32> resourceCount(
621         NoteCountOptions options = NoteCountOptions(
622             NoteCountOption::IncludeNonDeletedNotes)) const = 0;
623
624     [[nodiscard]] virtual QFuture<quint32> resourceCountPerNoteLocalId(
625         QString noteLocalId) const = 0;
626
627     [[nodiscard]] virtual QFuture<void> putResource(
628         qevercloud::Resource resource) = 0;
629
630     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Resource»
631         findResourceByLocalId(
632             QString resourceLocalId,
633             FetchResourceOptions options = {}) const = 0;
634
635     [[nodiscard]] virtual QFuture<std::optional<qevercloud::Resource»
636         findResourceByGuid(
637             qevercloud::Guid resourceGuid,
638             FetchResourceOptions options = {}) const = 0;
639
640     [[nodiscard]] virtual QFuture<void> expungeResourceByLocalId(
641         QString resourceLocalId) = 0;
642
643     [[nodiscard]] virtual QFuture<void> expungeResourceByGuid(
644         qevercloud::Guid resourceGuid) = 0;
645
646     // Saved searches API
647     [[nodiscard]] virtual QFuture<quint32> savedSearchCount() const = 0;
648
649     [[nodiscard]] virtual QFuture<void> putSavedSearch(
650         qevercloud::SavedSearch search) = 0;
651
652     [[nodiscard]] virtual QFuture<std::optional<qevercloud::SavedSearch»
653         findSavedSearchByLocalId(QString savedSearchLocalId) const = 0;
654
655     [[nodiscard]] virtual QFuture<std::optional<qevercloud::SavedSearch»
656         findSavedSearchByGuid(qevercloud::Guid guid) const = 0;
657
658     [[nodiscard]] virtual QFuture<std::optional<qevercloud::SavedSearch»
659         findSavedSearchByName(QString name) const = 0;
660
661     [[nodiscard]] virtual QFuture<QList<qevercloud::SavedSearch»
662         listSavedSearches(ListSavedSearchesOptions options = {}) const = 0;
663
664     [[nodiscard]] virtual QFuture<QSet<qevercloud::Guid» listSavedSearchGuids(
665         ListGuidsFilters filters) const = 0;
666
667     [[nodiscard]] virtual QFuture<void> expungeSavedSearchByLocalId(
668         QString savedSearchLocalId) = 0;
669
670     [[nodiscard]] virtual QFuture<void> expungeSavedSearchByGuid(
671         qevercloud::Guid guid) = 0;
672
673     // Synchronization API
674     [[nodiscard]] virtual QFuture<qint32> highestUpdateSequenceNumber(
675         HighestUsnOption option) const = 0;
676
677     [[nodiscard]] virtual QFuture<qint32> highestUpdateSequenceNumber(
678         qevercloud::Guid linkedNotebookGuid) const = 0;
679
686     [[nodiscard]] virtual ILocalStorageNotifier * notifier() const = 0;
687 };
688
689 [[nodiscard]] QUENTIER_EXPORT bool operator==(
690     const ILocalStorage::ListObjectsFilters & lhs,
691     const ILocalStorage::ListObjectsFilters & rhs) noexcept;
692
693 [[nodiscard]] QUENTIER_EXPORT bool operator==(
694     const ILocalStorage::ListOptionsBase & lhs,
695     const ILocalStorage::ListOptionsBase & rhs) noexcept;
696
697 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
698     const ILocalStorage::ListOptionsBase & lhs,
699     const ILocalStorage::ListOptionsBase & rhs) noexcept;
```

```
700
701 [[nodiscard]] QUENTIER_EXPORT bool operator==(
702     const ILocalStorage::ListNotebooksOptions & lhs,
703     const ILocalStorage::ListNotebooksOptions & rhs) noexcept;
704
705 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
706     const ILocalStorage::ListNotebooksOptions & lhs,
707     const ILocalStorage::ListNotebooksOptions & rhs) noexcept;
708
709 [[nodiscard]] QUENTIER_EXPORT bool operator==(
710     const ILocalStorage::ListLinkedNotebooksOptions & lhs,
711     const ILocalStorage::ListLinkedNotebooksOptions & rhs) noexcept;
712
713 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
714     const ILocalStorage::ListLinkedNotebooksOptions & lhs,
715     const ILocalStorage::ListLinkedNotebooksOptions & rhs) noexcept;
716
717 [[nodiscard]] QUENTIER_EXPORT bool operator==(
718     const ILocalStorage::ListSavedSearchesOptions & lhs,
719     const ILocalStorage::ListSavedSearchesOptions & rhs) noexcept;
720
721 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
722     const ILocalStorage::ListSavedSearchesOptions & lhs,
723     const ILocalStorage::ListSavedSearchesOptions & rhs) noexcept;
724
725 [[nodiscard]] QUENTIER_EXPORT bool operator==(
726     const ILocalStorage::ListNotesOptions & lhs,
727     const ILocalStorage::ListNotesOptions & rhs) noexcept;
728
729 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
730     const ILocalStorage::ListNotesOptions & lhs,
731     const ILocalStorage::ListNotesOptions & rhs) noexcept;
732
733 [[nodiscard]] QUENTIER_EXPORT bool operator==(
734     const ILocalStorage::ListTagsOptions & lhs,
735     const ILocalStorage::ListTagsOptions & rhs) noexcept;
736
737 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
738     const ILocalStorage::ListTagsOptions & lhs,
739     const ILocalStorage::ListTagsOptions & rhs) noexcept;
740
741 [[nodiscard]] QUENTIER_EXPORT bool operator==(
742     const ILocalStorage::ListGuidsFilters & lhs,
743     const ILocalStorage::ListGuidsFilters & rhs) noexcept;
744
745 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
746     const ILocalStorage::ListGuidsFilters & lhs,
747     const ILocalStorage::ListGuidsFilters & rhs) noexcept;
748
749 } // namespace quentier::local_storage
```

6.28 ILocalStorageNotifier.h

```
1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/local_storage/ILocalStorage.h>
22
23 #include <QObject>
24
25 namespace quentier::local_storage {
26
27 class QUENTIER_EXPORT ILocalStorageNotifier :  public QObject
28 {
29     Q_OBJECT
30 protected:
```

```
31     explicit ILocalStorageNotifier(QObject * parent = nullptr);
32
33 public:
34     ~ILocalStorageNotifier() override;
35
36 Q_SIGNALS:
37     // Notifications about user related events
38     void userPut(qevercloud::User user);
39     void userExpunged(qevercloud::UserID userId);
40
41     // Notifications about notebook related events
42     void notebookPut(qevercloud::Notebook notebook);
43     void notebookExpunged(QString notebookLocalId);
44
45     // Notifications about linked notebooks
46     void linkedNotebookPut(qevercloud::LinkedNotebook linkedNotebook);
47     void linkedNotebookExpunged(qevercloud::Guid linkedNotebookGuid);
48
49     // Notifications about note related events
50     void notePut(qevercloud::Note note);
51
52     void noteUpdated(
53         qevercloud::Note note, ILocalStorage::UpdateNoteOptions options);
54
55     void noteExpunged(QString noteLocalId);
56
57     // Notifications about tag related events
58     void tagPut(qevercloud::Tag tag);
59
60     void tagExpunged(QString tagLocalId, QStringList expungedChildTagLocalIds);
61
62     // Notifications about resource related events
63     void resourcePut(qevercloud::Resource resource);
64     void resourceMetadataPut(qevercloud::Resource resource);
65     void resourceExpunged(QString resourceLocalId);
66
67     // Notifications about saved search related events
68     void savedSearchPut(qevercloud::SavedSearch savedSearch);
69     void savedSearchExpunged(QString savedSearchLocalId);
70 };
71
72 } // namespace quentier::local_storage
```

6.29 IPatch.h

```
1 /*
2 * Copyright 2021-2022 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QFuture>
24
25 namespace quentier::local_storage {
26
33 class QUENTIER_EXPORT IPatch
34 {
35 public:
36     virtual ~IPatch() noexcept;
37
42     [[nodiscard]] virtual int fromVersion() const noexcept = 0;
43
48     [[nodiscard]] virtual int toVersion() const noexcept = 0;
49
53     [[nodiscard]] virtual QString patchShortDescription() const = 0;
54
58     [[nodiscard]] virtual QString patchLongDescription() const = 0;
```

```
59
69      [[nodiscard]] virtual QFuture<void> backupLocalStorage() = 0;
70
80      [[nodiscard]] virtual QFuture<void> restoreLocalStorageFromBackup() = 0;
81
92      [[nodiscard]] virtual QFuture<void> removeLocalStorageBackup() = 0;
93
101      [[nodiscard]] virtual QFuture<void> apply() = 0;
102 };
103
104 } // namespace quentier::local_storage
```

6.30 LocalStorageOpenException.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/exception/IQuentierException.h>
22
23 namespace quentier::local_storage {
24
29 class QUENTIER_EXPORT LocalStorageOpenException :  public IQuentierException
30 {
31 public:
32     explicit LocalStorageOpenException(const ErrorString & message);
33
34     [[nodiscard]] LocalStorageOpenException * clone() const override;
35     void raise() const override;
36
37 protected:
38     [[nodiscard]] QString exceptionDisplayName() const override;
39 };
40
41 } // namespace quentier::local_storage
```

6.31 LocalStorageOperationException.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/exception/IQuentierException.h>
22
23 namespace quentier::local_storage {
24
29 class QUENTIER_EXPORT LocalStorageOperationException :  public IQuentierException
```

```
30 {
31 public:
32     explicit LocalStorageOperationException(ErrorString message);
33
34     [[nodiscard]] LocalStorageOperationException * clone() const override;
35     void raise() const override;
36
37 protected:
38     [[nodiscard]] QString exceptionDisplayName() const override;
39 };
40
41 } // namespace quentier::local_storage
```

6.32 NoteSearchQuery.h

```
1 /*
2 * Copyright 2016-2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/ErrorString.h>
22
23 #include <QList>
24 #include <QSharedDataPointer>
25
26 namespace quentier::local_storage {
27
28 class QUENTIER_EXPORT NoteSearchQuery :  public Printable
29 {
30 public:
31     explicit NoteSearchQuery();
32
33     NoteSearchQuery(const NoteSearchQuery & other);
34     NoteSearchQuery(NoteSearchQuery && other) noexcept;
35
36     NoteSearchQuery & operator=(const NoteSearchQuery & other);
37     NoteSearchQuery & operator=(NoteSearchQuery && other) noexcept;
38
39     ~NoteSearchQuery() override;
40
41     [[nodiscard]] bool isEmpty() const;
42
43     void clear();
44
48     [[nodiscard]] QString queryString() const;
49
50     [[nodiscard]] bool setQueryString(
51         const QString & queryString, ErrorString & error);
52
58     [[nodiscard]] QString notebookModifier() const;
59
60     [[nodiscard]] bool hasAnyModifier() const;
61
62     [[nodiscard]] const QStringList & tagNames() const;
63     [[nodiscard]] const QStringList & negatedTagNames() const;
64     [[nodiscard]] bool hasAnyTag() const;
65     [[nodiscard]] bool hasNegatedAnyTag() const;
66
67     [[nodiscard]] const QStringList & titleNames() const;
68     [[nodiscard]] const QStringList & negatedTitleNames() const;
69     [[nodiscard]] bool hasAnyTitleName() const;
70     [[nodiscard]] bool hasNegatedAnyTitleName() const;
71
72     [[nodiscard]] const QList<qint64> & creationTimestamps() const;
73     [[nodiscard]] const QList<qint64> & negatedCreationTimestamps() const;
74     [[nodiscard]] bool hasAnyCreationTimestamp() const;
75     [[nodiscard]] bool hasNegatedAnyCreationTimestamp() const;
76
```

```
77      [[nodiscard]] const QList<qint64> & modificationTimestamps() const;
78      [[nodiscard]] const QList<qint64> & negatedModificationTimestamps() const;
79      [[nodiscard]] bool hasAnyModificationTimestamp() const;
80      [[nodiscard]] bool hasNegatedAnyModificationTimestamp() const;
81
82      [[nodiscard]] const QStringList & resourceMimeTypes() const;
83      [[nodiscard]] const QStringList & negatedResourceMimeTypes() const;
84      [[nodiscard]] bool hasAnyResourceMimeType() const;
85      [[nodiscard]] bool hasNegatedAnyResourceMimeType() const;
86
87      [[nodiscard]] const QList<qint64> & subjectDateTimestamps() const;
88      [[nodiscard]] const QList<qint64> & negatedSubjectDateTimestamps() const;
89      [[nodiscard]] bool hasAnySubjectDateTimestamp() const;
90      [[nodiscard]] bool hasNegatedAnySubjectDateTimestamp() const;
91
92      [[nodiscard]] const QList<double> & latitudes() const;
93      [[nodiscard]] const QList<double> & negatedLatitudes() const;
94      [[nodiscard]] bool hasAnyLatitude() const;
95      [[nodiscard]] bool hasNegatedAnyLatitude() const;
96
97      [[nodiscard]] const QList<double> & longitudes() const;
98      [[nodiscard]] const QList<double> & negatedLongitudes() const;
99      [[nodiscard]] bool hasAnyLongitude() const;
100     [[nodiscard]] bool hasNegatedAnyLongitude() const;
101
102     [[nodiscard]] const QList<double> & altitudes() const;
103     [[nodiscard]] const QList<double> & negatedAltitudes() const;
104     [[nodiscard]] bool hasAnyAltitude() const;
105     [[nodiscard]] bool hasNegatedAnyAltitude() const;
106
107     [[nodiscard]] const QStringList & authors() const;
108     [[nodiscard]] const QStringList & negatedAuthors() const;
109     [[nodiscard]] bool hasAnyAuthor() const;
110     [[nodiscard]] bool hasNegatedAnyAuthor() const;
111
112     [[nodiscard]] const QStringList & sources() const;
113     [[nodiscard]] const QStringList & negatedSources() const;
114     [[nodiscard]] bool hasAnySource() const;
115     [[nodiscard]] bool hasNegatedAnySource() const;
116
117     [[nodiscard]] const QStringList & sourceApplications() const;
118     [[nodiscard]] const QStringList & negatedSourceApplications() const;
119     [[nodiscard]] bool hasAnySourceApplication() const;
120     [[nodiscard]] bool hasNegatedAnySourceApplication() const;
121
122     [[nodiscard]] const QStringList & contentClasses() const;
123     [[nodiscard]] const QStringList & negatedContentClasses() const;
124     [[nodiscard]] bool hasAnyContentClass() const;
125     [[nodiscard]] bool hasNegatedAnyContentClass() const;
126
127     [[nodiscard]] const QStringList & placeNames() const;
128     [[nodiscard]] const QStringList & negatedPlaceNames() const;
129     [[nodiscard]] bool hasAnyPlaceName() const;
130     [[nodiscard]] bool hasNegatedAnyPlaceName() const;
131
132     [[nodiscard]] const QStringList & applicationData() const;
133     [[nodiscard]] const QStringList & negatedApplicationData() const;
134     [[nodiscard]] bool hasAnyApplicationData() const;
135     [[nodiscard]] bool hasNegatedAnyApplicationData() const;
136
137     [[nodiscard]] const QList<qint64> & reminderOrders() const;
138     [[nodiscard]] const QList<qint64> & negatedReminderOrders() const;
139     [[nodiscard]] bool hasAnyReminderOrder() const;
140     [[nodiscard]] bool hasNegatedAnyReminderOrder() const;
141
142     [[nodiscard]] const QList<qint64> & reminderTimes() const;
143     [[nodiscard]] const QList<qint64> & negatedReminderTimes() const;
144     [[nodiscard]] bool hasAnyReminderTime() const;
145     [[nodiscard]] bool hasNegatedAnyReminderTime() const;
146
147     [[nodiscard]] const QList<qint64> & reminderDoneTimes() const;
148     [[nodiscard]] const QList<qint64> & negatedReminderDoneTimes() const;
149     [[nodiscard]] bool hasAnyReminderDoneTime() const;
150     [[nodiscard]] bool hasNegatedAnyReminderDoneTime() const;
151
152     [[nodiscard]] bool hasUnfinishedToDo() const;
153     [[nodiscard]] bool hasNegatedUnfinishedToDo() const;
154
155     [[nodiscard]] bool hasFinishedToDo() const;
156     [[nodiscard]] bool hasNegatedFinishedToDo() const;
157
158     [[nodiscard]] bool hasAnyToDo() const;
159     [[nodiscard]] bool hasNegatedAnyToDo() const;
160
161     [[nodiscard]] bool hasEncryption() const;
162     [[nodiscard]] bool hasNegatedEncryption() const;
163
```

```
164     [[nodiscard]] const QStringList & contentSearchTerms() const;
165     [[nodiscard]] const QStringList & negatedContentSearchTerms() const;
166     [[nodiscard]] bool hasAnyContentSearchTerms() const;
167
168     [[nodiscard]] bool isMatcheable() const;
169
170     QTextStream & print(QTextStream & strm) const override;
171
172 private:
173     class Data;
174     QSharedDataPointer<Data> d;
175 };
176
177 [[nodiscard]] QUENTIER_EXPORT bool operator==(
178     const NoteSearchQuery & lhs, const NoteSearchQuery & rhs) noexcept;
179
180 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
181     const NoteSearchQuery & lhs, const NoteSearchQuery & rhs) noexcept;
182
183 } // namespace quentier::local_storage
```

6.33 MockILocalStorage.h

```
1 /*
2 * Copyright 2021-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/local_storage/ILocalStorage.h>
22
23 #include <gmock/gmock.h>
24
25 namespace quentier::local_storage::tests::mocks {
26
27 class MockILocalStorage :  public ILocalStorage
28 {
29 public:
30     MOCK_METHOD(QFuture<bool>, isVersionTooHigh, (), (const, override));
31     MOCK_METHOD(QFuture<bool>, requiresUpgrade, (), (const, override));
32
33     MOCK_METHOD(
34         QFuture<QList<IPatchPtr», requiredPatches, (), (const, override));
35
36     MOCK_METHOD(QFuture<qint32>, version, (), (const, override));
37
38     MOCK_METHOD(
39         QFuture<qint32>, highestSupportedVersion, (), (const, override));
40
41     MOCK_METHOD(QFuture<quint32>, userCount, (), (const, override));
42     MOCK_METHOD(QFuture<void>, putUser, (qevercloud::User user), (override));
43
44     MOCK_METHOD(
45         QFuture<std::optional<qevercloud::User», findUserById,
46         (qevercloud::UserID userId), (const, override));
47
48     MOCK_METHOD(
49         QFuture<void>, expungeUserById, (qevercloud::UserID userId),
50         (override));
51
52     MOCK_METHOD(QFuture<quint32>, notebookCount, (), (const, override));
53
54     MOCK_METHOD(
55         QFuture<void>, putNotebook, (qevercloud::Notebook notebook),
56         (override));
57
58     MOCK_METHOD(
59         QFuture<std::optional<qevercloud::Notebook», findNotebookByLocalId,
60         (QString localId), (const, override));
```

```
61
62      MOCK_METHOD(
63          QFuture<std::optional<qevercloud::Notebook>>, findNotebookByGuid,
64          (qevercloud::Guid guid), (const, override));
65
66      MOCK_METHOD(
67          QFuture<std::optional<qevercloud::Notebook>>, findNotebookByName,
68          (QString name, std::optional<qevercloud::Guid> linkedNotebookGuid),
69          (const, override));
70
71      MOCK_METHOD(
72          QFuture<std::optional<qevercloud::Notebook>>, findDefaultNotebook, (),
73          (const, override));
74
75      MOCK_METHOD(
76          QFuture<void>, expungeNotebookByLocalId, (QString localId), (override));
77
78      MOCK_METHOD(
79          QFuture<void>, expungeNotebookByGuid, (qevercloud::Guid guid),
80          (override));
81
82      MOCK_METHOD(
83          QFuture<void>, expungeNotebookByName,
84          (QString name, std::optional<qevercloud::Guid> linkedNotebookGuid),
85          (override));
86
87      MOCK_METHOD(
88          QFuture<QList<qevercloud::Notebook>>, listNotebooks,
89          (ListNotebooksOptions options), (const, override));
90
91      MOCK_METHOD(
92          QFuture<QList<qevercloud::SharedNotebook>>, listSharedNotebooks,
93          (qevercloud::Guid notebookGuid), (const, override));
94
95      MOCK_METHOD(
96          QFuture<QSet<qevercloud::Guid>>, listNotebookGuids,
97          (ListGuidsFilters filters,
98           std::optional<qevercloud::Guid> linkedNotebookGuid),
99          (const, override));
100
101     MOCK_METHOD(QFuture<quint32>, linkedNotebookCount, (), (const, override));
102
103     MOCK_METHOD(
104         QFuture<void>, putLinkedNotebook,
105         (qevercloud::LinkedNotebook linkedNotebook), (override));
106
107     MOCK_METHOD(
108         QFuture<std::optional<qevercloud::LinkedNotebook>>,
109         findLinkedNotebookByGuid, (qevercloud::Guid guid), (const, override));
110
111     MOCK_METHOD(
112         QFuture<void>, expungeLinkedNotebookByGuid, (qevercloud::Guid guid),
113         (override));
114
115     MOCK_METHOD(
116         QFuture<QList<qevercloud::LinkedNotebook>>, listLinkedNotebooks,
117         (ListLinkedNotebooksOptions options), (const, override));
118
119     MOCK_METHOD(
120         QFuture<quint32>, noteCount, (NoteCountOptions options),
121         (const, override));
122
123     MOCK_METHOD(
124         QFuture<quint32>, noteCountPerNotebookLocalId,
125         (QString notebookLocalId, NoteCountOptions options), (const, override));
126
127     MOCK_METHOD(
128         QFuture<quint32>, noteCountPerTagLocalId,
129         (QString tagLocalId, NoteCountOptions options), (const, override));
130
131     MOCK_METHOD(
132         (QFuture<QHash<QString, quint32>>), noteCountsPerTags,
133         (ListTagsOptions listTagsOptions, NoteCountOptions options),
134         (const, override));
135
136     MOCK_METHOD(
137         QFuture<quint32>, noteCountPerNotebookAndTagLocalIds,
138         (QStringList notebookLocalIds, QStringList tagLocalIds,
139          NoteCountOptions options),
140         (const, override));
141
142     MOCK_METHOD(QFuture<void>, putNote, (qevercloud::Note note), (override));
143
144     MOCK_METHOD(
145         QFuture<void>, updateNote,
146         (qevercloud::Note note, UpdateNoteOptions options), (override));
147
```

```
148     MOCK_METHOD(
149         QFuture<std::optional<qevercloud::Note», findNoteByLocalId,
150         (QString localId, FetchNoteOptions options), (const, override));
151
152     MOCK_METHOD(
153         QFuture<std::optional<qevercloud::Note», findNoteByGuid,
154         (qevercloud::Guid guid, FetchNoteOptions options), (const, override));
155
156     MOCK_METHOD(
157         QFuture<void>, expungeNoteByLocalId, (QString localId), (override));
158
159     MOCK_METHOD(
160         QFuture<void>, expungeNoteByGuid, (qevercloud::Guid guid), (override));
161
162     MOCK_METHOD(
163         QFuture<QList<qevercloud::Note», listNotes,
164         (FetchNoteOptions fetchOptions, ListNotesOptions options),
165         (const, override));
166
167     MOCK_METHOD(
168         QFuture<QList<qevercloud::Note», listNotesPerNotebookLocalId,
169         (QString notebookLocalId, FetchNoteOptions fetchOptions,
170          ListNotesOptions options),
171         (const, override));
172
173     MOCK_METHOD(
174         QFuture<QList<qevercloud::Note», listNotesPerTagLocalId,
175         (QString tagLocalId, FetchNoteOptions fetchOptions,
176          ListNotesOptions options),
177         (const, override));
178
179     MOCK_METHOD(
180         QFuture<QList<qevercloud::Note», listNotesPerNotebookAndTagLocalIds,
181         (QStringList notebookLocalIds, QStringList tagLocalIds,
182          FetchNoteOptions fetchOptions, ListNotesOptions options),
183         (const, override));
184
185     MOCK_METHOD(
186         QFuture<QList<qevercloud::Note», listNotesByLocalIds,
187         (QStringList noteLocalIds, FetchNoteOptions fetchOptions,
188          ListNotesOptions options),
189         (const, override));
190
191     MOCK_METHOD(
192         QFuture<QSet<qevercloud::Guid», listNoteGuids,
193         (ListGuidsFilters filters,
194          std::optional<qevercloud::Guid> linkedNotebookGuid),
195         (const, override));
196
197     MOCK_METHOD(
198         QFuture<QList<qevercloud::Note», queryNotes,
199         (NoteSearchQuery query, FetchNoteOptions fetchOptions),
200         (const, override));
201
202     MOCK_METHOD(
203         QFuture<QStringList>, queryNoteLocalIds, (NoteSearchQuery query),
204         (const, override));
205
206     MOCK_METHOD(QFuture<quint32>, tagCount, (), (const, override));
207     MOCK_METHOD(QFuture<void>, putTag, (qevercloud::Tag tag), (override));
208
209     MOCK_METHOD(
210         QFuture<std::optional<qevercloud::Tag», findTagByLocalId,
211         (QString tagLocalId), (const, override));
212
213     MOCK_METHOD(
214         QFuture<std::optional<qevercloud::Tag», findTagByGuid,
215         (qevercloud::Guid tagGuid), (const, override));
216
217     MOCK_METHOD(
218         QFuture<std::optional<qevercloud::Tag», findTagByName,
219         (QString tagName, std::optional<QString> linkedNotebookGuid),
220         (const, override));
221
222     MOCK_METHOD(
223         QFuture<QList<qevercloud::Tag», listTags, (ListTagsOptions options),
224         (const, override));
225
226     MOCK_METHOD(
227         QFuture<QList<qevercloud::Tag», listTagsPerNoteLocalId,
228         (QString noteLocalId, ListTagsOptions options), (const, override));
229
230     MOCK_METHOD(
231         QFuture<QSet<qevercloud::Guid», listTagGuids,
232         (ListGuidsFilters filters,
233          std::optional<qevercloud::Guid> linkedNotebookGuid),
234         (const, override));
```

```
235
236     MOCK_METHOD(
237         QFuture<void>, expungeTagByLocalId, (QString tagLocalId), (override));
238
239     MOCK_METHOD(
240         QFuture<void>, expungeTagByGuid, (qevercloud::Guid tagGuid),
241         (override));
242
243     MOCK_METHOD(
244         QFuture<void>, expungeTagByName,
245         (QString name, std::optional<qevercloud::Guid> linkedNotebookGuid),
246         (override));
247
248     MOCK_METHOD(
249         QFuture<quint32>, resourceCount, (NoteCountOptions options),
250         (const, override));
251
252     MOCK_METHOD(
253         QFuture<quint32>, resourceCountPerNoteLocalId, (QString noteLocalId),
254         (const, override));
255
256     MOCK_METHOD(
257         QFuture<void>, putResource, (qevercloud::Resource resource),
258         (override));
259
260     MOCK_METHOD(
261         QFuture<std::optional<qevercloud::Resource», findResourceByLocalId,
262         (QString resourceLocalId, FetchResourceOptions options),
263         (const, override));
264
265     MOCK_METHOD(
266         QFuture<std::optional<qevercloud::Resource», findResourceByGuid,
267         (qevercloud::Guid resourceGuid, FetchResourceOptions options),
268         (const, override));
269
270     MOCK_METHOD(
271         QFuture<void>, expungeResourceByLocalId, (QString resourceLocalId),
272         (override));
273
274     MOCK_METHOD(
275         QFuture<void>, expungeResourceByGuid, (qevercloud::Guid resourceGuid),
276         (override));
277
278     MOCK_METHOD(QFuture<quint32>, savedSearchCount, (), (const, override));
279
280     MOCK_METHOD(
281         QFuture<void>, putSavedSearch, (qevercloud::SavedSearch search),
282         (override));
283
284     MOCK_METHOD(
285         QFuture<std::optional<qevercloud::SavedSearch»,
286         findSavedSearchByLocalId, (QString localId), (const, override));
287
288     MOCK_METHOD(
289         QFuture<std::optional<qevercloud::SavedSearch», findSavedSearchByGuid,
290         (qevercloud::Guid guid), (const, override));
291
292     MOCK_METHOD(
293         QFuture<std::optional<qevercloud::SavedSearch», findSavedSearchByName,
294         (QString name), (const, override));
295
296     MOCK_METHOD(
297         QFuture<QList<qevercloud::SavedSearch», listSavedSearches,
298         (ListSavedSearchesOptions options), (const, override));
299
300     MOCK_METHOD(
301         QFuture<QSet<qevercloud::Guid», listSavedSearchGuids,
302         (ListGuidsFilters filters), (const, override));
303
304     MOCK_METHOD(
305         QFuture<void>, expungeSavedSearchByLocalId, (QString localId),
306         (override));
307
308     MOCK_METHOD(
309         QFuture<void>, expungeSavedSearchByGuid, (qevercloud::Guid guid),
310         (override));
311
312     MOCK_METHOD(
313         QFuture<qint32>, highestUpdateSequenceNumber, (HighestUsnOption option),
314         (const, override));
315
316     MOCK_METHOD(
317         QFuture<qint32>, highestUpdateSequenceNumber,
318         (qevercloud::Guid linkedNotebookGuid), (const, override));
319
320     MOCK_METHOD(ILocalStorageNotifier *, notifier, (), (const, override));
321 };
```

```
322
323 } // namespace quentier::local_storage::tests::mocks
```

6.34 QuentierLogger.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QDebug>
24 #include <QRegularExpression>
25 #include <QString>
26 #include <QTextStream>
27
28 namespace quentier {
29
34 enum class LogLevel
35 {
36     Trace,
37     Debug,
38     Info,
39     Warning,
40     Error
41 };
42
43 QUENTIER_EXPORT QDebug & operator<<(QDebug & dbg, LogLevel logLevel);
44
45 QUENTIER_EXPORT QTextStream & operator<<(QTextStream & strm, LogLevel logLevel);
46
54 void QUENTIER_EXPORT QuentierInitializeLogging();
55
59 void QUENTIER_EXPORT QuentierAddLogEntry(
60     const QString & sourceFileName, int sourceFileLineNumber,
61     const QString & component, const QString & message, LogLevel logLevel);
62
68 LogLevel QUENTIER_EXPORT QuentierMinLogLevel();
69
73 void QUENTIER_EXPORT QuentierSetMinLogLevel(LogLevel logLevel);
74
79 void QUENTIER_EXPORT QuentierAddStdOutLogDestination();
80
85 [[nodiscard]] bool QUENTIER_EXPORT QuentierIsLogLevelActive(LogLevel logLevel);
86
90 [[nodiscard]] QString QUENTIER_EXPORT QuentierLogFilesDirPath();
91
95 void QUENTIER_EXPORT QuentierRestartLogging();
96
100 [[nodiscard]] QRegularExpression QUENTIER_EXPORT QuentierLogComponentFilter();
101
105 void QUENTIER_EXPORT
106     QuentierSetLogComponentFilter(const QRegularExpression & filter);
107
108 } // namespace quentier
109
110 #define QNLOG_PRIVATE_BASE(component, message, level)                          \
111 if (quentier::QuentierIsLogLevelActive(quentier::LogLevel::level)) {       \
112 QString msg;                                                                \
113 QDebug dbg(&msg);                                                           \
114 dbg.nospace();                                                              \
115 dbg.noquote();                                                              \
116 dbg << message;                                                             \
117 quentier::QuentierAddLogEntry(                                              \
118 QStringLiteral(__FILE__), __LINE__, QString::fromUtf8(component),  \
119 msg, quentier::LogLevel::level);                                           \
120 }                                                                                  \
```

```
121 // QNLOG_PRIVATE_BASE
122
123 #define QNTRACE(component, message)                                          \
124 QNLOG_PRIVATE_BASE(component, message, Trace)                                \
125 // QNTRACE
126
127 #define QNDEBUG(component, message)                                          \
128 QNLOG_PRIVATE_BASE(component, message, Debug)                                \
129 // QNDEBUG
130
131 #define QNINFO(component, message)                                           \
132 QNLOG_PRIVATE_BASE(component, message, Info)                                 \
133 // QNINFO
134
135 #define QNWARNING(component, message)                                        \
136 QNLOG_PRIVATE_BASE(component, message, Warning)                              \
137 // QNWARNING
138
139 #define QNERROR(component, message)                                          \
140 QNLOG_PRIVATE_BASE(component, message, Error)                                \
141 // QNERROR
142
143 #define QUENTIER_SET_MIN_LOG_LEVEL(level)                                    \
144 quentier::QuentierSetMinLogLevel(quentier::LogLevel::level)
145   // QUENTIER_SET_MIN_LOG_LEVEL
146
147 #define QUENTIER_INITIALIZE_LOGGING() quentier::QuentierInitializeLogging()
148   // QUENTIER_INITIALIZE_LOGGING
149
150 // clang-format off
151 #define QUENTIER_ADD_STDOUT_LOG_DESTINATION()                                \
152 quentier::QuentierAddStdOutLogDestination()                                  \
153 // QUENTIER_ADD_STDOUT_LOG_DESTINATION
154 // clang-format on
155
156 #define QNLOG_FILE_LINENUMBER_DELIMITER ":"
```

6.35 INoteEditorBackend.h

```
1  /*
2  * Copyright 2016-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/local_storage/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23 #include <quentier/utility/Printable.h>
24
25 #include <QPalette>
26 #include <QPrinter>
27 #include <QStringList>
28 #include <QThread>
29 #include <QWidget>
30
31 class QUndoStack;
32
33 namespace quentier {
34
35 class Account;
36 class ErrorString;
37 class NoteEditor;
38 class SpellChecker;
39
40 class QUENTIER_EXPORT INoteEditorBackend
41 {
42 public:
43     virtual ~INoteEditorBackend() noexcept;
44
```

```
45      virtual void initialize(
46          local_storage::ILocalStoragePtr localStorage,
47          SpellChecker & spellChecker, const Account & account,
48          QThread * pBackgroundJobsThread) = 0;
49
50      [[nodiscard]] virtual QObject * object() = 0; // provide QObject interface
51      [[nodiscard]] virtual QWidget * widget() = 0; // provide QWidget interface
52
53      virtual void setAccount(const Account & account) = 0;
54      virtual void setUndoStack(QUndoStack * pUndoStack) = 0;
55
56      virtual void setInitialPageHtml(const QString & html) = 0;
57      virtual void setNoteNotFoundPageHtml(const QString & html) = 0;
58      virtual void setNoteDeletedPageHtml(const QString & html) = 0;
59      virtual void setNoteLoadingPageHtml(const QString & html) = 0;
60
61      [[nodiscard]] virtual bool isNoteLoaded() const = 0;
62      [[nodiscard]] virtual qint64 idleTime() const = 0;
63
64      virtual void convertToNote() = 0;
65      virtual void saveNoteToLocalStorage() = 0;
66      virtual void setNoteTitle(const QString & noteTitle) = 0;
67
68      virtual void setTagIds(
69          const QStringList & tagLocalUids, const QStringList & tagGuids) = 0;
70
71      virtual void undo() = 0;
72      virtual void redo() = 0;
73      virtual void cut() = 0;
74      virtual void copy() = 0;
75      virtual void paste() = 0;
76      virtual void pasteUnformatted() = 0;
77      virtual void selectAll() = 0;
78
79      virtual void formatSelectionAsSourceCode() = 0;
80
81      virtual void fontMenu() = 0;
82      virtual void textBold() = 0;
83      virtual void textItalic() = 0;
84      virtual void textUnderline() = 0;
85      virtual void textStrikethrough() = 0;
86      virtual void textHighlight() = 0;
87
88      virtual void alignLeft() = 0;
89      virtual void alignCenter() = 0;
90      virtual void alignRight() = 0;
91      virtual void alignFull() = 0;
92
93      [[nodiscard]] virtual QString selectedText() const = 0;
94      [[nodiscard]] virtual bool hasSelection() const = 0;
95
96      virtual void findNext(const QString & text, bool matchCase) const = 0;
97
98      virtual void findPrevious(const QString & text, bool matchCase) const = 0;
99
100     virtual void replace(
101         const QString & textToReplace, const QString & replacementText,
102         bool matchCase) = 0;
103
104     virtual void replaceAll(
105         const QString & textToReplace, const QString & replacementText,
106         bool matchCase) = 0;
107
108     virtual void insertToDoCheckbox() = 0;
109
110     virtual void insertInAppNoteLink(
111         const QString & userId, const QString & shardId,
112         const QString & noteGuid, const QString & linkText) = 0;
113
114     virtual void setSpellcheck(bool enabled) = 0;
115     [[nodiscard]] virtual bool spellCheckEnabled() const = 0;
116
117     virtual void setFont(const QFont & font) = 0;
118     virtual void setFontHeight(int height) = 0;
119     virtual void setFontColor(const QColor & color) = 0;
120     virtual void setBackgroundColor(const QColor & color) = 0;
121
122     [[nodiscard]] virtual QPalette defaultPalette() const = 0;
123     virtual void setDefaultPalette(const QPalette & pal) = 0;
124
125     [[nodiscard]] virtual const QFont * defaultFont() const = 0;
126     virtual void setDefaultFont(const QFont & font) = 0;
127
128     virtual void insertHorizontalLine() = 0;
129
130     virtual void increaseFontSize() = 0;
131     virtual void decreaseFontSize() = 0;
```

```
132
133     virtual void increaseIndentation() = 0;
134     virtual void decreaseIndentation() = 0;
135
136     virtual void insertBulletedList() = 0;
137     virtual void insertNumberedList() = 0;
138
139     virtual void insertTableDialog() = 0;
140
141     virtual void insertFixedWidthTable(
142         int rows, int columns, int widthInPixels) = 0;
143
144     virtual void insertRelativeWidthTable(
145         int rows, int columns, double relativeWidth) = 0;
146
147     virtual void insertTableRow() = 0;
148     virtual void insertTableColumn() = 0;
149     virtual void removeTableRow() = 0;
150     virtual void removeTableColumn() = 0;
151
152     virtual void addAttachmentDialog() = 0;
153     virtual void saveAttachmentDialog(const QByteArray & resourceHash) = 0;
154     virtual void saveAttachmentUnderCursor() = 0;
155     virtual void openAttachment(const QByteArray & resourceHash) = 0;
156     virtual void openAttachmentUnderCursor() = 0;
157     virtual void copyAttachment(const QByteArray & resourceHash) = 0;
158     virtual void copyAttachmentUnderCursor() = 0;
159     virtual void removeAttachment(const QByteArray & resourceHash) = 0;
160     virtual void removeAttachmentUnderCursor() = 0;
161     virtual void renameAttachment(const QByteArray & resourceHash) = 0;
162     virtual void renameAttachmentUnderCursor() = 0;
163
164     enum class Rotation
165     {
166         Clockwise,
167         Counterclockwise
168     };
169
170     friend QUENTIER_EXPORT QTextStream & operator<<(
171         QTextStream & strm, Rotation rotation);
172
173     friend QUENTIER_EXPORT QDebug & operator<<(QDebug & dbg, Rotation rotation);
174
175     virtual void rotateImageAttachment(
176         const QByteArray & resourceHash, Rotation rotationDirection) = 0;
177
178     virtual void rotateImageAttachmentUnderCursor(
179         Rotation rotationDirection) = 0;
180
181     virtual void encryptSelectedText() = 0;
182
183     virtual void decryptEncryptedTextUnderCursor() = 0;
184
185     virtual void decryptEncryptedText(
186         QString encryptedText, QString cipher, QString keyLength, QString hint,
187         QString enCryptIndex) = 0;
188
189     virtual void hideDecryptedTextUnderCursor() = 0;
190
191     virtual void hideDecryptedText(
192         QString encryptedText, QString decryptedText, QString cipher,
193         QString keyLength, QString hint, QString enDecryptedIndex) = 0;
194
195     virtual void editHyperlinkDialog() = 0;
196     virtual void copyHyperlink() = 0;
197     virtual void removeHyperlink() = 0;
198
199     virtual void onNoteLoadCancelled() = 0;
200
201     [[nodiscard]] virtual bool print(
202         QPrinter & printer, ErrorString & errorDescription) = 0;
203
204     [[nodiscard]] virtual bool exportToPdf(
205         const QString & absoluteFilePath, ErrorString & errorDescription) = 0;
206
207     [[nodiscard]] virtual bool exportToEnex(
208         const QStringList & tagNames, QString & enex,
209         ErrorString & errorDescription) = 0;
210
211     [[nodiscard]] virtual QString currentNoteLocalId() const = 0;
212     virtual void setCurrentNoteLocalId(const QString & noteLocalUid) = 0;
213
214     virtual void clear() = 0;
215
216     [[nodiscard]] virtual bool isModified() const = 0;
217     [[nodiscard]] virtual bool isEditorPageModified() const = 0;
218
```

```
219     virtual void setFocusToEditor() = 0;
220
221 protected:
222     INoteEditorBackend(NoteEditor * parent);
223     NoteEditor * m_pNoteEditor;
224 };
225
226 } // namespace quentier
```

6.36 NoteEditor.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/local_storage/Fwd.h>
22 #include <quentier/types/ErrorString.h>
23 #include <quentier/utility/Linkage.h>
24
25 #include <qevercloud/types/Note.h>
26 #include <qevercloud/types/Notebook.h>
27
28 #include <QPrinter>
29 #include <QStringList>
30 #include <QThread>
31 #include <QWidget>
32
33 class QUndoStack;
34
35 namespace quentier {
36
37 class Account;
38 class INoteEditorBackend;
39 class SpellChecker;
40
45 class QUENTIER_EXPORT NoteEditor :  public QWidget
46 {
47     Q_OBJECT
48 public:
49     explicit NoteEditor(
50         QWidget * parent = nullptr,
51 #if QT_VERSION >= QT_VERSION_CHECK(5, 15, 0)
52         Qt::WindowFlags flags = {});
53 #else
54         Qt::WindowFlags flags = 0); // NOLINT
55 #endif
56
57     ~NoteEditor() noexcept override;
58
75     void initialize(
76         local_storage::ILocalStoragePtr localStorage,
77         SpellChecker & spellChecker, const Account & account,
78         QThread * pBackgroundJobsThread = nullptr);
79
83     [[nodiscard]] INoteEditorBackend * backend() noexcept;
84
90     void setBackend(INoteEditorBackend * backend);
91
95     void setAccount(const Account & account);
96
100     [[nodiscard]] const QUndoStack * undoStack() const noexcept;
101
105     void setUndoStack(QUndoStack * pUndoStack);
106
110     void setInitialPageHtml(const QString & html);
111
116     void setNoteNotFoundPageHtml(const QString & html);
```

```
117
123     void setNoteDeletedPageHtml(const QString & html);
124
129     void setNoteLoadingPageHtml(const QString & html);
130
134     [[nodiscard]] QString currentNoteLocalId() const;
135
143     void setCurrentNoteLocalId(const QString & noteLocalId);
144
148     void clear();
149
154     [[nodiscard]] bool isModified() const noexcept;
155
160     [[nodiscard]] bool isEditorPageModified() const noexcept;
161
166     [[nodiscard]] bool isNoteLoaded() const noexcept;
167
173     [[nodiscard]] qint64 idleTime() const noexcept;
174
178     void setFocus();
179
180     [[nodiscard]] QString selectedText() const noexcept;
181     [[nodiscard]] bool hasSelection() const noexcept;
182
183     [[nodiscard]] bool spellCheckEnabled() const noexcept;
184
185     [[nodiscard]] bool print(
186         QPrinter & printer, ErrorString & errorDescription);
187
188     [[nodiscard]] bool exportToPdf(
189         const QString & absoluteFilePath, ErrorString & errorDescription);
190
191     [[nodiscard]] bool exportToEnex(
192         const QStringList & tagNames, QString & enex,
193         ErrorString & errorDescription);
194
202     [[nodiscard]] QPalette defaultPalette() const;
203
208     [[nodiscard]] const QFont * defaultFont() const;
209
210 Q_SIGNALS:
215     void contentChanged();
216
222     void noteAndNotebookFoundInLocalStorage(
223         qevercloud::Note note, qevercloud::Notebook notebook);
224
229     void noteNotFound(QString noteLocalId);
230
236     void noteDeleted(QString noteLocalId);
237
243     void noteModified();
244
249     void notifyError(ErrorString error);
250
255     void inAppNoteLinkClicked(
256         QString userId, QString shardId, QString noteGuid);
257
269     void inAppNoteLinkPasteRequested(
270         QString url, QString userId, QString shardId, QString noteGuid);
271
272     void convertedToNote(qevercloud::Note note);
273     void cantConvertToNote(ErrorString error);
274
275     void noteEditorHtmlUpdated(QString html);
276
277     void currentNoteChanged(qevercloud::Note note);
278
279     void spellCheckerNotReady();
280     void spellCheckerReady();
281
282     void noteLoaded();
283
290     void noteSavedToLocalStorage(QString noteLocalId);
291
296     void failedToSaveNoteToLocalStorage(
297         ErrorString errorDescription, QString noteLocalId);
298
299     // Signals to notify anyone interested of the formatting at the current
300     // cursor position
301     void textBoldState(bool state);
302     void textItalicState(bool state);
303     void textUnderlineState(bool state);
304     void textStrikethroughState(bool state);
305     void textAlignLeftState(bool state);
306     void textAlignCenterState(bool state);
307     void textAlignRightState(bool state);
308     void textAlignFullState(bool state);
```

```
309     void textInsideOrderedListState(bool state);
310     void textInsideUnorderedListState(bool state);
311     void textInsideTableState(bool state);
312 
313     void textFontFamilyChanged(QString fontFamily);
314     void textFontSizeChanged(int fontSize);
315 
316     void insertTableDialogRequested();
317 
318 public Q_SLOTS:
324     void convertToNote();
325 
334     void saveNoteToLocalStorage();
335 
345     void setNoteTitle(const QString & noteTitle);
346 
358     void setTagIds(
359         const QStringList & tagLocalIds, const QStringList & tagGuids);
360 
361     void undo();
362     void redo();
363     void cut();
364     void copy();
365     void paste();
366     void pasteUnformatted();
367     void selectAll();
368 
369     void formatSelectionAsSourceCode();
370 
371     void fontMenu();
372     void textBold();
373     void textItalic();
374     void textUnderline();
375     void textStrikethrough();
376     void textHighlight();
377 
378     void alignLeft();
379     void alignCenter();
380     void alignRight();
381     void alignFull();
382 
383     void findNext(const QString & text, bool matchCase) const;
384     void findPrevious(const QString & text, bool matchCase) const;
385 
386     void replace(
387         const QString & textToReplace, const QString & replacementText,
388         bool matchCase);
389 
390     void replaceAll(
391         const QString & textToReplace, const QString & replacementText,
392         bool matchCase);
393 
394     void insertToDoCheckbox();
395 
396     void insertInAppNoteLink(
397         const QString & userId, const QString & shardId,
398         const QString & noteGuid, const QString & linkText);
399 
400     void setSpellcheck(bool enabled);
401 
402     void setFont(const QFont & font);
403     void setFontHeight(int height);
404     void setFontColor(const QColor & color);
405     void setBackgroundColor(const QColor & color);
406 
422     void setDefaultPalette(const QPalette & pal);
423 
429     void setDefaultFont(const QFont & font);
430 
431     void insertHorizontalLine();
432 
433     void increaseFontSize();
434     void decreaseFontSize();
435 
436     void increaseIndentation();
437     void decreaseIndentation();
438 
439     void insertBulletedList();
440     void insertNumberedList();
441 
442     void insertTableDialog();
443 
444     void insertFixedWidthTable(int rows, int columns, int widthInPixels);
445 
446     void insertRelativeWidthTable(int rows, int columns, double relativeWidth);
447 
448     void insertTableRow();
```

```
449     void insertTableColumn();
450     void removeTableRow();
451     void removeTableColumn();
452 
453     void addAttachmentDialog();
454     void saveAttachmentDialog(const QByteArray & resourceHash);
455     void saveAttachmentUnderCursor();
456     void openAttachment(const QByteArray & resourceHash);
457     void openAttachmentUnderCursor();
458     void copyAttachment(const QByteArray & resourceHash);
459     void copyAttachmentUnderCursor();
460 
461     void encryptSelectedText();
462     void decryptEncryptedTextUnderCursor();
463 
464     void editHyperlinkDialog();
465     void copyHyperlink();
466     void removeHyperlink();
467 
468     void onNoteLoadCancelled();
469 
470 protected:
471     void dragMoveEvent(QDragMoveEvent * pEvent) override;
472     void dropEvent(QDropEvent * pEvent) override;
473 
474 private:
475     INoteEditorBackend * m_backend;
476 };
477 
478 } // namespace quentier
```

6.37 SpellChecker.h

```
1  /*
2  * Copyright 2017-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18 
19 #pragma once
20 
21 #include <quentier/utility/Linkage.h>
22 
23 #include <QList>
24 #include <QObject>
25 
26 #include <utility>
27 
28 namespace quentier {
29 
30 class Account;
31 class FileIOProcessorAsync;
32 class SpellCheckerPrivate;
33 
34 class QUENTIER_EXPORT SpellChecker :  public QObject
35 {
36     Q_OBJECT
37 public:
38     SpellChecker(
39         FileIOProcessorAsync * fileIOProcessorAsync, Account account,
40         QObject * parent = nullptr, const QString & userDictionaryPath = {});
41 
42     // The second bool in the pair indicates whether the dictionary
43     // is enabled or disabled
44     [[nodiscard]] QList<std::pair<QString, bool>> listAvailableDictionaries()
45         const;
46 
47     void setAccount(const Account & account);
48 
49     void enableDictionary(const QString & language);
50     void disableDictionary(const QString & language);
```

```
51
52      [[nodiscard]] bool checkSpell(const QString & word) const;
53
54      [[nodiscard]] QStringList spellCorrectionSuggestions(
55          const QString & misSpelledWord) const;
56
57      void addToUserWordlist(const QString & word);
58      void removeFromUserWordList(const QString & word);
59      void ignoreWord(const QString & word);
60      void removeWord(const QString & word);
61
62      [[nodiscard]] bool isReady() const noexcept;
63
64  Q_SIGNALS:
65      void ready();
66
67  private:
68      SpellCheckerPrivate * const d_ptr;
69      Q_DECLARE_PRIVATE(SpellChecker)
70  };
71
72  } // namespace quentier
```

6.38 IAuthenticator.h

```
1  /*
2  * Copyright 2022 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/types/Account.h>
23 #include <quentier/utility/Linkage.h>
24
25 #include <QFuture>
26
27 namespace quentier::synchronization {
28
29 class QUENTIER_EXPORT IAuthenticator
30 {
31 public:
32     virtual ~IAuthenticator() noexcept;
33
34     [[nodiscard]] virtual QFuture<IAuthenticationInfoPtr>
35         authenticateNewAccount() = 0;
36
37     [[nodiscard]] virtual QFuture<IAuthenticationInfoPtr> authenticateAccount(
38         Account account) = 0;
39 };
40
41 } // namespace quentier::synchronization
```

6.39 INoteStoreFactory.h

```
1  /*
2  * Copyright 2022-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
```

```
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <qevercloud/Fwd.h>
24 #include <qevercloud/services/Fwd.h>
25 #include <qevercloud/types/TypeAliases.h>
26
27 #include <optional>
28
29 namespace quentier::synchronization {
30
31 class QUENTIER_EXPORT INoteStoreFactory
32 {
33 public:
34     virtual ~INoteStoreFactory();
35
36     [[nodiscard]] virtual qevercloud::INoteStorePtr createNoteStore(
37         QString noteStoreUrl = {},
38         std::optional<qevercloud::Guid> linkedNotebookGuid = {},
39         qevercloud::IRequestContextPtr ctx = {},
40         qevercloud::IRetryPolicyPtr retryPolicy = {}) = 0;
41 };
42
43 } // namespace quentier::synchronization
```

6.40 ISyncConflictResolver.h

```
1 /*
2 * Copyright 2021-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <qevercloud/types/Note.h>
24 #include <qevercloud/types/Notebook.h>
25 #include <qevercloud/types/SavedSearch.h>
26 #include <qevercloud/types/Tag.h>
27
28 #include <QFuture>
29
30 #include <variant>
31
32 class QDebug;
33 class QTextStream;
34
35 namespace quentier::synchronization {
36
41 class QUENTIER_EXPORT ISyncConflictResolver
42 {
43 public:
48     struct QUENTIER_EXPORT ConflictResolution
49     {
54         struct QUENTIER_EXPORT UseTheirs
55         {};
56
61         struct QUENTIER_EXPORT UseMine
62         {};
63
```

```
69          struct QUENTIER_EXPORT IgnoreMine
70          {};
71
80          template <class T>
81          struct MoveMine
82          {
83              using value_type = T;
84
88              T mine;
89          };
90      };
91
92      using NotebookConflictResolution = std::variant<
93          ConflictResolution::UseTheirs, ConflictResolution::UseMine,
94          ConflictResolution::IgnoreMine,
95          ConflictResolution::MoveMine<qevercloud::Notebook>>;
96
97      using NoteConflictResolution = std::variant<
98          ConflictResolution::UseTheirs, ConflictResolution::UseMine,
99          ConflictResolution::IgnoreMine,
100          ConflictResolution::MoveMine<qevercloud::Note>>;
101
102     using SavedSearchConflictResolution = std::variant<
103         ConflictResolution::UseTheirs, ConflictResolution::UseMine,
104         ConflictResolution::IgnoreMine,
105         ConflictResolution::MoveMine<qevercloud::SavedSearch>>;
106
107     using TagConflictResolution = std::variant<
108         ConflictResolution::IgnoreMine, ConflictResolution::UseTheirs,
109         ConflictResolution::UseMine,
110         ConflictResolution::MoveMine<qevercloud::Tag>>;
111
112 public:
113     virtual ~ISyncConflictResolver() noexcept;
114
115     [[nodiscard]] virtual QFuture<NotebookConflictResolution>
116         resolveNotebookConflict(
117             qevercloud::Notebook theirs, qevercloud::Notebook mine) = 0;
118
119     [[nodiscard]] virtual QFuture<NoteConflictResolution> resolveNoteConflict(
120         qevercloud::Note theirs, qevercloud::Note mine) = 0;
121
122     [[nodiscard]] virtual QFuture<SavedSearchConflictResolution>
123         resolveSavedSearchConflict(
124             qevercloud::SavedSearch theirs, qevercloud::SavedSearch mine) = 0;
125
126     [[nodiscard]] virtual QFuture<TagConflictResolution> resolveTagConflict(
127         qevercloud::Tag theirs, qevercloud::Tag mine) = 0;
128 };
129
130 QUENTIER_EXPORT QTextStream & operator<<(
131     QTextStream & strm,
132     const ISyncConflictResolver::NotebookConflictResolution & resolution);
133
134 QUENTIER_EXPORT QDebug & operator<<(
135     QDebug & dbg,
136     const ISyncConflictResolver::NotebookConflictResolution & resolution);
137
138 QUENTIER_EXPORT QTextStream & operator<<(
139     QTextStream & strm,
140     const ISyncConflictResolver::NoteConflictResolution & resolution);
141
142 QUENTIER_EXPORT QDebug & operator<<(
143     QDebug & dbg,
144     const ISyncConflictResolver::NoteConflictResolution & resolution);
145
146 QUENTIER_EXPORT QTextStream & operator<<(
147     QTextStream & strm,
148     const ISyncConflictResolver::SavedSearchConflictResolution & resolution);
149
150 QUENTIER_EXPORT QDebug & operator<<(
151     QDebug & dbg,
152     const ISyncConflictResolver::SavedSearchConflictResolution & resolution);
153
154 QUENTIER_EXPORT QTextStream & operator<<(
155     QTextStream & strm,
156     const ISyncConflictResolver::TagConflictResolution & resolution);
157
158 QUENTIER_EXPORT QDebug & operator<<(
159     QDebug & dbg,
160     const ISyncConflictResolver::TagConflictResolution & resolution);
161
162 } // namespace quentier::synchronization
```

6.41 ISyncEventsNotifier.h

```
1 /*
2 * Copyright 2023-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/Fwd.h>
22 #include <quentier/synchronization/types/Fwd.h>
23 #include <quentier/utility/Linkage.h>
24
25 #include <qevercloud/types/LinkedNotebook.h>
26
27 #include <QList>
28 #include <QObject>
29
30 namespace quentier::synchronization {
31
32 class QUENTIER_EXPORT ISyncEventsNotifier :  public QObject
33 {
34     Q_OBJECT
35 protected:
36     explicit ISyncEventsNotifier(QObject * parent = nullptr);
37
38 public:
39     ~ISyncEventsNotifier() override;
40
41 Q_SIGNALS:
58     void syncChunksDownloadProgress(
59         qint32 highestDownloadedUsn, qint32 highestServerUsn,
60         qint32 lastPreviousUsn);
61
66     void syncChunksDownloaded();
67
72     void syncChunksDataProcessingProgress(ISyncChunksDataCountersPtr counters);
73
81     void startLinkedNotebooksDataDownloading(
82         const QList<qevercloud::LinkedNotebook> & linkedNotebooks);
83
103     void linkedNotebookSyncChunksDownloadProgress(
104         qint32 highestDownloadedUsn, qint32 highestServerUsn,
105         qint32 lastPreviousUsn,
106         const qevercloud::LinkedNotebook & linkedNotebook);
107
114     void linkedNotebookSyncChunksDownloaded(
115         const qevercloud::LinkedNotebook & linkedNotebook);
116
125     void linkedNotebookSyncChunksDataProcessingProgress(
126         ISyncChunksDataCountersPtr counters,
127         const qevercloud::LinkedNotebook & linkedNotebook);
128
138     void notesDownloadProgress(
139         quint32 notesDownloaded, quint32 totalNotesToDownload);
140
151     void linkedNotebookNotesDownloadProgress(
152         quint32 notesDownloaded, quint32 totalNotesToDownload,
153         const qevercloud::LinkedNotebook & linkedNotebook);
154
165     void resourcesDownloadProgress(
166         quint32 resourcesDownloaded, quint32 totalResourcesToDownload);
167
179     void linkedNotebookResourcesDownloadProgress(
180         quint32 resourcesDownloaded, quint32 totalResourcesToDownload,
181         const qevercloud::LinkedNotebook & linkedNotebook);
182
191     void downloadFinished(bool dataDownloaded);
192
199     void userOwnSendStatusUpdate(ISendStatusPtr sendStatus);
200
209     void linkedNotebookSendStatusUpdate(
210         const qevercloud::Guid & linkedNotebookGuid, ISendStatusPtr sendStatus);
211 };
```

```
212
213 } // namespace quentier::synchronization
```

6.42 ISynchronizer.h

```
1 /*
2 * Copyright 2021-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/local_storage/Fwd.h>
22 #include <quentier/synchronization/Fwd.h>
23 #include <quentier/synchronization/types/Fwd.h>
24 #include <quentier/types/Account.h>
25 #include <quentier/utility/Linkage.h>
26 #include <quentier/utility/cancelers/Fwd.h>
27
28 #include <qevercloud/types/TypeAliases.h>
29
30 #include <QFuture>
31
32 #include <memory>
33 #include <utility>
34
35 namespace quentier {
36
37 class Account;
38
39 } // namespace quentier
40
41 namespace quentier::synchronization {
42
43 class QUENTIER_EXPORT ISynchronizer
44 {
45 public:
46     virtual ~ISynchronizer() noexcept;
47
48     [[nodiscard]] virtual QFuture<std::pair<Account, IAuthenticationInfoPtr»
49         authenticateNewAccount() = 0;
50
51     [[nodiscard]] virtual QFuture<IAuthenticationInfoPtr> authenticateAccount(
52         Account account) = 0;
53
54     using SyncResult =
55         std::pair<QFuture<ISyncResultPtr>, ISyncEventsNotifier *>;
56
57     [[nodiscard]] virtual SyncResult synchronizeAccount(
58         Account account, local_storage::ILocalStoragePtr localStorage,
59         utility::cancelers::ICancelerPtr canceler,
60         ISyncOptionsPtr options = nullptr,
61         ISyncConflictResolverPtr syncConflictResolver = nullptr) = 0;
62
63     virtual void revokeAuthentication(qevercloud::UserID userId) = 0;
64 };
65
66 } // namespace quentier::synchronization
```

6.43 ISyncStateStorage.h

```
1 /*
2 * Copyright 2020-2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
```

```
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/types/Account.h>
23 #include <quentier/utility/Linkage.h>
24
25 #include <QObject>
26
27 namespace quentier::synchronization {
28
34 class QUENTIER_EXPORT ISyncStateStorage :  public QObject
35 {
36     Q_OBJECT
37 protected:
38     explicit ISyncStateStorage(QObject * parent = nullptr);
39
40 public:
41     ~ISyncStateStorage() override;
42
43     [[nodiscard]] virtual ISyncStatePtr getSyncState(
44         const Account & account) = 0;
45
46     virtual void setSyncState(
47         const Account & account, ISyncStatePtr syncState) = 0;
48
49 Q_SIGNALS:
55     void notifySyncStateUpdated(Account account, ISyncStatePtr syncState);
56 };
57
58 } // namespace quentier::synchronization
```

6.44 IUserStoreFactory.h

```
1  /*
2  * Copyright 2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <qevercloud/Fwd.h>
24 #include <qevercloud/services/Fwd.h>
25
26 #include <QString>
27
28 namespace quentier::synchronization {
29
30 class QUENTIER_EXPORT IUserStoreFactory
31 {
32 public:
33     virtual ~IUserStoreFactory();
34
35     [[nodiscard]] virtual qevercloud::IUserStorePtr createUserStore(
36         QString userStoreUrl = {}, qevercloud::IRequestContextPtr ctx = {},
37         qevercloud::IRetryPolicyPtr retryPolicy = {}) = 0;
```

```
38 };
39
40 } // namespace quentier::synchronization
```

6.45 MockIAuthenticator.h

```
1 /*
2 * Copyright 2022 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/IAuthenticator.h>
22
23 #include <gmock/gmock.h>
24
25 namespace quentier::synchronization::tests::mocks {
26
27 class MockIAuthenticator :  public IAuthenticator
28 {
29 public:
30     MOCK_METHOD(
31         QFuture<IAuthenticationInfoPtr>, authenticateNewAccount, (),
32         (override));
33
34     MOCK_METHOD(
35         QFuture<IAuthenticationInfoPtr>, authenticateAccount, (Account account),
36         (override));
37 };
38
39 } // namespace quentier::synchronization::tests::mocks
```

6.46 MockINoteStoreFactory.h

```
1 /*
2 * Copyright 2022-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/INoteStoreFactory.h>
22
23 #include <gmock/gmock.h>
24
25 namespace quentier::synchronization::tests::mocks {
26
27 class MockINoteStoreFactory :  public INoteStoreFactory
28 {
29 public:
30     MOCK_METHOD(
31         ::qevercloud::INoteStorePtr, createNoteStore,
```

```
32         (QString noteStoreUrl,
33          std::optional<::qevercloud::Guid> linkedNotebookGuid,
34          ::qevercloud::IRequestContextPtr ctx,
35          ::qevercloud::IRetryPolicyPtr retryPolicy),
36         (override));
37 };
38
39 } // namespace quentier::synchronization::tests::mocks
```

6.47 MockISyncConflictResolver.h

```
1 /*
2 * Copyright 2022-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/ISyncConflictResolver.h>
22
23 #include <gmock/gmock.h>
24
25 namespace quentier::synchronization::tests::mocks {
26
27 class MockISyncConflictResolver :  public ISyncConflictResolver
28 {
29 public:
30     MOCK_METHOD(
31         QFuture<NotebookConflictResolution>, resolveNotebookConflict,
32         (::qevercloud::Notebook theirs, ::qevercloud::Notebook mine),
33         (override));
34
35     MOCK_METHOD(
36         QFuture<NoteConflictResolution>, resolveNoteConflict,
37         (::qevercloud::Note theirs, ::qevercloud::Note mine), (override));
38
39     MOCK_METHOD(
40         QFuture<SavedSearchConflictResolution>, resolveSavedSearchConflict,
41         (::qevercloud::SavedSearch theirs, ::qevercloud::SavedSearch mine),
42         (override));
43
44     MOCK_METHOD(
45         QFuture<TagConflictResolution>, resolveTagConflict,
46         (::qevercloud::Tag theirs, ::qevercloud::Tag mine), (override));
47 };
48
49 } // namespace quentier::synchronization::tests::mocks
```

6.48 MockISyncStateStorage.h

```
1 /*
2 * Copyright 2022 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
```

```
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/ISyncStateStorage.h>
22
23 #include <gmock/gmock.h>
24
25 namespace quentier::synchronization::tests::mocks {
26
27 class MockISyncStateStorage :  public ISyncStateStorage
28 {
29     Q_OBJECT
30 public:
31     MOCK_METHOD(
32         ISyncStatePtr, getSyncState, (const Account & account), (override));
33
34     MOCK_METHOD(
35         void, setSyncState, (const Account & account, ISyncStatePtr syncState),
36         (override));
37 };
38
39 } // namespace quentier::synchronization::tests::mocks
```

6.49 Errors.h

```
1 /*
2 * Copyright 2022-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QtGlobal>
24
25 #include <optional>
26 #include <variant>
27
28 namespace quentier::synchronization {
29
36 struct QUENTIER_EXPORT RateLimitReachedError
37 {
44     std::optional<qint32> rateLimitDurationSec;
45 };
46
47 [[nodiscard]] QUENTIER_EXPORT bool operator==(
48     const RateLimitReachedError & lhs,
49     const RateLimitReachedError & rhs) noexcept;
50
51 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
52     const RateLimitReachedError & lhs,
53     const RateLimitReachedError & rhs) noexcept;
54
60 struct QUENTIER_EXPORT AuthenticationExpiredError
61 {};
62
63 [[nodiscard]] QUENTIER_EXPORT bool operator==(
64     const AuthenticationExpiredError & lhs,
65     const AuthenticationExpiredError & rhs) noexcept;
66
67 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
68     const AuthenticationExpiredError & lhs,
69     const AuthenticationExpiredError & rhs) noexcept;
70
76 using StopSynchronizationError = std::variant<
77     RateLimitReachedError, AuthenticationExpiredError, std::monostate>;
78
79 } // namespace quentier::synchronization
```

6.50 IAuthenticationInfo.h

```
1 /*
2 * Copyright 2022-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23 #include <quentier/utility/Printable.h>
24
25 #include <qevercloud/types/TypeAliases.h>
26
27 #include <QList>
28 #include <QNetworkCookie>
29 #include <QString>
30
31 namespace quentier::synchronization {
32
37 class QUENTIER_EXPORT IAuthenticationInfo :  public Printable
38 {
39 public:
43     [[nodiscard]] virtual qevercloud::UserID userId() const = 0;
44
48     [[nodiscard]] virtual QString authToken() const = 0;
49
53     [[nodiscard]] virtual qevercloud::Timestamp authTokenExpirationTime()
54         const = 0;
55
59     [[nodiscard]] virtual qevercloud::Timestamp authenticationTime() const = 0;
60
65     [[nodiscard]] virtual QString shardId() const = 0;
66
70     [[nodiscard]] virtual QString noteStoreUrl() const = 0;
71
76     [[nodiscard]] virtual QString webApiUrlPrefix() const = 0;
77
84     [[nodiscard]] virtual QList<QNetworkCookie> userStoreCookies() const = 0;
85 };
86
87 } // namespace quentier::synchronization
```

6.51 IAuthenticationInfoBuilder.h

```
1 /*
2 * Copyright 2023 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <qevercloud/types/TypeAliases.h>
```

```
25
26 #include <QList>
27 #include <QNetworkCookie>
28 #include <QString>
29
30 namespace quentier::synchronization {
31
32 class QUENTIER_EXPORT IAuthenticationInfoBuilder
33 {
34 public:
35     virtual ~IAuthenticationInfoBuilder() noexcept;
36
37     virtual IAuthenticationInfoBuilder & setUserId(
38         qevercloud::UserID userId) = 0;
39
40     virtual IAuthenticationInfoBuilder & setAuthToken(QString token) = 0;
41
42     virtual IAuthenticationInfoBuilder & setAuthTokenExpirationTime(
43         qevercloud::Timestamp expirationTime) = 0;
44
45     virtual IAuthenticationInfoBuilder & setAuthenticationTime(
46         qevercloud::Timestamp authenticationTime) = 0;
47
48     virtual IAuthenticationInfoBuilder & setShardId(QString shardId) = 0;
49
50     virtual IAuthenticationInfoBuilder & setNoteStoreUrl(
51         QString noteStoreUrl) = 0;
52
53     virtual IAuthenticationInfoBuilder & setWebApiUrlPrefix(
54         QString webApiUrlPrefix) = 0;
55
56     virtual IAuthenticationInfoBuilder & setUserStoreCookies(
57         QList<QNetworkCookie> cookies) = 0;
58
59     [[nodiscard]] virtual IAuthenticationInfoPtr build() = 0;
60 };
61
62 [[nodiscard]] QUENTIER_EXPORT IAuthenticationInfoBuilderPtr
63     createAuthenticationInfoBuilder();
64
65 } // namespace quentier::synchronization
```

6.52 IDownloadNotesStatus.h

```
1 /*
2 * Copyright 2022-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Errors.h>
22 #include <quentier/synchronization/types/Fwd.h>
23 #include <quentier/utility/Linkage.h>
24 #include <quentier/utility/Printable.h>
25
26 #include <qevercloud/types/Note.h>
27 #include <qevercloud/types/TypeAliases.h>
28
29 #include <QException>
30 #include <QList>
31
32 #include <memory>
33 #include <utility>
34
35 namespace quentier::synchronization {
36
41 class QUENTIER_EXPORT IDownloadNotesStatus :  public Printable
42 {
43 public:
```

```
44      using QExceptionPtr = std::shared_ptr<QException>;
45      using NoteWithException = std::pair<qevercloud::Note, QExceptionPtr>;
46      using GuidWithException = std::pair<qevercloud::Guid, QExceptionPtr>;
47      using UpdateSequenceNumbersByGuid = QHash<qevercloud::Guid, qint32>;
48
49      [[nodiscard]] virtual quint64 totalNewNotes() const = 0;
50      [[nodiscard]] virtual quint64 totalUpdatedNotes() const = 0;
51      [[nodiscard]] virtual quint64 totalExpungedNotes() const = 0;
52
53      [[nodiscard]] virtual QList<NoteWithException> notesWhichFailedToDownload()
54          const = 0;
55
56      [[nodiscard]] virtual QList<NoteWithException> notesWhichFailedToProcess()
57          const = 0;
58
59      [[nodiscard]] virtual QList<GuidWithException>
60          noteGuidsWhichFailedToExpunge() const = 0;
61
62      [[nodiscard]] virtual UpdateSequenceNumbersByGuid
63          processedNoteGuidsAndUsns() const = 0;
64
65      [[nodiscard]] virtual UpdateSequenceNumbersByGuid
66          cancelledNoteGuidsAndUsns() const = 0;
67
68      [[nodiscard]] virtual QList<qevercloud::Guid> expungedNoteGuids() const = 0;
69
70      [[nodiscard]] virtual StopSynchronizationError stopSynchronizationError()
71          const = 0;
72  };
73
74  } // namespace quentier::synchronization
```

6.53 IDownloadResourcesStatus.h

```
1  /*
2  * Copyright 2022 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Errors.h>
22 #include <quentier/utility/Linkage.h>
23 #include <quentier/utility/Printable.h>
24
25 #include <qevercloud/types/Resource.h>
26 #include <qevercloud/types/TypeAliases.h>
27
28 #include <QException>
29
30 #include <memory>
31 #include <utility>
32
33 namespace quentier::synchronization {
34
35 class QUENTIER_EXPORT IDownloadResourcesStatus :  public Printable
36 {
37 public:
38     ~IDownloadResourcesStatus() noexcept override;
39
40     using QExceptionPtr = std::shared_ptr<QException>;
41
42     using ResourceWithException =
43         std::pair<qevercloud::Resource, QExceptionPtr>;
44
45     using UpdateSequenceNumbersByGuid = QHash<qevercloud::Guid, qint32>;
46
47     [[nodiscard]] virtual quint64 totalNewResources() const = 0;
48     [[nodiscard]] virtual quint64 totalUpdatedResources() const = 0;
49
```

```
50      [[nodiscard]] virtual QList<ResourceWithException>
51          resourcesWhichFailedToDownload() const = 0;
52
53      [[nodiscard]] virtual QList<ResourceWithException>
54          resourcesWhichFailedToProcess() const = 0;
55
56      [[nodiscard]] virtual UpdateSequenceNumbersByGuid
57          processedResourceGuidsAndUsns() const = 0;
58
59      [[nodiscard]] virtual UpdateSequenceNumbersByGuid
60          cancelledResourceGuidsAndUsns() const = 0;
61
62      [[nodiscard]] virtual StopSynchronizationError stopSynchronizationError()
63          const = 0;
64 };
65
66 } // namespace quentier::synchronization
```

6.54 ISendStatus.h

```
1  /*
2  * Copyright 2022-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Errors.h>
22 #include <quentier/utility/Linkage.h>
23 #include <quentier/utility/Printable.h>
24
25 #include <qevercloud/types/Note.h>
26 #include <qevercloud/types/Notebook.h>
27 #include <qevercloud/types/SavedSearch.h>
28 #include <qevercloud/types/Tag.h>
29 #include <qevercloud/types/TypeAliases.h>
30
31 #include <QException>
32 #include <QList>
33
34 #include <memory>
35 #include <utility>
36
37 namespace quentier::synchronization {
38
44 class QUENTIER_EXPORT ISendStatus :  public Printable
45 {
46 public:
47     using QExceptionPtr = std::shared_ptr<QException>;
48
49     using NoteWithException = std::pair<qevercloud::Note, QExceptionPtr>;
50
51     using NotebookWithException =
52         std::pair<qevercloud::Notebook, QExceptionPtr>;
53
54     using SavedSearchWithException =
55         std::pair<qevercloud::SavedSearch, QExceptionPtr>;
56
57     using TagWithException = std::pair<qevercloud::Tag, QExceptionPtr>;
58
59 public:
60     // Total
61
65     [[nodiscard]] virtual quint64 totalAttemptedToSendNotes() const = 0;
66
70     [[nodiscard]] virtual quint64 totalAttemptedToSendNotebooks() const = 0;
71
75     [[nodiscard]] virtual quint64 totalAttemptedToSendSavedSearches() const = 0;
76
80     [[nodiscard]] virtual quint64 totalAttemptedToSendTags() const = 0;
```

```
81
82      // Notes
83
87      [[nodiscard]] virtual quint64 totalSuccessfullySentNotes() const = 0;
88
93      [[nodiscard]] virtual QList<NoteWithException> failedToSendNotes()
94          const = 0;
95
96      // Notebooks
97
101      [[nodiscard]] virtual quint64 totalSuccessfullySentNotebooks() const = 0;
102
107      [[nodiscard]] virtual QList<NotebookWithException> failedToSendNotebooks()
108          const = 0;
109
110      // Saved searches
111
115      [[nodiscard]] virtual quint64 totalSuccessfullySentSavedSearches()
116          const = 0;
117
122      [[nodiscard]] virtual QList<SavedSearchWithException>
123          failedToSendSavedSearches() const = 0;
124
125      // Tags
126
130      [[nodiscard]] virtual quint64 totalSuccessfullySentTags() const = 0;
131
136      [[nodiscard]] virtual QList<TagWithException> failedToSendTags() const = 0;
137
138      // General
139
145      [[nodiscard]] virtual StopSynchronizationError stopSynchronizationError()
146          const = 0;
147
154      [[nodiscard]] virtual bool needToRepeatIncrementalSync() const = 0;
155 };
156
157 } // namespace quentier::synchronization
```

6.55 ISyncChunksDataCounters.h

```
1  /*
2  * Copyright 2021-2023 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22 #include <quentier/utility/Printable.h>
23
24 #include <QtGlobal>
25
26 namespace quentier::synchronization {
27
33 struct QUENTIER_EXPORT ISyncChunksDataCounters :  public Printable
34 {
35     // ================ Saved searches ================
36
40     [[nodiscard]] virtual quint64 totalSavedSearches() const noexcept = 0;
41
45     [[nodiscard]] virtual quint64 totalExpungedSavedSearches()
46         const noexcept = 0;
47
52     [[nodiscard]] virtual quint64 addedSavedSearches() const noexcept = 0;
53
58     [[nodiscard]] virtual quint64 updatedSavedSearches() const noexcept = 0;
59
64     [[nodiscard]] virtual quint64 expungedSavedSearches() const noexcept = 0;
65
```

```
66      // ================= Tags =================
67
71      [[nodiscard]] virtual quint64 totalTags() const noexcept = 0;
72
76      [[nodiscard]] virtual quint64 totalExpungedTags() const noexcept = 0;
77
81      [[nodiscard]] virtual quint64 addedTags() const noexcept = 0;
82
86      [[nodiscard]] virtual quint64 updatedTags() const noexcept = 0;
87
91      [[nodiscard]] virtual quint64 expungedTags() const noexcept = 0;
92
93      // ================= Linked notebooks =================
94
98      [[nodiscard]] virtual quint64 totalLinkedNotebooks() const noexcept = 0;
99
103     [[nodiscard]] virtual quint64 totalExpungedLinkedNotebooks()
104         const noexcept = 0;
105
110     [[nodiscard]] virtual quint64 addedLinkedNotebooks() const noexcept = 0;
111
116     [[nodiscard]] virtual quint64 updatedLinkedNotebooks() const noexcept = 0;
117
122     [[nodiscard]] virtual quint64 expungedLinkedNotebooks() const noexcept = 0;
123
124     // ================= Notebooks =================
125
129     [[nodiscard]] virtual quint64 totalNotebooks() const noexcept = 0;
130
134     [[nodiscard]] virtual quint64 totalExpungedNotebooks() const noexcept = 0;
135
139     [[nodiscard]] virtual quint64 addedNotebooks() const noexcept = 0;
140
144     [[nodiscard]] virtual quint64 updatedNotebooks() const noexcept = 0;
145
150     [[nodiscard]] virtual quint64 expungedNotebooks() const noexcept = 0;
151 };
152
153 } // namespace quentier::synchronization
```

6.56 ISyncOptions.h

```
1  /*
2  * Copyright 2022-2023 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22 #include <quentier/utility/Printable.h>
23
24 #include <qevercloud/Fwd.h>
25
26 #include <QDir>
27 #include <QtGlobal>
28
29 #include <optional>
30
31 namespace quentier::synchronization {
32
36 class QUENTIER_EXPORT ISyncOptions :  public Printable
37 {
38 public:
39     ~ISyncOptions() noexcept override;
40
46     [[nodiscard]] virtual bool downloadNoteThumbnails() const = 0;
47
62     [[nodiscard]] virtual std::optional<QDir> inkNoteImagesStorageDir()
63         const = 0;
```

```
64
69      [[nodiscard]] virtual qevercloud::IRequestContextPtr requestContext()
70          const = 0;
71
76      [[nodiscard]] virtual qevercloud::IRetryPolicyPtr retryPolicy() const = 0;
77
84      [[nodiscard]] virtual std::optional<quint32> maxConcurrentNoteDownloads()
85          const = 0;
86
93      [[nodiscard]] virtual std::optional<quint32>
94          maxConcurrentResourceDownloads() const = 0;
95  };
96
97  } // namespace quentier::synchronization
```

6.57 ISyncOptionsBuilder.h

```
1  /*
2  * Copyright 2022-2023 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <qevercloud/Fwd.h>
25
26 #include <QDir>
27
28 #include <optional>
29
30 namespace quentier::synchronization {
31
32 class QUENTIER_EXPORT ISyncOptionsBuilder
33 {
34 public:
35     virtual ~ISyncOptionsBuilder() noexcept;
36
37     virtual ISyncOptionsBuilder & setDownloadNoteThumbnails(bool value) = 0;
38
39     virtual ISyncOptionsBuilder & setInkNoteImagesStorageDir(
40         std::optional<QDir> dir) = 0;
41
42     virtual ISyncOptionsBuilder & setRequestContext(
43         qevercloud::IRequestContextPtr ctx) = 0;
44
45     virtual ISyncOptionsBuilder & setRetryPolicy(
46         qevercloud::IRetryPolicyPtr retryPolicy) = 0;
47
48     virtual ISyncOptionsBuilder & setMaxConcurrentNoteDownloads(
49         std::optional<quint32> maxConcurrentNoteDownloads) = 0;
50
51     virtual ISyncOptionsBuilder & setMaxConcurrentResourceDownloads(
52         std::optional<quint32> maxConcurrentResourceDownloads) = 0;
53
54     [[nodiscard]] virtual ISyncOptionsPtr build() = 0;
55 };
56
57 [[nodiscard]] QUENTIER_EXPORT ISyncOptionsBuilderPtr createSyncOptionsBuilder();
58
59 } // namespace quentier::synchronization
```

6.58 ISyncResult.h

```
1  /*
```

```
2 * Copyright 2022-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/Fwd.h>
22 #include <quentier/synchronization/types/Errors.h>
23 #include <quentier/synchronization/types/Fwd.h>
24 #include <quentier/utility/Linkage.h>
25 #include <quentier/utility/Printable.h>
26
27 #include <qevercloud/types/TypeAliases.h>
28
29 #include <QHash>
30 #include <QSet>
31
32 namespace quentier::synchronization {
33
34 class QUENTIER_EXPORT ISyncResult :  public Printable
35 {
36 public:
37     [[nodiscard]] virtual ISyncStatePtr syncState() const = 0;
38
39     [[nodiscard]] virtual ISyncChunksDataCountersPtr
40         userAccountSyncChunksDataCounters() const = 0;
41
42     [[nodiscard]] virtual QHash<qevercloud::Guid, ISyncChunksDataCountersPtr>
43         linkedNotebookSyncChunksDataCounters() const = 0;
44
45     [[nodiscard]] virtual bool userAccountSyncChunksDownloaded() const = 0;
46
47     [[nodiscard]] virtual QSet<qevercloud::Guid>
48         linkedNotebookGuidsWithSyncChunksDownloaded() const = 0;
49
50     [[nodiscard]] virtual IDownloadNotesStatusPtr
51         userAccountDownloadNotesStatus() const = 0;
52
53     [[nodiscard]] virtual QHash<qevercloud::Guid, IDownloadNotesStatusPtr>
54         linkedNotebookDownloadNotesStatuses() const = 0;
55
56     [[nodiscard]] virtual IDownloadResourcesStatusPtr
57         userAccountDownloadResourcesStatus() const = 0;
58
59     [[nodiscard]] virtual QHash<qevercloud::Guid, IDownloadResourcesStatusPtr>
60         linkedNotebookDownloadResourcesStatuses() const = 0;
61
62     [[nodiscard]] virtual ISendStatusPtr userAccountSendStatus() const = 0;
63
64     [[nodiscard]] virtual QHash<qevercloud::Guid, ISendStatusPtr>
65         linkedNotebookSendStatuses() const = 0;
66
67     [[nodiscard]] virtual StopSynchronizationError stopSynchronizationError()
68         const = 0;
69 };
70
71 } // namespace quentier::synchronization
```

6.59 ISyncState.h

```
1 /*
2 * Copyright 2022 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
```

```
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
 */

#pragma once

#include <quentier/utility/Linkage.h>
#include <quentier/utility/Printable.h>

#include <qevercloud/types/TypeAliases.h>

#include <QHash>
#include <QString>

namespace quentier::synchronization {

class QUENTIER_EXPORT ISyncState :  public Printable
{
public:
    [[nodiscard]] virtual qint32 userDataUpdateCount() const = 0;

    [[nodiscard]] virtual qevercloud::Timestamp userDataLastSyncTime()
        const = 0;

    [[nodiscard]] virtual QHash<qevercloud::Guid, qint32>
        linkedNotebookUpdateCounts() const = 0;

    [[nodiscard]] virtual QHash<qevercloud::Guid, qevercloud::Timestamp>
        linkedNotebookLastSyncTimes() const = 0;
};

} // namespace quentier::synchronization
```

6.60 ISyncStateBuilder.h

```
/*
 * Copyright 2023 Dmitry Ivanov
 *
 * This file is part of libquentier
 *
 * libquentier is free software; you can redistribute it and/or modify
 * it under the terms of the GNU Lesser General Public License as published by
 * the Free Software Foundation, version 3 of the License.
 *
 * libquentier is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU Lesser General Public License for more details.
 *
 * You should have received a copy of the GNU Lesser General Public License
 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
 */

#pragma once

#include <quentier/synchronization/types/Fwd.h>
#include <quentier/utility/Linkage.h>

#include <qevercloud/types/TypeAliases.h>

#include <QHash>

namespace quentier::synchronization {

class QUENTIER_EXPORT ISyncStateBuilder
{
public:
    virtual ~ISyncStateBuilder() noexcept;

    virtual ISyncStateBuilder & setUserDataUpdateCount(qint32 updateCount) = 0;

    virtual ISyncStateBuilder & setUserDataLastSyncTime(
        qevercloud::Timestamp lastSyncTime) = 0;

    virtual ISyncStateBuilder & setLinkedNotebookUpdateCounts(
        QHash<qevercloud::Guid, qint32> updateCounts) = 0;

    virtual ISyncStateBuilder & setLinkedNotebookLastSyncTimes(
        QHash<qevercloud::Guid, qevercloud::Timestamp> lastSyncTimes) = 0;
```

```
45
46     [[nodiscard]] virtual ISyncStatePtr build() = 0;
47 };
48
49 [[nodiscard]] QUENTIER_EXPORT ISyncStateBuilderPtr createSyncStateBuilder();
50
51 } // namespace quentier::synchronization
```

6.61 AuthenticationInfo.h

```
1 /*
2 * Copyright 2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QJsonObject>
25
26 namespace quentier::synchronization {
27
31 [[nodiscard]] QJsonObject QUENTIER_EXPORT
32     serializeAuthenticationInfoToJson(const IAuthenticationInfo & info);
33
39 [[nodiscard]] IAuthenticationInfoPtr QUENTIER_EXPORT
40     deserializeAuthenticationInfoFromJson(const QJsonObject & json);
41
42 } // namespace quentier::synchronization
```

6.62 DownloadNotesStatus.h

```
1 /*
2 * Copyright 2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QJsonObject>
25
26 namespace quentier::synchronization {
27
31 [[nodiscard]] QJsonObject QUENTIER_EXPORT
32     serializeDownloadNotesStatusToJson(const IDownloadNotesStatus & status);
33
39 [[nodiscard]] IDownloadNotesStatusPtr QUENTIER_EXPORT
40     deserializeDownloadNotesStatusFromJson(const QJsonObject & json);
41
42 } // namespace quentier::synchronization
```

6.63 DownloadResourcesStatus.h

```
1 /*
2 * Copyright 2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QJsonObject>
25
26 namespace quentier::synchronization {
27
31 [[nodiscard]] QJsonObject QUENTIER_EXPORT
32     serializeDownloadResourcesStatusToJson(
33         const IDownloadResourcesStatus & status);
34
40 [[nodiscard]] IDownloadResourcesStatusPtr QUENTIER_EXPORT
41     deserializeDownloadResourcesStatusFromJson(const QJsonObject & json);
42
43 } // namespace quentier::synchronization
```

6.64 SendStatus.h

```
1 /*
2 * Copyright 2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QJsonObject>
25
26 namespace quentier::synchronization {
27
31 [[nodiscard]] QJsonObject QUENTIER_EXPORT
32     serializeSendStatusToJson(const ISendStatus & sendStatus);
33
39 [[nodiscard]] ISendStatusPtr QUENTIER_EXPORT
40     deserializeSendStatusFromJson(const QJsonObject & json);
41
42 } // namespace quentier::synchronization
```

6.65 SyncChunksDataCounters.h

```
1 /*
2 * Copyright 2024 Dmitry Ivanov
```

```
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QJsonObject>
25
26 namespace quentier::synchronization {
27
31 [[nodiscard]] QJsonObject QUENTIER_EXPORT serializeSyncChunksDataCountersToJson(
32     const ISyncChunksDataCounters & counters);
33
39 [[nodiscard]] ISyncChunksDataCountersPtr QUENTIER_EXPORT
40     deserializeSyncChunksDataCountersFromJson(const QJsonObject & json);
41
42 } // namespace quentier::synchronization
```

6.66 SyncResult.h

```
1 /*
2 * Copyright 2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QJsonObject>
25
26 namespace quentier::synchronization {
27
31 [[nodiscard]] QJsonObject QUENTIER_EXPORT
32     serializeSyncResultToJson(const ISyncResult & syncResult);
33
39 [[nodiscard]] ISyncResultPtr QUENTIER_EXPORT
40     deserializeSyncResultFromJson(const QJsonObject & json);
41
42 } // namespace quentier::synchronization
```

6.67 SyncState.h

```
1 /*
2 * Copyright 2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
```

```
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/synchronization/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QJsonObject>
25
26 namespace quentier::synchronization {
27
31 [[nodiscard]] QJsonObject QUENTIER_EXPORT
32     serializeSyncStateToJson(const ISyncState & syncState);
33
39 [[nodiscard]] ISyncStatePtr QUENTIER_EXPORT
40     deserializeSyncStateFromJson(const QJsonObject & json);
41
42 } // namespace quentier::synchronization
```

6.68 Future.h

```
1 /*
2 * Copyright 2021-2022 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QAbstractEventDispatcher>
24 #include <QFuture>
25 #include <QFutureWatcher>
26 #include <QMutex>
27 #include <QMutexLocker>
28 #include <QObject>
29 #include <QPointer>
30
31 #include <quentier/threading/QtFutureContinuations.h>
32
33 #if QT_VERSION >= QT_VERSION_CHECK(6, 0, 0)
34 #include <QPromise>
35 #include <exception>
36 #else
37 #include <quentier/threading/Qt5Promise.h>
38 #endif
39
40 #include <algorithm>
41 #include <cmath>
42 #include <memory>
43 #include <type_traits>
44 #include <utility>
45
46 namespace quentier::threading {
47
51 template <class T>
52 [[nodiscard]] std::enable_if_t<
53     std::negation_v<std::is_same<std::decay_t<T>, void»,
54     QFuture<std::decay_t<T»>
55     makeReadyFuture(T t)
56 {
```

```
57     QPromise<std::decay_t<T>> promise;
58     QFuture<std::decay_t<T» future = promise.future();
59
60     promise.start();
61     promise.addResult(std::move(t));
62     promise.finish();
63
64     return future;
65 }
66
67 [[nodiscard]] QFuture<void> QUENTIER_EXPORT makeReadyFuture();
68
73 template <class T, class E>
74 [[nodiscard]] std::enable_if_t<std::is_base_of_v<QException, E>, QFuture<T»
75     makeExceptionalFuture(const E & e)
76 {
77     QPromise<std::decay_t<T>> promise;
78     QFuture<std::decay_t<T» future = promise.future();
79
80     promise.start();
81     promise.setException(e);
82     promise.finish();
83
84     return future;
85 }
86
87 #if QT_VERSION >= QT_VERSION_CHECK(6, 0, 0)
92 template <class T>
93 [[nodiscard]] QFuture<T> makeExceptionalFuture(std::exception_ptr e)
94 {
95     QPromise<std::decay_t<T>> promise;
96     QFuture<std::decay_t<T» future = promise.future();
97
98     promise.start();
99     promise.setException(std::move(e));
100     promise.finish();
101
102     return future;
103 }
104 #endif // QT_VERSION
105
110 template <class T, class U>
111 void bindCancellation(const QFuture<T> & from, QFuture<U> to)
112 {
113     auto watcher = std::make_unique<QFutureWatcher<T»();
114     auto * rawWatcher = watcher.get();
115
116     QObject::connect(
117         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
118         [rawWatcher, to]() mutable {
119             to.cancel();
120             rawWatcher->deleteLater();
121         });
122
123     QObject::connect(
124         rawWatcher, &QFutureWatcher<T>::finished, rawWatcher,
125         [rawWatcher] { rawWatcher->deleteLater(); });
126
127     watcher->setFuture(from);
128     Q_UNUSED(watcher.release());
129 }
130
138 [[nodiscard]] QFuture<void> QUENTIER_EXPORT
139     whenAll(QList<QFuture<void» futures);
140
150 template <class T>
151 [[nodiscard]] std::enable_if_t<
152     !std::is_void_v<std::decay_t<T», QFuture<QList<std::decay_t<T»»
153     whenAll(QList<QFuture<std::decay_t<T»> futures)
154 {
155     if (Q_UNLIKELY(futures.isEmpty())) {
156         return makeReadyFuture<QList<std::decay_t<T»>({});
157     }
158
159     auto promise = std::make_shared<QPromise<QList<std::decay_t<T»»();
160     auto future = promise->future();
161
162     for (auto & f:  futures) {
163         threading::bindCancellation(future, f);
164     }
165
166     const auto totalItemCount = futures.size();
167     promise->setProgressRange(0, static_cast<int>(totalItemCount));
168     promise->setProgressValue(0);
169
170     promise->start();
171
```

```
172     auto resultIndexedList =
173         std::make_shared<QList<std::pair<int, std::decay_t<T>>>>();
174
175     auto processedItemsCount = std::make_shared<int>(0);
176     auto exceptionFlag = std::make_shared<bool>(false);
177     auto mutex = std::make_shared<QMutex>();
178
179     for (int i = 0; i < futures.size(); ++i) {
180         auto & f = futures[i];
181         auto thenFuture = then(
182             std::move(f),
183             [promise, processedItemsCount, totalItemCount, exceptionFlag, mutex,
184              resultIndexedList, i](std::decay_t<T> result) {
185                 if (promise->isCanceled()) {
186                     return;
187                 }
188
189                 int count = 0;
190                 {
191                     const QMutexLocker locker{mutex.get()};
192
193                     if (*exceptionFlag) {
194                         return;
195                     }
196
197                     ++(*processedItemsCount);
198                     count = *processedItemsCount;
199                     promise->setProgressValue(count);
200
201                     resultIndexedList->append(
202                         std::make_pair(i, std::move(result)));
203                 }
204
205                 if (count == totalItemCount) {
206                     std::sort(
207                         resultIndexedList->begin(), resultIndexedList->end(),
208                         [](const auto & lhs, const auto & rhs) {
209                             return lhs.first < rhs.first;
210                         });
211
212                     auto resultList =
213                         std::make_shared<QList<std::decay_t<T>>>();
214                     resultList->reserve(resultIndexedList->size());
215                     for (auto & [i, v]:  *resultIndexedList) {
216                         resultList->append(std::move(v));
217                     }
218
219                     promise->addResult(*resultList);
220                     promise->finish();
221                 }
222             });
223
224         onFailed(
225             std::move(thenFuture),
226             [promise, mutex, exceptionFlag](const QException & e) {
227                 if (promise->isCanceled()) {
228                     return;
229                 }
230
231                 {
232                     const QMutexLocker locker{mutex.get()};
233
234                     if (*exceptionFlag) {
235                         return;
236                     }
237
238                     *exceptionFlag = true;
239                 }
240
241                 promise->setException(e);
242                 promise->finish();
243             });
244     }
245
246     return future;
247 }
248
254 template <class T, class U>
255 void mapFutureProgress(
256     const QFuture<T> & future, const std::shared_ptr<QPromise<U>> & promise)
257 {
258     const auto futureProgressMinimum = future.progressMinimum();
259     const auto futureProgressRange =
260         future.progressMaximum() - futureProgressMinimum;
261
262     Q_ASSERT(futureProgressRange >= 0);
263
```

```
264     const auto promiseFuture = promise->future();
265     const auto promiseProgressMinimum = promiseFuture.progressMinimum();
266     const auto promiseProgressMaximum = promiseFuture.progressMaximum();
267 
268     const auto promiseProgressRange =
269         promiseProgressMaximum - promiseProgressMinimum;
270 
271     Q_ASSERT(promiseProgressRange >= 0);
272 
273     auto futureWatcher = std::make_unique<QFutureWatcher<T»();
274 
275     QObject::connect(
276         futureWatcher.get(), &QFutureWatcher<T>::progressValueChanged,
277         futureWatcher.get(),
278         [promise, futureProgressMinimum, futureProgressRange,
279          promiseProgressRange, promiseProgressMinimum,
280          promiseProgressMaximum](int progressValue) {
281             if (Q_UNLIKELY(futureProgressRange == 0)) {
282                 promise->setProgressValue(0);
283                 return;
284             }
285 
286             const auto progressPart =
287                 static_cast<double>(progressValue - futureProgressMinimum) /
288                 static_cast<double>(futureProgressRange);
289 
290             const auto mappedProgressValue = static_cast<int>(
291                 std::round(progressPart * promiseProgressRange));
292 
293             promise->setProgressValue(std::clamp(
294                 promiseProgressMinimum + mappedProgressValue,
295                 promiseProgressMinimum, promiseProgressMaximum));
296         });
297 
298     QObject::connect(
299         futureWatcher.get(), &QFutureWatcher<T>::finished, futureWatcher.get(),
300         [futureWatcherWeak = QPointer<QFutureWatcher<T»(futureWatcher.get())] {
301             if (!futureWatcherWeak.isNull()) {
302                 futureWatcherWeak->deleteLater();
303             }
304         });
305 
306     QObject::connect(
307         futureWatcher.get(), &QFutureWatcher<T>::canceled, futureWatcher.get(),
308         [futureWatcherWeak = QPointer<QFutureWatcher<T»(futureWatcher.get())] {
309             if (!futureWatcherWeak.isNull()) {
310                 futureWatcherWeak->deleteLater();
311             }
312         });
313 
314     futureWatcher->setFuture(future);
315     Q_UNUSED(futureWatcher.release());
316 }
317 
318 } // namespace quentier::threading
```

6.69 Post.h

```
1  /*
2  * Copyright 2021-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18 
19 #pragma once
20 
21 #include <QAbstractEventDispatcher>
22 #include <QMetaObject>
23 #include <QObject>
24 
25 #include <QThread>
```

```
26
27 #include <memory>
28 #include <utility>
29
30 namespace quentier::threading {
31
32 template <typename Function>
33 void postToObject(QObject * object, Function && function)
34 {
35     Q_ASSERT(object);
36
37     QMetaObject::invokeMethod(
38         object, std::forward<Function>(function), Qt::QueuedConnection);
39 }
40
41 template <typename Function>
42 void postToThread(QThread * pThread, Function && function)
43 {
44     Q_ASSERT(pThread);
45     Q_ASSERT(!pThread->isFinished());
46
47     QObject * pObject = QAbstractEventDispatcher::instance(pThread);
48     if (!pObject) {
49         // Thread's event loop has not been started yet.  Create a dummy QObject,
50         // move it to the target thread, set things up so that it would be
51         // destroyed after the job is done and use postToObject.
52         auto pDummyObj = std::make_unique<QObject>();
53         pDummyObj->moveToThread(pThread);
54         postToObject(
55             pDummyObj.get(),
56             [pObj = pDummyObj.get(),
57              function = std::forward<Function>(function)]() mutable {
58                 pObj->deleteLater();
59                 function();
60             });
61         Q_UNUSED(pDummyObj.release()) // NOLINT
62         return;
63     }
64
65     if (pThread == QThread::currentThread()) {
66         // Already on the target thread, executing the function right away
67         function();
68         return;
69     }
70
71     QMetaObject::invokeMethod(pObject, std::forward<Function>(function));
72 }
73
74 } // namespace quentier::threading
```

6.70 Qt5Promise.h

```
1 /*
2 * Copyright 2021-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <QFutureInterface>
22
23 #include <type_traits>
24
25 // Partial backport of QPromise from Qt6 to Qt5
26 template <typename T>
27 class QPromise
28 {
29     static_assert(
30         std::is_copy_constructible_v<T> || std::is_move_constructible_v<T> ||
31             std::is_same_v<T, void>,
```

```
32          "Type with copy or move constructors or type void is required");
33
34  public:
35      QPromise() = default;
36
37      Q_DISABLE_COPY(QPromise)
38
39      QPromise(QPromise<T> && other) noexcept :  d(other.d)
40      {
41          other.d = QFutureInterface<T>();
42      }
43
44      QPromise(QFutureInterface<T> & other) :  d(other) {}
45
46      QPromise & operator=(QPromise<T> && other) noexcept
47      {
48          QPromise<T> tmp(std::move(other));
49          tmp.swap(*this);
50          return *this;
51      }
52
53      ~QPromise()
54      {
55          // If QFutureInterface has no state, there is nothing to be done
56          if (d.queryState(QFutureInterfaceBase::State::NoState)) {
57              return;
58          }
59
60          // Otherwise, if computation is not finished at this point, cancel
61          // potential waits
62          if (!d.queryState(QFutureInterfaceBase::State::Finished)) {
63              d.cancel();
64              finish(); // required to finalize the state
65          }
66      }
67
68      // Core QPromise APIs
69      QFuture<T> future()const
70  {
71          return d.future();
72      }
73      template <
74          typename U,
75          typename = std::enable_if_t<
76              std::is_same_v<U, T> || std::is_convertible_v<U, T>>>
77      void addResult(U && result, int index = -1)
78      {
79          d.reportResult(std::forward<U>(result), index);
80      }
81
82      void setException(const QException & e)
83      {
84          d.reportException(e);
85      }
86
87      void start()
88      {
89          d.reportStarted();
90      }
91      void finish()
92      {
93          d.reportFinished();
94      }
95
96      void suspendIfRequested()
97      {
98          d.suspendIfRequested();
99      }
100
101      bool isCanceled()const
102  {
103          return d.isCanceled();
104      }
105
106      // Progress methods
107      void setProgressRange(int minimum, int maximum)
108      {
109          d.setProgressRange(minimum, maximum);
110      }
111      void setProgressValue(int progressValue)
112      {
113          d.setProgressValue(progressValue);
114      }
115      void setProgressValueAndText(
116          int progressValue, const QString & progressText)
117      {
118          d.setProgressValueAndText(progressValue, progressText);
```

```
119     }
120
121     void swap(QPromise<T> & other) noexcept
122     {
123         qSwap(this->d, other.d);
124     }
125
126 private:
127     mutable QFutureInterface<T> d = QFutureInterface<T>();
128 };
129
130 template <typename T>
131 inline void swap(QPromise<T> & a, QPromise<T> & b) noexcept
132 {
133     a.swap(b);
134 }
```

6.71 QtFutureContinuations.h

```
1 /*
2 * Copyright 2021-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <QtGlobal>
22
23 #include <QFutureWatcher>
24 #include <QRunnable>
25 #include <QThreadPool>
26 #include <quentier/exception/RuntimeError.h>
27 #include <quentier/threading/Post.h>
28 #include <quentier/threading/QtFutureHelpers.h>
29
30 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
31 #include <quentier/threading/Qt5Promise.h>
32 #endif // QT_VERSION
33
34 #include <quentier/threading/Runnable.h>
35
36 #include <boost/core/demangle.hpp>
37
38 #include <memory>
39 #include <type_traits>
40 #include <typeinfo>
41
42 namespace quentier::threading {
43
44 // NOTE: "native" implementation of continuations for Qt6 is currently disabled
45 // due to bugs in their implementation, in particular (but not limited to)
46 // https://bugreports.qt.io/browse/QTBUG-119579 and
47 // https://bugreports.qt.io/browse/QTBUG-117918.  It's a shame but it is what it
48 // is.
49 /*
50 #if QT_VERSION >= QT_VERSION_CHECK(6, 0, 0)
51
52 template <class T, class Function>
53 QFuture<typename QtPrivate::ResultTypeHelper<Function, T>::ResultType> then(
54 QFuture<T> && future, Function && function)
55 {
56 return future.then(std::forward<decltype(function)>(function));
57 }
58
59 template <class T, class Function>
60 QFuture<typename QtPrivate::ResultTypeHelper<Function, T>::ResultType> then(
61 QFuture<T> && future, QtFuture::Launch policy, Function && function)
62 {
63 return future.then(policy, std::forward<decltype(function)>(function));
64 }
```

```
65
66 template <class T, class Function>
67 QFuture<typename QtPrivate::ResultTypeHelper<Function, T>::ResultType> then(
68 QFuture<T> && future, QThreadPool * pool, Function && function)
69 {
70 return future.then(pool, std::forward<decltype(function)>(function));
71 }
72
73 template <class T, class Function>
74 QFuture<typename QtPrivate::ResultTypeHelper<Function, T>::ResultType> then(
75 QFuture<T> && future, QObject * context, Function && function)
76 {
77 return future.then(context, std::forward<decltype(function)>(function));
78 }
79
80 template <class T, class Function>
81 std::enable_if_t<!QtPrivate::ArgResolver<Function>::HasExtraArgs, QFuture<T»
82 onFailed(QFuture<T> && future, Function && handler)
83 {
84 return future.onFailed(std::forward<decltype(handler)>(handler));
85 }
86
87 template <class T, class Function>
88 std::enable_if_t<!QtPrivate::ArgResolver<Function>::HasExtraArgs, QFuture<T»
89 onFailed(QFuture<T> && future, QObject * context, Function && handler)
90 {
91 return future.onFailed(context, std::forward<decltype(handler)>(handler));
92 }
93
94 #else // QT_VERSION
95
96 // implementation for Qt5
97 */
98
99 namespace detail {
100
101 template <class T, class Function>
102 void processParentFuture(
103     std::shared_ptr<
104         QPromise<typename ResultTypeHelper<Function, T>::ResultType»
105         promise,
106     QFuture<T> && future, Function && function)
107 {
108     Q_ASSERT(promise);
109
110     using ResultType = typename ResultTypeHelper<Function, T>::ResultType;
111
112     promise->start();
113
114     // If future contains exception, just forward it to the promise and
115     // don't call the function at all
116     try {
117         future.waitForFinished();
118     }
119     catch (const QException & e) {
120         promise->setException(e);
121         promise->finish();
122         return;
123     }
124     // NOTE: there cannot be other exception types in this context in Qt5
125     // because exception store can only contain QExceptions
126
127     // Try to run the handler, in case of success forward the result to promise
128     // (unless it is void), catch possible exceptions and if caught put them
129     // to the promise
130     try {
131         if constexpr (std::is_void_v<ResultType>) {
132             if constexpr (std::is_void_v<T>) {
133                 function();
134             }
135             else {
136                 if (future.resultCount() == 0) {
137                     promise->setException(RuntimeError{ErrorString{
138                         QString::fromUtf8(
139                             "Invalid future continuation:  detected future "
140                             "without result for type %1")
141                             .arg(QString::fromStdString(std::string{
142                                 boost::core::demangle(typeid(T).name())}))}});
143                     promise->finish();
144                     return;
145                 }
146
147                 function(future.result());
148             }
149         }
150         else {
151             if constexpr (std::is_void_v<T>) {
```

```
152                 promise->addResult(function());
153             }
154             else {
155                 promise->addResult(function(future.result()));
156             }
157         }
158     }
159     catch (const QException & e) {
160         promise->setException(e);
161     }
162     catch (const std::exception & e) {
163         ErrorString error{QT_TRANSLATE_NOOP(
164             "utility", "Unknown std::exception in then future handler")};
165         error.details() = QString::fromStdString(std::string{e.what()});
166         promise->setException(RuntimeError{std::move(error)});
167     }
168     catch (...)  {
169         ErrorString error{QT_TRANSLATE_NOOP(
170             "utility", "Unknown exception in then future handler")};
171         promise->setException(RuntimeError{std::move(error)});
172     }
173
174     promise->finish();
175 }
176
177 } // namespace detail
178
179 template <class T, class Function>
180 QFuture<typename detail::ResultTypeHelper<Function, T>::ResultType> then(
181     QFuture<T> && future, Function && function)
182 {
183     using ResultType =
184         typename detail::ResultTypeHelper<Function, T>::ResultType;
185
186     auto promise = std::make_shared<QPromise<ResultType»();
187     auto result = promise->future();
188
189     if (future.isFinished()) {
190         detail::processParentFuture(
191             std::move(promise), std::move(future),
192             std::forward<decltype(function)>(function));
193         return result;
194     }
195
196     auto watcher = std::make_unique<QFutureWatcher<T»();
197     auto * rawWatcher = watcher.get();
198     QObject::connect(
199         rawWatcher, &QFutureWatcher<T>::finished, rawWatcher,
200         [rawWatcher, function = std::forward<decltype(function)>(function),
201          promise = std::move(promise)]() mutable {
202             detail::processParentFuture(
203                 std::move(promise), rawWatcher->future(),
204                 std::forward<decltype(function)>(function));
205             rawWatcher->deleteLater();
206         });
207
208     QObject::connect(
209         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
210         [rawWatcher] { rawWatcher->deleteLater(); });
211
212     watcher->setFuture(std::move(future));
213     Q_UNUSED(watcher.release())
214
215     return result;
216 }
217
218 template <class T, class Function>
219 QFuture<typename detail::ResultTypeHelper<Function, T>::ResultType> then(
220     QFuture<T> && future, QtFuture::Launch policy, Function && function)
221 {
222     if (policy == QtFuture::Launch::Sync) {
223         return then(
224             std::move(future), std::forward<decltype(function)>(function));
225     }
226
227     return then(
228         std::move(future), QThreadPool::globalInstance(),
229         std::forward<decltype(function)>(function));
230 }
231
232 template <class T, class Function>
233 QFuture<typename detail::ResultTypeHelper<Function, T>::ResultType> then(
234     QFuture<T> && future, QThreadPool * pool, Function && function)
235 {
236     using ResultType =
237         typename detail::ResultTypeHelper<Function, T>::ResultType;
238
```

```
239     auto promise = std::make_shared<QPromise<ResultType»();
240     auto result = promise->future();
241 
242     if (future.isFinished()) {
243         auto * runnable = createFunctionRunnable(
244             [future = std::move(future), promise = std::move(promise),
245              function = std::forward<decltype(function)>(function)]() mutable {
246                 detail::processParentFuture(
247                     std::move(promise), std::move(future),
248                     std::forward<decltype(function)>(function));
249             });
250         runnable->setAutoDelete(true);
251         pool->start(runnable);
252         return result;
253     }
254 
255     auto watcher = std::make_unique<QFutureWatcher<T»();
256     auto * rawWatcher = watcher.get();
257     QObject::connect(
258         rawWatcher, &QFutureWatcher<T>::finished, rawWatcher,
259         [rawWatcher, function = std::forward<decltype(function)>(function),
260          promise = std::move(promise), pool]() mutable {
261             auto * runnable = createFunctionRunnable(
262                 [function = std::forward<decltype(function)>(function),
263                  promise = std::move(promise),
264                  future = rawWatcher->future()]() mutable {
265                     detail::processParentFuture(
266                         std::move(promise), std::move(future),
267                         std::forward<decltype(function)>(function));
268                 });
269             runnable->setAutoDelete(true);
270             pool->start(runnable);
271             rawWatcher->deleteLater();
272         });
273 
274     QObject::connect(
275         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
276         [rawWatcher] { rawWatcher->deleteLater(); });
277 
278     watcher->setFuture(std::move(future));
279     Q_UNUSED(watcher.release())
280 
281     return result;
282 }
283 
284 template <class T, class Function>
285 QFuture<typename detail::ResultTypeHelper<Function, T>::ResultType> then(
286     QFuture<T> && future, QObject * context, Function && function)
287 {
288     using ResultType =
289         typename detail::ResultTypeHelper<Function, T>::ResultType;
290 
291     auto promise = std::make_shared<QPromise<ResultType»();
292     auto result = promise->future();
293 
294     if (future.isFinished()) {
295         postToObject(
296             context,
297             [future = std::move(future), promise = std::move(promise),
298              function = std::forward<decltype(function)>(function)]() mutable {
299                 detail::processParentFuture(
300                     std::move(promise), std::move(future),
301                     std::forward<decltype(function)>(function));
302             });
303         return result;
304     }
305 
306     auto watcher = std::make_unique<QFutureWatcher<T»();
307     auto * rawWatcher = watcher.get();
308 
309     QObject::connect(
310         rawWatcher, &QFutureWatcher<T>::finished, context,
311         [context, rawWatcher,
312          function = std::forward<decltype(function)>(function),
313          promise = std::move(promise)]() mutable {
314             postToObject(
315                 context,
316                 [function = std::forward<decltype(function)>(function),
317                  promise = std::move(promise),
318                  future = rawWatcher->future()]() mutable {
319                     detail::processParentFuture(
320                         std::move(promise), std::move(future),
321                         std::forward<decltype(function)>(function));
322                 });
323             rawWatcher->deleteLater();
324         });
325 
```

```
326     QObject::connect(
327         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
328         [rawWatcher] { rawWatcher->deleteLater(); });
329 
330     watcher->setFuture(std::move(future));
331     Q_UNUSED(watcher.release())
332 
333     return result;
334 }
335 
336 namespace detail {
337 
338 template <class T, class Function>
339 std::enable_if_t<!QtPrivate::ArgResolver<Function>::HasExtraArgs, void>
340     processPossibleFutureException(
341         std::shared_ptr<QPromise<T>> promise, QFuture<T> && future,
342         Function && handler)
343 {
344     Q_ASSERT(promise);
345 
346     using ArgType = typename QtPrivate::ArgResolver<Function>::First;
347     using ResultType =
348         typename ResultTypeHelper<Function, std::decay_t<ArgType»::ResultType;
349     static_assert(std::is_convertible_v<ResultType, T>);
350 
351     promise->start();
352 
353     try {
354         try {
355             future.waitForFinished();
356         }
357         catch (const ArgType & e) {
358             try {
359                 if constexpr (std::is_void_v<ResultType>) {
360                     handler(e);
361                 }
362                 else {
363                     promise->addResult(handler(e));
364                 }
365             }
366             catch (const QException & e) {
367                 promise->setException(e);
368             }
369             catch (const std::exception & e) {
370                 ErrorString error{QT_TRANSLATE_NOOP(
371                     "utility",
372                     "Unknown std::exception in onFailed future handler")};
373                 error.details() = QString::fromStdString(std::string{e.what()});
374                 promise->setException(RuntimeError{std::move(error)});
375             }
376             catch (...)  {
377                 ErrorString error{QT_TRANSLATE_NOOP(
378                     "utility", "Unknown exception in onFailed future handler")};
379                 promise->setException(RuntimeError{std::move(error)});
380             }
381         }
382     }
383     // Exception doesn't match with handler's argument type, propagate
384     // the exception to be handled later.
385     catch (const QException & e) {
386         promise->setException(e);
387     }
388     catch (const std::exception & e) {
389         ErrorString error{QT_TRANSLATE_NOOP(
390             "utility",
391             "Unknown std::exception which did not match with onFailed "
392             "future handler")};
393         error.details() = QString::fromStdString(std::string{e.what()});
394         promise->setException(RuntimeError{std::move(error)});
395     }
396     catch (...)  {
397         ErrorString error{QT_TRANSLATE_NOOP(
398             "utility",
399             "Unknown which did not match with onFailed "
400             "future handler")};
401         promise->setException(RuntimeError{std::move(error)});
402     }
403 
404     promise->finish();
405 }
406 
407 } // namespace detail
408 
409 // WARNING! "Chaining" of onFailed calls would only work properly with Qt5 if
410 // all involved exceptions subclass QException.  It is due to the way exception
411 // storage is implemented in Qt5.  In Qt6 is was made to store std::exception_ptr
412 // so there's no requirement to use QExceptions in Qt6.
```

```
413
414 template <class T, class Function>
415 std::enable_if_t<!QtPrivate::ArgResolver<Function>::HasExtraArgs, QFuture<T»
416     onFailed(QFuture<T> && future, Function && handler)
417 {
418     auto promise = std::make_shared<QPromise<T»();
419     auto result = promise->future();
420
421     if (future.isFinished()) {
422         detail::processPossibleFutureException(
423             std::move(promise), std::move(future),
424             std::forward<decltype(handler)>(handler));
425         return result;
426     }
427
428     auto watcher = std::make_unique<QFutureWatcher<T»();
429     auto * rawWatcher = watcher.get();
430     QObject::connect(
431         rawWatcher, &QFutureWatcher<T>::finished, rawWatcher,
432         [rawWatcher, promise = std::move(promise),
433          handler = std::forward<decltype(handler)>(handler)]() mutable {
434             auto future = rawWatcher->future();
435             rawWatcher->deleteLater();
436             detail::processPossibleFutureException(
437                 std::move(promise), std::move(future),
438                 std::forward<decltype(handler)>(handler));
439         });
440
441     QObject::connect(
442         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
443         [rawWatcher] { rawWatcher->deleteLater(); });
444
445     watcher->setFuture(std::move(future));
446     Q_UNUSED(watcher.release())
447
448     return result;
449 }
450
451 template <class T, class Function>
452 std::enable_if_t<!QtPrivate::ArgResolver<Function>::HasExtraArgs, QFuture<T»
453     onFailed(QFuture<T> && future, QObject * context, Function && handler)
454 {
455     auto promise = std::make_shared<QPromise<T»();
456     auto result = promise->future();
457
458     if (future.isFinished()) {
459         postToObject(
460             context,
461             [promise = std::move(promise), future = std::move(future),
462              handler = std::forward<decltype(handler)>(handler)]() mutable {
463                 detail::processPossibleFutureException(
464                     std::move(promise), std::move(future),
465                     std::forward<decltype(handler)>(handler));
466             });
467         return result;
468     }
469
470     auto watcher = std::make_unique<QFutureWatcher<T»();
471     auto * rawWatcher = watcher.get();
472     QObject::connect(
473         rawWatcher, &QFutureWatcher<T>::finished, context,
474         [context, rawWatcher, promise = std::move(promise),
475          handler = std::forward<decltype(handler)>(handler)]() mutable {
476             postToObject(
477                 context,
478                 [promise = std::move(promise), future = rawWatcher->future(),
479                  handler = std::forward<decltype(handler)>(handler)]() mutable {
480                     detail::processPossibleFutureException(
481                         std::move(promise), std::move(future),
482                         std::forward<decltype(handler)>(handler));
483                 });
484             rawWatcher->deleteLater();
485         });
486
487     QObject::connect(
488         rawWatcher, &QFutureWatcher<T>::canceled, rawWatcher,
489         [rawWatcher] { rawWatcher->deleteLater(); });
490
491     watcher->setFuture(std::move(future));
492     Q_UNUSED(watcher.release())
493
494     return result;
495 }
496
497 // #endif // QT_VERSION
498
499 // Convenience functions for both Qt versions
```

```
500
501 template <class T, class U, class Function>
502 void thenOrFailed(
503     QFuture<T> && future, std::shared_ptr<QPromise<U» promise,
504     Function && function)
505 {
506     auto thenFuture =
507         then(std::move(future), std::forward<decltype(function)>(function));
508
509     onFailed(std::move(thenFuture), [promise](const QException & e) {
510         promise->setException(e);
511         promise->finish();
512     });
513 }
514
515 template <class T, class U, class Function>
516 void thenOrFailed(
517     QFuture<T> && future, QThread * thread,
518     std::shared_ptr<QPromise<U>> promise, Function && function)
519 {
520     auto thenFuture =
521         then(std::move(future), thread, std::forward<Function>(function));
522
523     onFailed(std::move(thenFuture), thread, [promise](const QException & e) {
524         promise->setException(e);
525         promise->finish();
526     });
527 }
528
529 template <class T, class U>
530 void thenOrFailed(QFuture<T> && future, std::shared_ptr<QPromise<U>> promise)
531 {
532     thenOrFailed(std::move(future), promise, [promise] { promise->finish(); });
533 }
534
535 template <class T, class U>
536 void thenOrFailed(
537     QFuture<T> && future, QThread * thread,
538     std::shared_ptr<QPromise<U>> promise)
539 {
540     thenOrFailed(
541         std::move(future), thread, promise, [promise] { promise->finish(); });
542 }
543
544 } // namespace quentier::threading
```

6.72 QtFutureHelpers.h

```
1 /*
2 * Copyright 2021-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <QFuture>
22
23 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
24 #include "Qt5Promise.h"
25 #endif
26
27 #include <type_traits>
28
29 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
30 // Backports of some helpers for QFuture continuations from Qt6 to Qt5
31 namespace QtFuture {
32
33 // Inherit option from Qt6 is not supported in Qt5
34 enum class Launch
35 {
```

```
36     Sync,
37     Async,
38 };
39
40 } // namespace QtFuture
41
42 namespace QtPrivate {
43
44 template <typename...>
45 struct ArgsType;
46
47 template <typename Arg, typename...  Args>
48 struct ArgsType<Arg, Args...>
49 {
50     using First = Arg;
51     using PromiseType = void;
52     using IsPromise = std::false_type;
53     static const bool HasExtraArgs = (sizeof...(Args) > 0);
54
55     template <class Class, class Callable>
56     static const bool CanInvokeWithArgs =
57         std::is_invocable_v<Callable, Class, Arg, Args...>;
58 };
59
60 template <typename Arg, typename...  Args>
61 struct ArgsType<QPromise<Arg> &, Args...>
62 {
63     using First = QPromise<Arg> &;
64     using PromiseType = Arg;
65     using IsPromise = std::true_type;
66     static const bool HasExtraArgs = (sizeof...(Args) > 0);
67
68     template <class Class, class Callable>
69     static const bool CanInvokeWithArgs =
70         std::is_invocable_v<Callable, Class, QPromise<Arg> &, Args...>;
71 };
72
73 template <>
74 struct ArgsType<>
75 {
76     using First = void;
77     using PromiseType = void;
78     using IsPromise = std::false_type;
79     static const bool HasExtraArgs = false;
80     using AllArgs = void;
81
82     template <class Class, class Callable>
83     static const bool CanInvokeWithArgs = std::is_invocable_v<Callable, Class>;
84 };
85
86 template <typename F>
87 struct ArgResolver :  ArgResolver<decltype(&std::decay_t<F>::operator())>
88 {};
89
90 template <typename F>
91 struct ArgResolver<std::reference_wrapper<F» :
92     ArgResolver<decltype(&std::decay_t<F>::operator())>
93 {};
94
95 template <typename R, typename...  Args>
96 struct ArgResolver<R(Args...)> :  public ArgsType<Args...>
97 {};
98
99 template <typename R, typename...  Args>
100 struct ArgResolver<R (*)(Args...)> :  public ArgsType<Args...>
101 {};
102
103 template <typename R, typename...  Args>
104 struct ArgResolver<R (*&)(Args...)> :  public ArgsType<Args...>
105 {};
106
107 template <typename R, typename...  Args>
108 struct ArgResolver<R (*const)(Args...)> :  public ArgsType<Args...>
109 {};
110
111 template <typename R, typename...  Args>
112 struct ArgResolver<R (&)(Args...)> :  public ArgsType<Args...>
113 {};
114
115 template <typename Class, typename R, typename...  Args>
116 struct ArgResolver<R (Class::*)(Args...)> :  public ArgsType<Args...>
117 {};
118
119 template <typename Class, typename R, typename...  Args>
120 struct ArgResolver<R (Class::*)(Args...)  noexcept> :  public ArgsType<Args...>
121 {};
122
```

```
123 template <typename Class, typename R, typename...  Args>
124 struct ArgResolver<R (Class::*)(Args...)  const> :  public ArgsType<Args...>
125 {};
126
127 template <typename Class, typename R, typename...  Args>
128 struct ArgResolver<R (Class::*)(Args...)  const noexcept> :
129     public ArgsType<Args...>
130 {};
131
132 template <typename Class, typename R, typename...  Args>
133 struct ArgResolver<R (Class::*const)(Args...)  const> :  public ArgsType<Args...>
134 {};
135
136 template <typename Class, typename R, typename...  Args>
137 struct ArgResolver<R (Class::*const)(Args...)  const noexcept> :
138     public ArgsType<Args...>
139 {};
140
141 } // namespace QtPrivate
142 #endif // QT_VERSION
143
144 namespace quentier::threading::detail {
145
146 template <typename F, typename Arg, typename Enable = void>
147 struct ResultTypeHelper
148 {};
149
150 // The callable takes an argument of type Arg
151 template <typename F, typename Arg>
152 struct ResultTypeHelper<
153     F, Arg,
154     typename std::enable_if_t<
155         !std::is_invocable_v<std::decay_t<F>, QFuture<Arg>>>
156 {
157     using ResultType = std::invoke_result_t<std::decay_t<F>, std::decay_t<Arg>;
158 };
159
160 // The callable takes an argument of type QFuture<Arg>
161 template <class F, class Arg>
162 struct ResultTypeHelper<
163     F, Arg,
164     typename std::enable_if_t<
165         std::is_invocable_v<std::decay_t<F>, QFuture<Arg>>>
166 {
167     using ResultType = std::invoke_result_t<std::decay_t<F>, QFuture<Arg>;
168 };
169
170 // The callable takes an argument of type QFuture<void>
171 template <class F>
172 struct ResultTypeHelper<
173     F, void,
174     typename std::enable_if_t<
175         std::is_invocable_v<std::decay_t<F>, QFuture<void>>>
176 {
177     using ResultType = std::invoke_result_t<std::decay_t<F>, QFuture<void>;
178 };
179
180 // The callable doesn't take argument
181 template <class F>
182 struct ResultTypeHelper<
183     F, void,
184     typename std::enable_if_t<
185         !std::is_invocable_v<std::decay_t<F>, QFuture<void>>>
186 {
187     using ResultType = std::invoke_result_t<std::decay_t<F>;
188 };
189
190 } // namespace quentier::threading::detail
```

6.73 Runnable.h

```
1 /*
2 * Copyright 2021 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
```

```
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <functional>
24
25 class QRunnable;
26
27 namespace quentier::threading {
28
33 [[nodiscard]] auto QUENTIER_EXPORT
34     createFunctionRunnable(std::function<void()> function) -> QRunnable *;
35
36 } // namespace quentier::threading
```

6.74 TrackedTask.h

```
1 /*
2 * Copyright 2022-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <functional>
22 #include <type_traits>
23 #include <utility>
24
25 namespace quentier::threading {
26
27 namespace detail {
28
29 template <typename LockableObject, typename Function, typename...  Arguments>
30 constexpr std::enable_if_t<std::is_invocable_v<Function, Arguments...>> invoke(
31     LockableObject & lockableObject, Function & function,
32     Arguments &&...  arguments)
33 {
34     const auto lockedObject = lockableObject.lock();
35     if (lockedObject) {
36         std::invoke(function, std::forward<Arguments>(arguments)...);
37     }
38 }
39
40 template <typename LockableObject, typename Function, typename...  Arguments>
41 constexpr std::enable_if_t<
42     !std::is_invocable_v<Function, Arguments...> &&
43     std::is_member_function_pointer_v<Function>>
44     invoke(
45         LockableObject & lockableObject, Function & function,
46         Arguments &&...  arguments)
47 {
48     const auto lockedObject = lockableObject.lock();
49     if (lockedObject) {
50         std::invoke(
51             function, *lockedObject, std::forward<Arguments>(arguments)...);
52     }
53 }
54
55 } // namespace detail
56
73 template <typename LockableObject, typename Function>
74 class TrackedTask
75 {
76 public:
```

```
77      template <typename SomeLockableObject, typename SomeFunction>
78      constexpr TrackedTask(
79          SomeLockableObject && someLockableObject, SomeFunction && function) :
80          m_lockableObject{std::forward<SomeLockableObject>(someLockableObject)},
81          m_function{std::forward<SomeFunction>(function)}
82      {}
83
84      template <
85          typename...  Arguments,
86          typename = std::enable_if_t<
87              std::is_invocable_v<Function, Arguments...> ||
88              std::is_member_function_pointer_v<Function»>
89      constexpr void operator()(Arguments &&...  arguments)
90      {
91          detail::invoke(
92              m_lockableObject, m_function,
93              std::forward<Arguments>(arguments)...);
94      }
95
96      template <
97          typename...  Arguments,
98          typename = std::enable_if_t<
99              std::is_invocable_v<Function, Arguments...> ||
100              std::is_member_function_pointer_v<Function»>
101     constexpr void operator()(Arguments &&...  arguments)const
102 {
103         detail::invoke(
104             m_lockableObject, m_function,
105             std::forward<Arguments>(arguments)...);
106     }
107
108 private:
109     LockableObject m_lockableObject;
110     Function m_function;
111 };
112
113 template <typename LockableObject, typename Function>
114 TrackedTask(LockableObject, Function) -> TrackedTask<LockableObject, Function>;
115
116 } // namespace quentier::threading
```

6.75 Account.h

```
1  /*
2  * Copyright 2016-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Printable.h>
22
23 #include <qevercloud/QEverCloud.h>
24
25 #include <QSharedDataPointer>
26 #include <QString>
27
28 namespace quentier {
29
30 class AccountData;
31
37 class QUENTIER_EXPORT Account :  public Printable
38 {
39 public:
40     enum class Type
41     {
42         Local,
43         Evernote
44     };
45
```

```
46      friend QUENTIER_EXPORT QTextStream & operator«(
47          QTextStream & strm, Type type);
48
49      friend QUENTIER_EXPORT QDebug & operator«(QDebug & dbg, Type type);
50
51      enum class EvernoteAccountType
52      {
53          Free,
54          Plus,
55          Premium,
56          Business
57      };
58
59      friend QUENTIER_EXPORT QTextStream & operator«(
60          QTextStream & strm, EvernoteAccountType type);
61
62      friend QUENTIER_EXPORT QDebug & operator«(
63          QDebug & dbg, EvernoteAccountType type);
64
65  public:
66      explicit Account();
67
68      explicit Account(
69          QString name, Type type, qevercloud::UserID userId = -1,
70          EvernoteAccountType evernoteAccountType = EvernoteAccountType::Free,
71          QString evernoteHost = {}, QString shardId = {});
72
73      Account(const Account & other);
74      Account(Account && other) noexcept;
75
76      Account & operator=(const Account & other);
77      Account & operator=(Account && other) noexcept;
78
79      ~Account() noexcept override;
80
81      [[nodiscard]] bool operator==(const Account & other) const noexcept;
82      [[nodiscard]] bool operator!=(const Account & other) const noexcept;
83
89      [[nodiscard]] bool isEmpty() const;
90
94      [[nodiscard]] QString name() const;
95
99      void setName(QString name);
100
106     [[nodiscard]] QString displayName() const;
107
111     void setDisplayName(QString displayName);
112
116     [[nodiscard]] Type type() const;
117
123     [[nodiscard]] qevercloud::UserID id() const;
124
129     [[nodiscard]] EvernoteAccountType evernoteAccountType() const;
130
135     [[nodiscard]] QString evernoteHost() const;
136
142     [[nodiscard]] QString shardId() const;
143
144     void setEvernoteAccountType(EvernoteAccountType evernoteAccountType);
145     void setEvernoteHost(QString evernoteHost);
146     void setShardId(QString shardId);
147
148     [[nodiscard]] qint32 mailLimitDaily() const;
149     [[nodiscard]] qint64 noteSizeMax() const;
150     [[nodiscard]] qint64 resourceSizeMax() const;
151     [[nodiscard]] qint32 linkedNotebookMax() const;
152     [[nodiscard]] qint32 noteCountMax() const;
153     [[nodiscard]] qint32 notebookCountMax() const;
154     [[nodiscard]] qint32 tagCountMax() const;
155     [[nodiscard]] qint32 noteTagCountMax() const;
156     [[nodiscard]] qint32 savedSearchCountMax() const;
157     [[nodiscard]] qint32 noteResourceCountMax() const;
158
159     void setEvernoteAccountLimits(const qevercloud::AccountLimits & limits);
160
161     QTextStream & print(QTextStream & strm) const override;
162
163 private:
164     QSharedDataPointer<AccountData> d;
165 };
166
167 } // namespace quentier
```

6.76 ErrorString.h

```
1 /*
2 * Copyright 2017-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Printable.h>
22
23 #include <QSharedDataPointer>
24
25 namespace quentier {
26
27 class ErrorStringData;
28
42 class QUENTIER_EXPORT ErrorString :  public Printable
43 {
44 public:
45     explicit ErrorString(const char * error = nullptr);
46     explicit ErrorString(const QString & error);
47
48     ErrorString(const ErrorString & other);
49     ErrorString(ErrorString && other) noexcept;
50
51     ErrorString & operator=(const ErrorString & other);
52     ErrorString & operator=(ErrorString && other) noexcept;
53
54     ~ErrorString() override;
55
56     [[nodiscard]] const QString & base() const noexcept;
57     [[nodiscard]] QString & base();
58
59     [[nodiscard]] const QStringList & additionalBases() const noexcept;
60     [[nodiscard]] QStringList & additionalBases();
61
62     [[nodiscard]] const QString & details() const noexcept;
63     [[nodiscard]] QString & details();
64
65     void setBase(QString error);
66     void setBase(const char * error);
67
68     void appendBase(const QString & error);
69     void appendBase(const QStringList & errors);
70     void appendBase(const char * error);
71
72     void setDetails(const QString & error);
73     void setDetails(const char * error);
74
75     [[nodiscard]] bool isEmpty() const;
76     void clear();
77
78     [[nodiscard]] QString localizedString() const;
79     [[nodiscard]] QString nonLocalizedString() const;
80
81     QTextStream & print(QTextStream & strm) const override;
82
83 private:
84     QSharedDataPointer<ErrorStringData> d;
85 };
86
87 [[nodiscard]] QUENTIER_EXPORT bool operator==(
88     const ErrorString & lhs, const ErrorString & rhs) noexcept;
89
90 [[nodiscard]] QUENTIER_EXPORT bool operator!=(
91     const ErrorString & lhs, const ErrorString & rhs) noexcept;
92
93 } // namespace quentier
```

6.77 NoteUtils.h

```
1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <qevercloud/types/Fwd.h>
25
26 #include <QStringList>
27
28 #include <utility>
29
30 namespace quentier {
31
32 [[nodiscard]] QUENTIER_EXPORT bool isInkNote(const qevercloud::Note & note);
33
34 [[nodiscard]] QUENTIER_EXPORT bool noteContentContainsCheckedToDo(
35     const QString & noteContent);
36
37 [[nodiscard]] QUENTIER_EXPORT bool noteContentContainsUncheckedToDo(
38     const QString & noteContent);
39
40 [[nodiscard]] QUENTIER_EXPORT bool noteContentContainsToDo(
41     const QString & noteContent);
42
43 [[nodiscard]] QUENTIER_EXPORT bool noteContentContainsEncryptedFragments(
44     const QString & noteContent);
45
46 [[nodiscard]] QUENTIER_EXPORT QString noteContentToPlainText(
47     const QString & noteContent, ErrorString * errorDescription = nullptr);
48
49 [[nodiscard]] QUENTIER_EXPORT QStringList noteContentToListOfWords(
50     const QString & noteContent, ErrorString * errorDescription = nullptr);
51
52 [[nodiscard]] QUENTIER_EXPORT std::pair<QString, QStringList>
53     noteContentToPlainTextAndListOfWords(
54         const QString & noteContent, ErrorString * errorDescription = nullptr);
55
56 } // namespace quentier
```

6.78 RegisterMetatypes.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
```

```
23 namespace quentier {
24
25 QUENTIER_EXPORT void registerMetatypes();
26
27 } // namespace quentier
```

6.79 ResourceRecognitionIndexItem.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22 #include <quentier/utility/Printable.h>
23
24 #include <QByteArray>
25 #include <QList>
26 #include <QSharedDataPointer>
27
28 #include <memory>
29
30 namespace quentier {
31
32 class ResourceRecognitionIndexItemData;
33
34 class QUENTIER_EXPORT ResourceRecognitionIndexItem :  public Printable
35 {
36 public:
37     explicit ResourceRecognitionIndexItem();
38
39     ResourceRecognitionIndexItem(const ResourceRecognitionIndexItem & other);
40
41     ResourceRecognitionIndexItem(
42         ResourceRecognitionIndexItem && other) noexcept;
43
44     ResourceRecognitionIndexItem & operator=(
45         const ResourceRecognitionIndexItem & other);
46
47     ResourceRecognitionIndexItem & operator=(
48         ResourceRecognitionIndexItem && other) noexcept;
49
50     ~ResourceRecognitionIndexItem() override;
51
52     [[nodiscard]] bool isValid() const;
53
54     [[nodiscard]] int x() const;
55     void setX(int x);
56
57     [[nodiscard]] int y() const;
58     void setY(int y);
59
60     [[nodiscard]] int h() const;
61     void setH(int h);
62
63     [[nodiscard]] int w() const;
64     void setW(int w);
65
66     [[nodiscard]] int offset() const;
67     void setOffset(int offset);
68
69     [[nodiscard]] int duration() const;
70     void setDuration(int duration);
71
72     [[nodiscard]] QList<int> strokes() const;
73     void setStrokes(QList<int> strokes);
74
75     struct QUENTIER_EXPORT ITextItem
```

```
76      {
77          virtual ~ITextItem();
78
79          [[nodiscard]] virtual QString text() const = 0;
80          [[nodiscard]] virtual int weight() const = 0;
81      };
82
83      using ITextItemPtr = std::shared_ptr<ITextItem>;
84
85      [[nodiscard]] QList<ITextItemPtr> textItems() const;
86      void setTextItems(QList<ITextItemPtr> textItems);
87
88      struct QUENTIER_EXPORT IObjectItem
89      {
90          virtual ~IObjectItem();
91
92          [[nodiscard]] virtual QString objectType() const = 0;
93          [[nodiscard]] virtual int weight() const = 0;
94      };
95
96      using IObjectItemPtr = std::shared_ptr<IObjectItem>;
97
98      [[nodiscard]] QList<IObjectItemPtr> objectItems() const;
99      void setObjectItems(QList<IObjectItemPtr> objectItems);
100
101      struct QUENTIER_EXPORT IShapeItem
102      {
103          virtual ~IShapeItem();
104
105          [[nodiscard]] virtual QString shape() const = 0;
106          [[nodiscard]] virtual int weight() const = 0;
107      };
108
109      using IShapeItemPtr = std::shared_ptr<IShapeItem>;
110
111      [[nodiscard]] QList<IShapeItemPtr> shapeItems() const;
112      void setShapeItems(QList<IShapeItemPtr> shapeItems);
113
114      struct QUENTIER_EXPORT IBarcodeItem
115      {
116          virtual ~IBarcodeItem();
117
118          [[nodiscard]] virtual QString barcode() const = 0;
119          [[nodiscard]] virtual int weight() const = 0;
120      };
121
122      using IBarcodeItemPtr = std::shared_ptr<IBarcodeItem>;
123
124      [[nodiscard]] QList<IBarcodeItemPtr> barcodeItems() const;
125      void setBarcodeItems(QList<IBarcodeItemPtr> barcodeItems);
126
127      QTextStream & print(QTextStream & strm) const override;
128
129  private:
130      QSharedDataPointer<ResourceRecognitionIndexItemData> d;
131  };
132
133  } // namespace quentier
```

6.80 ResourceRecognitionIndices.h

```
1  /*
2  * Copyright 2016-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/ResourceRecognitionIndexItem.h>
22
```

```
23 #include <QByteArray>
24 #include <QSharedDataPointer>
25 #include <QVector>
26
27 namespace quentier {
28
29 class ResourceRecognitionIndicesData;
30
31 class QUENTIER_EXPORT ResourceRecognitionIndices :  public Printable
32 {
33 public:
34     explicit ResourceRecognitionIndices();
35
36     explicit ResourceRecognitionIndices(
37         const QByteArray & rawRecognitionIndicesData);
38
39     ResourceRecognitionIndices(const ResourceRecognitionIndices & other);
40     ResourceRecognitionIndices(ResourceRecognitionIndices && other) noexcept;
41
42     ResourceRecognitionIndices & operator=(
43         const ResourceRecognitionIndices & other);
44
45     ResourceRecognitionIndices & operator=(
46         ResourceRecognitionIndices && other) noexcept;
47
48     ~ResourceRecognitionIndices() override;
49
50     [[nodiscard]] bool isNull() const;
51     [[nodiscard]] bool isValid() const;
52
53     [[nodiscard]] QString objectId() const;
54     [[nodiscard]] QString objectType() const;
55     [[nodiscard]] QString recoType() const;
56     [[nodiscard]] QString engineVersion() const;
57     [[nodiscard]] QString docType() const;
58     [[nodiscard]] QString lang() const;
59
60     [[nodiscard]] int objectHeight() const;
61     [[nodiscard]] int objectWidth() const;
62
63     [[nodiscard]] QVector<ResourceRecognitionIndexItem> items() const;
64
65     bool setData(const QByteArray & rawRecognitionIndicesData);
66
67     QTextStream & print(QTextStream & strm) const override;
68
69 private:
70     QSharedDataPointer<ResourceRecognitionIndicesData> d;
71 };
72
73 } // namespace quentier
```

6.81 ResourceUtils.h

```
1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <qevercloud/types/Fwd.h>
24
25 #include <QString>
26
27 namespace quentier {
28
29 [[nodiscard]] QUENTIER_EXPORT QString
```

```
30     resourceDisplayName(const qevercloud::Resource & resource);
31
32 [[nodiscard]] QUENTIER_EXPORT QString
33     preferredFileSuffix(const qevercloud::Resource & resource);
34
35 } // namespace quentier
```

6.82 Result.h

```
1  /*
2  * Copyright 2023-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/exception/RuntimeError.h>
22 #include <quentier/types/ErrorString.h>
23
24 #include <type_traits>
25 #include <variant>
26
27 namespace quentier {
28
33 template <
34     class T, class Error,
35     typename = typename std::enable_if_t<
36         !std::is_same_v<std::decay_t<T>, std::decay_t<Error>>>
37 class Result
38 {
39     using ValueType = std::conditional_t<
40         std::is_same_v<std::decay_t<T>, void>, std::monostate, T>;
41
42 public:
43     template <
44         typename T1 = T,
45         typename std::enable_if_t<!std::is_void_v<std::decay_t<T1>> * =
46             nullptr>
47     explicit Result(T1 t) :  m_valueOrError{std::move(t)}
48     {}
49
50     template <
51         typename T1 = T,
52         typename std::enable_if_t<std::is_void_v<std::decay_t<T1>> * = nullptr>
53     explicit Result() :  m_valueOrError{std::monostate{}}
54     {}
55
56     explicit Result(Error error) :  m_valueOrError{std::move(error)} {}
57
58     Result(const Result<T, Error> & other) :
59         m_valueOrError{other.m_valueOrError}
60     {}
61
62     Result(Result<T, Error> && other) :
63         m_valueOrError{std::move(other.m_valueOrError)}
64     {}
65
66     Result & operator=(const Result<T, Error> & other)
67     {
68         if (this != &other) {
69             m_valueOrError = other.m_valueOrError;
70         }
71
72         return *this;
73     }
74
75     Result & operator=(Result<T, Error> && other)
76     {
77         if (this != &other) {
78             m_valueOrError = std::move(other.m_valueOrError);
```

```
79          }
80
81          return *this;
82      }
83
87      [[nodiscard]] bool isValid() const noexcept
88      {
89          return std::holds_alternative<ValueType>(m_valueOrError);
90      }
91
92      operator bool() const noexcept
93      {
94          return isValid();
95      }
96
97      template <
98          typename T1 = T,
99          typename std::enable_if_t<!std::is_void_v<std::decay_t<T1>> * =
100             nullptr>
101     [[nodiscard]] T1 & get()
102     {
103         // NOTE: std::get also performs the check of what is stored inside the
104         // variant but it throws std::bad_variant_access which doesn't implement
105         // QException so this exception is not representable inside QFuture
106         // in Qt5.  Due to this for Qt5 also performing another check and using
107         // another exception type
108 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
109         if (Q_UNLIKELY(!isValid())) {
110             throw RuntimeError{
111                 ErrorString{"Detected attempt to get value from empty Result"}};
112         }
113 #endif
114
115         return std::get<T>(m_valueOrError);
116     }
117
118     template <
119         typename T1 = T,
120         typename std::enable_if_t<!std::is_void_v<std::decay_t<T1>> * =
121             nullptr>
122     [[nodiscard]] const T1 & get()const
123 {
124         // NOTE: std::get also performs the check of what is stored inside the
125         // variant but it throws std::bad_variant_access which doesn't implement
126         // QException so this exception is not representable inside QFuture
127         // in Qt5.  Due to this for Qt5 also performing another check and using
128         // another exception type
129 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
130         if (Q_UNLIKELY(!isValid())) {
131             throw RuntimeError{
132                 ErrorString{"Detected attempt to get value from empty Result"}};
133         }
134 #endif
135
136         return std::get<T>(m_valueOrError);
137     }
138
139     template <
140         typename T1 = T,
141         typename std::enable_if_t<!std::is_void_v<std::decay_t<T1>> * =
142             nullptr>
143     [[nodiscard]] T1 & operator*()
144     {
145         return get();
146     }
147
148     template <
149         typename T1 = T,
150         typename std::enable_if_t<!std::is_void_v<std::decay_t<T1>> * =
151             nullptr>
152     [[nodiscard]] const T1 & operator*()const
153 {
154         return get();
155     }
156
157     [[nodiscard]] const Error & error()const
158 {
159         // NOTE: std::get also performs the check of what is stored inside the
160         // variant but it throws std::bad_variant_access which doesn't implement
161         // QException so this exception is not representable inside QFuture
162         // in Qt5.  Due to this for Qt5 also performing another check and using
163         // another exception type
164 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
165         if (Q_UNLIKELY(isValid())) {
166             throw RuntimeError{ErrorString{
167                 "Detected attempt to get error from non-empty Result"}};
168         }
```

```
169 #endif
170
171         return std::get<Error>(m_valueOrError);
172     }
173
174     [[nodiscard]] Error & error()
175     {
176         // NOTE: std::get also performs the check of what is stored inside the
177         // variant but it throws std::bad_variant_access which doesn't implement
178         // QException so this exception is not representable inside QFuture
179         // in Qt5.  Due to this for Qt5 also performing another check and using
180         // another exception type
181 #if QT_VERSION < QT_VERSION_CHECK(6, 0, 0)
182         if (Q_UNLIKELY(isValid())) {
183             throw RuntimeError{ErrorString{
184                 "Detected attempt to get error from non-empty Result"}};
185         }
186 #endif
187
188         return std::get<Error>(m_valueOrError);
189     }
190
191 private:
192     std::variant<ValueType, Error> m_valueOrError;
193 };
194
195 } // namespace quentier
```

6.83 Validation.h

```
1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/Fwd.h>
22 #include <quentier/utility/Linkage.h>
23
24 class QString;
25
26 namespace quentier {
27
38 [[nodiscard]] QUENTIER_EXPORT bool validateNoteTitle(
39     const QString & noteTitle,
40     ErrorString * errorDescription = nullptr) noexcept;
41
52 [[nodiscard]] QUENTIER_EXPORT bool validateNotebookName(
53     const QString & notebookName,
54     ErrorString * errorDescription = nullptr) noexcept;
55
67 [[nodiscard]] QUENTIER_EXPORT bool validateSavedSearchName(
68     const QString & savedSearchName,
69     ErrorString * errorDescription = nullptr) noexcept;
70
81 [[nodiscard]] QUENTIER_EXPORT bool validateTagName(
82     const QString & tagName, ErrorString * errorDescription = nullptr) noexcept;
83
84 } // namespace quentier
```

6.84 ApplicationSettings.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
```

```
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/Account.h>
22
23 #include <QSettings>
24
25 #include <string_view>
26
27 namespace quentier {
28
34 class QUENTIER_EXPORT ApplicationSettings :  public QSettings, public Printable
35 {
36     Q_OBJECT
37 public:
47     explicit ApplicationSettings(const QString & settingsName = {});
48
60     explicit ApplicationSettings(
61         const Account & account, const QString & settingsName = {});
62
79     ApplicationSettings(
80         const Account & account, const char * settingsName,
81         int settingsNameSize = -1);
82
96     ApplicationSettings(const Account & account, std::string_view settingsName);
97
101     ~ApplicationSettings() override;
102
103 public:
108     struct ArrayCloser
109     {
110         ArrayCloser(ApplicationSettings & settings) :  m_settings(settings) {}
111
112         ~ArrayCloser()
113         {
114             m_settings.endArray();
115             m_settings.sync();
116         }
117
118         ApplicationSettings & m_settings;
119     };
120
125     struct GroupCloser
126     {
127         GroupCloser(ApplicationSettings & settings) :  m_settings(settings) {}
128
129         ~GroupCloser()
130         {
131             m_settings.endGroup();
132             m_settings.sync();
133         }
134
135         ApplicationSettings & m_settings;
136     };
137
138 public:
145     void beginGroup(const QString & prefix);
146
157     void beginGroup(const char * prefix, int size = -1);
158
168     void beginGroup(std::string_view prefix);
169
177     [[nodiscard]] int beginReadArray(const QString & prefix);
178
189     [[nodiscard]] int beginReadArray(const char * prefix, int size = -1);
190
198     [[nodiscard]] int beginReadArray(std::string_view prefix);
199
210     void beginWriteArray(const QString & prefix, int arraySize = -1);
211
226     void beginWriteArray(
227         const char * prefix, int arraySize = -1, int prefixSize = -1);
228
```

```
240     void beginWriteArray(std::string_view prefix, int arraySize = -1);
241
249     [[nodiscard]] bool contains(const QString & key) const;
250
262     [[nodiscard]] bool contains(const char * key, int size = -1) const;
263
272     [[nodiscard]] bool contains(std::string_view key) const;
273
280     void remove(const QString & key);
281
292     void remove(const char * key, int size = -1);
293
301     void remove(std::string_view key);
302
310     void setValue(const QString & key, const QVariant & value);
311
323     void setValue(const char * key, const QVariant & value, int keySize = -1);
324
333     void setValue(std::string_view key, const QVariant & value);
334
345     [[nodiscard]] QVariant value(
346         const QString & key, const QVariant & defaultValue = {}) const;
347
362     [[nodiscard]] QVariant value(
363         const char * key, const QVariant & defaultValue = {},
364         int keySize = -1) const;
365
377     [[nodiscard]] QVariant value(
378         std::string_view key, const QVariant & defaultValue = {}) const;
379
380 public:
381     QTextStream & print(QTextStream & strm) const override;
382
383 private:
384     Q_DISABLE_COPY(ApplicationSettings)
385 };
386
387 } // namespace quentier
```

6.85 AnyOfCanceler.h

```
1  /*
2  * Copyright 2022 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/cancelers/Fwd.h>
22 #include <quentier/utility/cancelers/ICanceler.h>
23
24 #include <QList>
25
26 namespace quentier::utility::cancelers {
27
28 class QUENTIER_EXPORT AnyOfCanceler :  public ICanceler
29 {
30 public:
31     explicit AnyOfCanceler(QList<ICancelerPtr> cancelers);
32     AnyOfCanceler(AnyOfCanceler && other) noexcept;
33     AnyOfCanceler & operator=(AnyOfCanceler && other) noexcept;
34     ~AnyOfCanceler() noexcept override;
35
36     [[nodiscard]] bool isCanceled() const noexcept override;
37
38 private:
39     class Impl;
40     std::unique_ptr<Impl> m_impl;
41 };
42
43 } // namespace quentier::utility::cancelers
```

6.86 FutureCanceler.h

```
1 /*
2 * Copyright 2022 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/cancelers/ICanceler.h>
22
23 #include <QFuture>
24
25 namespace quentier::utility::cancelers {
26
30 template <class T>
31 class FutureCanceler :  public ICanceler
32 {
33 public:
34     explicit FutureCanceler(QFuture<T> future) :  m_future{std::move(future)} {}
35
36     [[nodiscard]] bool isCanceled() const noexcept override
37     {
38         return m_future.isCanceled();
39     }
40
41 private:
42     QFuture<T> m_future;
43 };
44
45 } // namespace quentier::utility::cancelers
```

6.87 ICanceler.h

```
1 /*
2 * Copyright 2022 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 namespace quentier::utility::cancelers {
24
29 class QUENTIER_EXPORT ICanceler
30 {
31 public:
32     virtual ~ICanceler() = default;
33
34     [[nodiscard]] virtual bool isCanceled() const = 0;
35 };
36
37 } // namespace quentier::utility::cancelers
```

6.88 ManualCanceler.h

```
1 /*
2 * Copyright 2022 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/cancelers/ICanceler.h>
22
23 #include <atomic>
24 #include <memory>
25
26 namespace quentier::utility::cancelers {
27
32 class QUENTIER_EXPORT ManualCanceler :  public ICanceler
33 {
34 public:
35     ManualCanceler();
36     ManualCanceler(ManualCanceler && other) noexcept;
37     ManualCanceler & operator=(ManualCanceler && other) noexcept;
38     ~ManualCanceler() noexcept override;
39
43     void cancel() noexcept;
44
45     [[nodiscard]] bool isCanceled() const noexcept override;
46
47 private:
48     class Impl;
49     std::unique_ptr<Impl> m_impl;
50 };
51
52 } // namespace quentier::utility::cancelers
```

6.89 Checks.h

```
1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QString>
24
25 namespace quentier {
26
33 [[nodiscard]] bool QUENTIER_EXPORT checkGuid(const QString & guid);
34
44 [[nodiscard]] bool QUENTIER_EXPORT
45     checkUpdateSequenceNumber(qint32 updateSequenceNumber);
46
47 } // namespace quentier
```

6.90 Compat.h

```
1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <QHash>
22 #include <QString>
23 #include <QtGlobal>
24
25 // Compatibility with boost parts which require to take a hash of QString
26
27 inline std::size_t hash_value(const QString & x) noexcept
28 {
29     return qHash(x);
30 }
```

6.91 DateTime.h

```
1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QFlags>
24
25 namespace quentier {
26
27 [[nodiscard]] constexpr int secondsToMilliseconds(int seconds) noexcept
28 {
29     return seconds * 1000;
30 }
31
35 enum class DateTimePrintOption
36 {
41     IncludeNumericTimestamp = 1 << 1,
45     IncludeMilliseconds = 1 << 2,
51     IncludeTimezone = 1 << 3
52 };
53
54 Q_DECLARE_FLAGS(DateTimePrintOptions, DateTimePrintOption)
55 Q_DECLARE_OPERATORS_FOR_FLAGS(DateTimePrintOptions)
56
57
74 [[nodiscard]] QString QUENTIER_EXPORT printableDateTimeFromTimestamp(
75     qint64 timestamp,
76     DateTimePrintOptions options = DateTimePrintOptions(
77         DateTimePrintOption::IncludeNumericTimestamp |
78         DateTimePrintOption::IncludeMilliseconds |
79         DateTimePrintOption::IncludeTimezone),
```

```
80     const char * customFormat = nullptr);
81
82 } // namespace quentier
```

6.92 EncryptionManager.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/ErrorString.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QObject>
25 #include <QString>
26 #include <QUuid>
27
28 namespace quentier {
29
30 class EncryptionManagerPrivate;
31
37 class QUENTIER_EXPORT EncryptionManager :  public QObject
38 {
39     Q_OBJECT
40 public:
41     explicit EncryptionManager(QObject * parent = nullptr);
42     ~EncryptionManager() noexcept override;
43
44     [[nodiscard]] bool decrypt(
45         const QString & encryptedText, const QString & passphrase,
46         const QString & cipher, size_t keyLength, QString & decryptedText,
47         ErrorString & errorDescription);
48
49     [[nodiscard]] bool encrypt(
50         const QString & textToEncrypt, const QString & passphrase,
51         QString & cipher, size_t & keyLength, QString & encryptedText,
52         ErrorString & errorDescription);
53
54 Q_SIGNALS:
55     void decryptedText(
56         QString text, bool success, ErrorString errorDescription,
57         QUuid requestId);
58
59     void encryptedText(
60         QString encryptedText, bool success, ErrorString errorDescription,
61         QUuid requestId);
62
63 public Q_SLOTS:
64     void onDecryptTextRequest(
65         QString encryptedText, QString passphrase, QString cipher,
66         size_t keyLength, QUuid requestId);
67
68     void onEncryptTextRequest(
69         QString textToEncrypt, QString passphrase, QString cipher,
70         size_t keyLength, QUuid requestId);
71
72 private:
73     EncryptionManagerPrivate * const d_ptr;
74     Q_DECLARE_PRIVATE(EncryptionManager)
75 };
76
77 } // namespace quentier
```

6.93 EventLoopWithExitStatus.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/ErrorString.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QEventLoop>
25
26 class QDebug;
27 class QTextStream;
28
29 namespace quentier {
30
31 class QUENTIER_EXPORT EventLoopWithExitStatus :  public QEventLoop
32 {
33     Q_OBJECT
34 public:
35     explicit EventLoopWithExitStatus(QObject * parent = nullptr);
36
37     enum class ExitStatus
38     {
39         Success,
40         Failure,
41         Timeout
42     };
43
44     friend QDebug & operator<<(QDebug & dbg, ExitStatus status);
45     friend QTextStream & operator<<(QTextStream & strm, ExitStatus status);
46
47     [[nodiscard]] ExitStatus exitStatus() const;
48     [[nodiscard]] const ErrorString & errorDescription() const;
49
50 public Q_SLOTS:
51     void exitAsSuccess();
52     void exitAsFailure();
53     void exitAsFailureWithError(QString errorDescription);
54     void exitAsFailureWithErrorString(ErrorString errorDescription);
55     void exitAsTimeout();
56
57 private:
58     ExitStatus m_exitStatus;
59     ErrorString m_errorDescription;
60 };
61
62 } // namespace quentier
```

6.94 FileCopier.h

```
1 /*
2 * Copyright 2018-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
```

```
17 */
18
19 #pragma once
20
21 #include <quentier/types/ErrorString.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QObject>
25 #include <QString>
26
27 class QDebug;
28 class QTextStream;
29
30 namespace quentier {
31
32 class FileCopierPrivate;
33
34 class QUENTIER_EXPORT FileCopier :  public QObject
35 {
36     Q_OBJECT
37 public:
38     explicit FileCopier(QObject * parent = nullptr);
39
40     enum class State
41     {
42         Idle = 0,
43         Copying,
44         Cancelling
45     };
46
47     friend QDebug & operator<<(QDebug & dbg, State state);
48     friend QTextStream & operator<<(QTextStream & strm, State state);
49
50     [[nodiscard]] State state() const;
51
52     [[nodiscard]] QString sourceFilePath() const;
53     [[nodiscard]] QString destinationFilePath() const;
54
55     [[nodiscard]] double currentProgress() const;
56
57 Q_SIGNALS:
58     void progressUpdate(double progress);
59     void finished(QString sourcePath, QString destPath);
60     void cancelled(QString sourcePath, QString destPath);
61     void notifyError(ErrorString error);
62
63 public Q_SLOTS:
64     void copyFile(QString sourcePath, QString destPath);
65     void cancel();
66
67 private:
68     Q_DISABLE_COPY(FileCopier)
69
70 private:
71     FileCopierPrivate * d_ptr;
72     Q_DECLARE_PRIVATE(FileCopier)
73 };
74
75 } // namespace quentier
```

6.95 FileIOProcessorAsync.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/ErrorString.h>
```

```
22 #include <quentier/utility/Linkage.h>
23
24 #include <QByteArray>
25 #include <QIODevice>
26 #include <QObject>
27 #include <QString>
28 #include <QUuid>
29
30 namespace quentier {
31
32 class FileIOProcessorAsyncPrivate;
33
38 class QUENTIER_EXPORT FileIOProcessorAsync :  public QObject
39 {
40     Q_OBJECT
41 public:
42     explicit FileIOProcessorAsync(QObject * parent = nullptr);
43
53     void setIdleTimePeriod(qint32 seconds);
54
55 Q_SIGNALS:
62     void readyForIO();
63
74     void writeFileRequestProcessed(
75         bool success, ErrorString errorDescription, QUuid requestId);
76
88     void readFileRequestProcessed(
89         bool success, ErrorString errorDescription, QByteArray data,
90         QUuid requestId);
91
92 public Q_SLOTS:
104     void onWriteFileRequest(
105         QString absoluteFilePath, QByteArray data, QUuid requestId,
106         bool append);
107
115     void onReadFileRequest(QString absoluteFilePath, QUuid requestId);
116
117 private:
118     FileIOProcessorAsyncPrivate * const d_ptr;
119     Q_DECLARE_PRIVATE(FileIOProcessorAsync)
120 };
121
122 } // namespace quentier
```

6.96 FileSystem.h

```
1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QString>
24
25 namespace quentier {
26
27 class ErrorString;
28
43 [[nodiscard]] QString QUENTIER_EXPORT relativePathFromAbsolutePath(
44     const QString & absolutePath, const QString & relativePathRootFolderPath);
45
54 [[nodiscard]] bool QUENTIER_EXPORT removeFile(const QString & filePath);
55
65 [[nodiscard]] bool QUENTIER_EXPORT removeDir(const QString & dirPath);
66
78 [[nodiscard]] QByteArray QUENTIER_EXPORT
79     readFileContents(const QString & filePath, ErrorString & errorDescription);
```

```
80
95 [[nodiscard]] bool QUENTIER_EXPORT renameFile(
96     const QString & from, const QString & to, ErrorString & errorDescription);
97
98 } // namespace quentier
```

6.97 FileSystemWatcher.h

```
1  /*
2  * Copyright 2016-2024 Dmitry Ivanov
3  *
4  * This file is part of libquentier
5  *
6  * libquentier is free software; you can redistribute it and/or modify
7  * it under the terms of the GNU Lesser General Public License as published by
8  * the Free Software Foundation, version 3 of the License.
9  *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QObject>
24 #include <QStringList>
25
26 #define FILE_SYSTEM_WATCHER_DEFAULT_REMOVAL_TIMEOUT_MSEC (500)
27
28 namespace quentier {
29
30 class FileSystemWatcherPrivate;
31
32 class QUENTIER_EXPORT FileSystemWatcher :  public QObject
33 {
34     Q_OBJECT
35 public:
36     explicit FileSystemWatcher(
37         int removalTimeoutMSec =
38             FILE_SYSTEM_WATCHER_DEFAULT_REMOVAL_TIMEOUT_MSEC,
39         QObject * parent = nullptr);
40
41     explicit FileSystemWatcher(
42         const QStringList & paths,
43         int removalTimeoutMSec =
44             FILE_SYSTEM_WATCHER_DEFAULT_REMOVAL_TIMEOUT_MSEC,
45         QObject * parent = nullptr);
46
47     ~FileSystemWatcher() override;
48
49     void addPath(const QString & path);
50     void addPaths(const QStringList & paths);
51
52     [[nodiscard]] QStringList directories() const;
53     [[nodiscard]] QStringList files() const;
54
55     void removePath(const QString & path);
56     void removePaths(const QStringList & paths);
57
58 Q_SIGNALS:
59     void directoryChanged(const QString & path);
60     void directoryRemoved(const QString & path);
61
62     void fileChanged(const QString & path);
63     void fileRemoved(const QString & path);
64
65 private:
66     Q_DISABLE_COPY(FileSystemWatcher)
67
68 private:
69     FileSystemWatcherPrivate * d_ptr;
70     Q_DECLARE_PRIVATE(FileSystemWatcher)
71 };
72
73 } // namespace quentier
```

6.98 IKeychainService.h

```
1 /*
2 * Copyright 2018-2022 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/exception/IQuentierException.h>
22 #include <quentier/types/ErrorString.h>
23 #include <quentier/utility/Fwd.h>
24 #include <quentier/utility/Linkage.h>
25
26 #include <QFuture>
27
28 class QDebug;
29
30 namespace quentier {
31
36 class QUENTIER_EXPORT IKeychainService
37 {
38 public:
39     virtual ~IKeychainService() noexcept;
40
44     enum class ErrorCode
45     {
49         NoError,
53         EntryNotFound,
57         CouldNotDeleteEntry,
61         AccessDeniedByUser,
65         AccessDenied,
69         NoBackendAvailable,
73         NotImplemented,
77         OtherError
78     };
79
80     friend QUENTIER_EXPORT QTextStream & operator<<(
81         QTextStream & strm, ErrorCode errorCode);
82
83     friend QUENTIER_EXPORT QDebug & operator<<(
84         QDebug & dbg, ErrorCode errorCode);
85
90     class QUENTIER_EXPORT Exception :  public IQuentierException
91     {
92     public:
93         explicit Exception(ErrorCode errorCode) noexcept;
94
95         explicit Exception(
96             ErrorCode errorCode, ErrorString errorDescription) noexcept;
97
98         [[nodiscard]] ErrorCode errorCode() const noexcept;
99         [[nodiscard]] QString exceptionDisplayName() const override;
100
101         void raise() const override;
102         [[nodiscard]] Exception * clone() const override;
103
104     private:
105         const ErrorCode m_errorCode;
106     };
107
108 public:
122     [[nodiscard]] virtual QFuture<void> writePassword(
123         QString service, QString key, QString password) = 0;
124
138     [[nodiscard]] virtual QFuture<QString> readPassword(
139         QString service, QString key) const = 0;
140
153     [[nodiscard]] virtual QFuture<void> deletePassword(
154         QString service, QString key) = 0;
155 };
156
157 [[nodiscard]] QUENTIER_EXPORT IKeychainServicePtr newQtKeychainService();
158
```

```
159 [[nodiscard]] QUENTIER_EXPORT IKeychainServicePtr
160     newObfuscatingKeychainService();
161
162 [[nodiscard]] QUENTIER_EXPORT IKeychainServicePtr newCompositeKeychainService(
163     QString name, IKeychainServicePtr primaryKeychain,
164     IKeychainServicePtr secondaryKeychain);
165
166 [[nodiscard]] QUENTIER_EXPORT IKeychainServicePtr newMigratingKeychainService(
167     IKeychainServicePtr sourceKeychain, IKeychainServicePtr sinkKeychain);
168
169 } // namespace quentier
```

6.99 Initialize.h

```
1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 namespace quentier {
24
29 void QUENTIER_EXPORT initializeLibquentier();
30
31 } // namespace quentier
```

6.100 LRUCache.hpp

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <QHash>
22
23 #include <cstddef>
24 #include <list>
25
26 namespace quentier {
27
28 template <
29     class Key, class Value,
30     class Allocator = std::allocator<std::pair<Key, Value>>
31 class LRUCache
32 {
33 public:
34     LRUCache(const size_t maxSize = 100) :  m_maxSize(maxSize) {}
35
```

```
36      using key_type = Key;
37      using mapped_type = Value;
38      using allocator_type = Allocator;
39      using value_type = std::pair<key_type, mapped_type>;
40      using container_type = std::list<value_type, allocator_type>;
41      using size_type = typename container_type::size_type;
42      using difference_type = typename container_type::difference_type;
43      using iterator = typename container_type::iterator;
44      using const_iterator = typename container_type::const_iterator;
45      using reverse_iterator = std::reverse_iterator<iterator>;
46      using const_reverse_iterator = std::reverse_iterator<const_iterator>;
47
48      using reference = value_type &;
49      using const_reference = const value_type &;
50      using pointer = typename std::allocator_traits<allocator_type>::pointer;
51
52      using const_pointer =
53          typename std::allocator_traits<allocator_type>::const_pointer;
54
55      [[nodiscard]] iterator begin() noexcept
56      {
57          return m_container.begin();
58      }
59
60      [[nodiscard]] const_iterator begin() const noexcept
61      {
62          return m_container.begin();
63      }
64
65      [[nodiscard]] reverse_iterator rbegin() noexcept
66      {
67          return m_container.rbegin();
68      }
69
70      [[nodiscard]] const_reverse_iterator rbegin() const noexcept
71      {
72          return m_container.rbegin();
73      }
74
75      [[nodiscard]] iterator end() noexcept
76      {
77          return m_container.end();
78      }
79
80      [[nodiscard]] const_iterator end() const noexcept
81      {
82          return m_container.end();
83      }
84
85      [[nodiscard]] reverse_iterator rend() noexcept
86      {
87          return m_container.rend();
88      }
89
90      [[nodiscard]] const_reverse_iterator rend() const noexcept
91      {
92          return m_container.rend();
93      }
94
95      [[nodiscard]] bool empty() const noexcept
96      {
97          return m_container.empty();
98      }
99
100     [[nodiscard]] size_t size() const noexcept
101     {
102         return m_currentSize;
103     }
104
105     [[nodiscard]] size_t max_size() const noexcept
106     {
107         return m_maxSize;
108     }
109
110     void clear()
111     {
112         m_container.clear();
113         m_mapper.clear();
114         m_currentSize = 0;
115     }
116
117     void put(const key_type & key, const mapped_type & value)
118     {
119         Q_UNUSED(remove(key))
120
121         m_container.push_front(value_type(key, value));
122         m_mapper[key] = m_container.begin();
```

```
123         ++m_currentSize;
124
125         fixupSize();
126     }
127
128     [[nodiscard]] const mapped_type * get(const key_type & key) const noexcept
129     {
130         auto mapperIt = m_mapper.find(key);
131         if (mapperIt == m_mapper.end()) {
132             return nullptr;
133         }
134
135         auto it = mapperIt.value();
136         if (it == m_container.end()) {
137             return nullptr;
138         }
139
140         m_container.splice(m_container.begin(), m_container, it);
141         mapperIt.value() = m_container.begin();
142         return &(mapperIt.value()->second);
143     }
144
145     [[nodiscard]] bool exists(const key_type & key) const noexcept
146     {
147         const auto mapperIt = m_mapper.find(key);
148         if (mapperIt == m_mapper.end()) {
149             return false;
150         }
151
152         const auto it = mapperIt.value();
153         return (it != m_container.end());
154     }
155
156     bool remove(const key_type & key) noexcept
157     {
158         const auto mapperIt = m_mapper.find(key);
159         if (mapperIt == m_mapper.end()) {
160             return false;
161         }
162
163         const auto it = mapperIt.value();
164         Q_UNUSED(m_container.erase(it))
165         Q_UNUSED(m_mapper.erase(mapperIt))
166
167         if (m_currentSize != 0) {
168             --m_currentSize;
169         }
170
171         return true;
172     }
173
174     void setMaxSize(const size_t maxSize)
175     {
176         if (maxSize >= m_maxSize) {
177             m_maxSize = maxSize;
178             return;
179         }
180
181         size_t diff = m_maxSize - maxSize;
182         for (size_t i = 0; (i < diff) && !m_container.empty(); ++i) {
183             auto lastIt = m_container.end();
184             --lastIt;
185
186             const key_type & lastElementKey = lastIt->first;
187             Q_UNUSED(m_mapper.remove(lastElementKey))
188             Q_UNUSED(m_container.erase(lastIt))
189
190             if (m_currentSize != 0) {
191                 --m_currentSize;
192             }
193         }
194     }
195
196 private:
197     void fixupSize()
198     {
199         if (m_currentSize <= m_maxSize) {
200             return;
201         }
202
203         if (Q_UNLIKELY(m_container.empty())) {
204             return;
205         }
206
207         auto lastIt = m_container.end();
208         --lastIt;
209
```

```
210         const key_type & lastElementKey = lastIt->first;
211
212         Q_UNUSED(m_mapper.remove(lastElementKey))
213         Q_UNUSED(m_container.erase(lastIt))
214
215         if (m_currentSize != 0) {
216             --m_currentSize;
217         }
218     }
219
220 private:
221     mutable container_type m_container;
222     size_t m_currentSize = 0;
223     size_t m_maxSize;
224
225     mutable QHash<Key, iterator> m_mapper;
226 };
227
228 } // namespace quentier
```

6.101 MessageBox.h

```
1 /*
2 * Copyright 2017-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QMessageBox>
24
25 namespace quentier {
26
27 int QUENTIER_EXPORT genericMessageBox(
28     QWidget * parent, const QString & title, const QString & briefText,
29     const QString & detailedText = {},
30     QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
31
32 int QUENTIER_EXPORT informationMessageBox(
33     QWidget * parent, const QString & title, const QString & briefText,
34     const QString & detailedText = {},
35     QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
36
37 int QUENTIER_EXPORT warningMessageBox(
38     QWidget * parent, const QString & title, const QString & briefText,
39     const QString & detailedText = {},
40     QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
41
42 int QUENTIER_EXPORT criticalMessageBox(
43     QWidget * parent, const QString & title, const QString & briefText,
44     const QString & detailedText = {},
45     QMessageBox::StandardButtons standardButtons = QMessageBox::Ok);
46
47 [[nodiscard]] int QUENTIER_EXPORT questionMessageBox(
48     QWidget * parent, const QString & title, const QString & briefText,
49     const QString & detailedText = {},
50     QMessageBox::StandardButtons standardButtons = QMessageBox::Ok |
51         QMessageBox::Cancel);
52
58 void QUENTIER_EXPORT
59     internalErrorMessageBox(QWidget * parent, QString detailedText = {});
60
61 } // namespace quentier
```

6.102 Printable.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QDebug>
24 #include <QHash>
25 #include <QIODevice>
26 #include <QSet>
27 #include <QString>
28 #include <QTextStream>
29
30 namespace quentier {
31
37 class QUENTIER_EXPORT Printable
38 {
39 public:
40     virtual ~Printable() noexcept;
41
42     virtual QTextStream & print(QTextStream & strm) const = 0;
43
44     [[nodiscard]] QString toString() const;
45
46     friend QUENTIER_EXPORT QTextStream & operator<<(
47         QTextStream & strm, const Printable & printable);
48
49     friend QUENTIER_EXPORT QDebug & operator<<(
50         QDebug & debug, const Printable & printable);
51 };
52
53 } // namespace quentier
54
55 // printing operators for existing classes not inheriting from Printable
56
57 template <class T>
58 [[nodiscard]] QString ToString(const T & object)
59 {
60     QString str;
61     QTextStream strm(&str, QIODevice::WriteOnly);
62     strm << object;
63     return str;
64 }
65
66 template <class TKey, class TValue>
67 [[nodiscard]] QString ToString(const QHash<TKey, TValue> & object)
68 {
69     QString str;
70     QTextStream strm(&str, QIODevice::WriteOnly);
71     strm << QStringLiteral("QHash:  \n");
72
73     using CIter = typename QHash<TKey, TValue>::const_iterator;
74     CIter hashEnd = object.end();
75     for (CIter it = object.begin(); it != hashEnd; ++it) {
76         strm << QStringLiteral("[") << it.key() << QStringLiteral("] = ")
77              << it.value() << QStringLiteral(";\n");
78     }
79     return str;
80 }
81
82 template <class T>
83 [[nodiscard]] QString ToString(const QSet<T> & object)
84 {
85     QString str;
86     QTextStream strm(&str, QIODevice::WriteOnly);
87     strm << QStringLiteral("QSet:  \n");
88
89     using CIter = typename QSet<T>::const_iterator;
90     CIter setEnd = object.end();
```

```
91     for (CIter it = object.begin(); it != setEnd; ++it) {
92         strm « QStringLiteral("[") « *it « QStringLiteral("];\n");
93     }
94     return str;
95 }
96
97 #define QUENTIER_DECLARE_PRINTABLE(type, ...)
       \
98 QUENTIER_EXPORT QTextStream & operator«(                                    \
99 QTextStream & strm, const type & obj);                                    \
100 inline QDebug & operator«(QDebug & debug, const type & obj)                \
101 {                                                                          \
102 debug « ToString<type, ##__VA_ARGS__>(obj);                               \
103 return debug;                                                             \
104 }
105 // QUENTIER_DECLARE_PRINTABLE
```

6.103 QuentierApplication.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QApplication>
24
25 namespace quentier {
26
27 class QUENTIER_EXPORT QuentierApplication :  public QApplication
28 {
29     Q_OBJECT
30 public:
31     QuentierApplication(int & argc, char * argv[]); // NOLINT
32     ~QuentierApplication() noexcept override;
33
34     [[nodiscard]] bool notify(QObject * pObject, QEvent * pEvent) override;
35     [[nodiscard]] bool event(QEvent * pEvent) override;
36 };
37
38 } // namespace quentier
```

6.104 QuentierUndoCommand.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
```

```
20
21 #include <quentier/types/ErrorString.h>
22
23 #include <QObject>
24 #include <QUndoCommand>
25
26 namespace quentier {
27
53 class QuentierUndoCommand :  public QObject, public QUndoCommand
54 {
55     Q_OBJECT
56 public:
57     QuentierUndoCommand(QUndoCommand * parent = nullptr);
58     QuentierUndoCommand(const QString & text, QUndoCommand * parent = nullptr);
59     ~QuentierUndoCommand() noexcept override;
60
61     void undo() final;
62     void redo() final;
63
64     [[nodiscard]] bool onceUndoExecuted() const noexcept
65     {
66         return m_onceUndoExecuted;
67     }
68
69 Q_SIGNALS:
70     void notifyError(ErrorString error);
71
72 protected:
73     virtual void undoImpl() = 0;
74     virtual void redoImpl() = 0;
75
76 private:
77     bool m_onceUndoExecuted = false;
78 };
79
80 } // namespace quentier
```

6.105 ShortcutManager.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/Account.h>
22 #include <quentier/utility/Linkage.h>
23
24 #include <QKeySequence>
25 #include <QObject>
26
27 namespace quentier {
28
29 QT_FORWARD_DECLARE_CLASS(ShortcutManagerPrivate)
30
31 class QUENTIER_EXPORT ShortcutManager :  public QObject
32 {
33     Q_OBJECT
34 public:
35     explicit ShortcutManager(QObject * parent = nullptr);
36
37     enum QuentierShortcutKey
38     {
39         NewNote = 5000,
40         NewTag,
41         NewNotebook,
42         NewSavedSearch,
43         AddAttachment,
44         SaveAttachment,
```

```
45          OpenAttachment,
46          CopyAttachment,
47          CutAttachment,
48          RemoveAttachment,
49          RenameAttachment,
50          AddAccount,
51          ExitAccount,
52          SwitchAccount,
53          AccountInfo,
54          NoteSearch,
55          NewNoteSearch,
56          ShowNotes,
57          ShowNotebooks,
58          ShowTags,
59          ShowSavedSearches,
60          ShowDeletedNotes,
61          ShowStatusBar,
62          ShowToolBar,
63          PasteUnformatted,
64          Font,
65          UpperIndex,
66          LowerIndex,
67          AlignLeft,
68          AlignCenter,
69          AlignRight,
70          AlignFull,
71          IncreaseIndentation,
72          DecreaseIndentation,
73          IncreaseFontSize,
74          DecreaseFontSize,
75          InsertNumberedList,
76          InsertBulletedList,
77          Strikethrough,
78          Highlight,
79          InsertTable,
80          InsertRow,
81          InsertColumn,
82          RemoveRow,
83          RemoveColumn,
84          InsertHorizontalLine,
85          InsertToDoTag,
86          EditHyperlink,
87          CopyHyperlink,
88          RemoveHyperlink,
89          Encrypt,
90          Decrypt,
91          DecryptPermanently,
92          BackupLocalStorage,
93          RestoreLocalStorage,
94          UpgradeLocalStorage,
95          LocalStorageStatus,
96          SpellCheck,
97          SpellCheckIgnoreWord,
98          SpellCheckAddWordToUserDictionary,
99          SaveImage,
100          AnnotateImage,
101          ImageRotateClockwise,
102          ImageRotateCounterClockwise,
103          Synchronize,
104          FullSync,
105          ImportFolders,
106          Preferences,
107          ReleaseNotes,
108          ViewLogs,
109          About,
110          UnknownKey = 100000
111      };
112
118      [[nodiscard]] QKeySequence shortcut(
119          int key, const Account & account, const QString & context = {}) const;
120
126      [[nodiscard]] QKeySequence shortcut(
127          const QString & nonStandardKey, const Account & account,
128          const QString & context = {}) const;
129
134      [[nodiscard]] QKeySequence defaultShortcut(
135          int key, const Account & account, const QString & context = {}) const;
136
141      [[nodiscard]] QKeySequence defaultShortcut(
142          const QString & nonStandardKey, const Account & account,
143          const QString & context = {}) const;
144
149      [[nodiscard]] QKeySequence userShortcut(
150          int key, const Account & account, const QString & context = {}) const;
151
156      [[nodiscard]] QKeySequence userShortcut(
157          const QString & nonStandardKey, const Account & account,
```

```
158         const QString & context = {}) const;
159
160 Q_SIGNALS:
161     void shortcutChanged(
162         int key, QKeySequence shortcut, const Account & account,
163         QString context);
164
165     void nonStandardShortcutChanged(
166         QString nonStandardKey, QKeySequence shortcut, const Account & account,
167         QString context);
168
169 public Q_SLOTS:
170     void setUserShortcut(
171         int key, QKeySequence shortcut, const Account & account,
172         QString context = {});
173
174     void setNonStandardUserShortcut(
175         QString nonStandardKey, QKeySequence shortcut, const Account & account,
176         QString context = {});
177
178     void setDefaultShortcut(
179         int key, QKeySequence shortcut, const Account & account,
180         QString context = {});
181
182     void setNonStandardDefaultShortcut(
183         QString nonStandardKey, QKeySequence shortcut, const Account & account,
184         QString context = {});
185
186 private:
187     ShortcutManagerPrivate * const d_ptr;
188     Q_DECLARE_PRIVATE(ShortcutManager)
189 };
190
191 } // namespace quentier
```

6.106 Size.h

```
1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QString>
24
25 namespace quentier {
26
36 [[nodiscard]] QString QUENTIER_EXPORT humanReadableSize(quint64 bytes);
37
38 } // namespace quentier
```

6.107 StandardPaths.h

```
1 /*
2 * Copyright 2017-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
```

```
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/types/Account.h>
22 #include <quentier/utility/Linkage.h>
23
28 #define LIBQUENTIER_PERSISTENCE_STORAGE_PATH                                   \
29 "LIBQUENTIER_PERSISTENCE_STORAGE_PATH"
30
31 namespace quentier {
32
41 [[nodiscard]] QString QUENTIER_EXPORT
42     applicationPersistentStoragePath(bool * pNonStandardLocation = nullptr);
43
53 [[nodiscard]] QString QUENTIER_EXPORT
54     accountPersistentStoragePath(const Account & account);
55
60 [[nodiscard]] QString QUENTIER_EXPORT applicationTemporaryStoragePath();
61
67 [[nodiscard]] QString QUENTIER_EXPORT homePath();
68
72 [[nodiscard]] QString QUENTIER_EXPORT documentsPath();
73
74 } // namespace quentier
```

6.108 StringUtils.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QList>
24 #include <QSet>
25 #include <QString>
26
27 namespace quentier {
28
29 QT_FORWARD_DECLARE_CLASS(StringUtilsPrivate)
30
31 class QUENTIER_EXPORT StringUtils
32 {
33 public:
34     StringUtils();
35     ~StringUtils() noexcept;
36
37     void removePunctuation(
38         QString & str, const QList<QChar> & charactersToPreserve = {}) const;
39
40     void removeDiacritics(QString & str) const;
41     void removeNewlines(QString & str) const;
42
43 private:
44     StringUtilsPrivate * const d_ptr;
45     Q_DECLARE_PRIVATE(StringUtils);
46 };
47
48 } // namespace quentier
```

6.109 SuppressWarnings.h

```
1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
22 // Common macros
24
25 #define STRINGIFY(a) #a
26
27 // Define empty macros doing nothing for supported compilers, they would be used
28 // as fallback when any of these compilers are not actually used
29
30 #define SAVE_WARNINGS
31
32 #define CLANG_SUPPRESS_WARNING(warning)
33 #define GCC_SUPPRESS_WARNING(warning)
34 #define MSVC_SUPPRESS_WARNING(warning)
35
36 #define RESTORE_WARNINGS
37
39 // Clang implementation
41
42 #if defined(__clang__)
43
44 #undef CLANG_SUPPRESS_WARNING
45
46 #define CLANG_SUPPRESS_WARNING(warning)                                        \
47 _Pragma(STRINGIFY(clang diagnostic ignored #warning))
48
49 #undef SAVE_WARNINGS
50
51 #define SAVE_WARNINGS _Pragma("clang diagnostic push")
52
53 #undef RESTORE_WARNINGS
54
55 #define RESTORE_WARNINGS _Pragma("clang diagnostic pop")
56
57 #endif // clang
58
60 // GCC implementation
62
63 // Clang can mimic gcc so need to ensure it's indeed gcc
64 #if defined(__GNUC__) && !defined(__clang__)
65
66 #undef GCC_SUPPRESS_WARNING
67
68 #define GCC_SUPPRESS_WARNING(warning)                                          \
69 _Pragma(STRINGIFY(GCC diagnostic ignored #warning))
70
71 #undef SAVE_WARNINGS
72
73 #define SAVE_WARNINGS _Pragma("GCC diagnostic push")
74
75 #undef RESTORE_WARNINGS
76
77 #define RESTORE_WARNINGS _Pragma("GCC diagnostic pop")
78
79 #endif // GCC
80
82 // MSVC implementation
84
85 #if defined(_MSC_VER)
86
87 #undef MSVC_SUPPRESS_WARNING
88
89 #define MSVC_SUPPRESS_WARNING(number) __pragma(warning(disable :  number))
90
91 #undef SAVE_WARNINGS
92
93 #define SAVE_WARNINGS __pragma(warning(push))
```

```
94
95 #undef RESTORE_WARNINGS
96
97 #define RESTORE_WARNINGS __pragma(warning(pop))
98
99 #endif // MSVC
```

6.110 SysInfo.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QString>
24
25 namespace quentier {
26
27 QT_FORWARD_DECLARE_CLASS(SysInfoPrivate)
28
29 class QUENTIER_EXPORT SysInfo
30 {
31 public:
32     SysInfo();
33     ~SysInfo() noexcept;
34
35     [[nodiscard]] qint64 pageSize();
36     [[nodiscard]] qint64 totalMemory();
37     [[nodiscard]] qint64 freeMemory();
38
39     [[nodiscard]] QString stackTrace();
40
41     [[nodiscard]] QString platformName();
42
43 private:
44     Q_DISABLE_COPY(SysInfo)
45
46 private:
47     SysInfoPrivate * const d_ptr;
48     Q_DECLARE_PRIVATE(SysInfo)
49 };
50
51 } // namespace quentier
```

6.111 System.h

```
1 /*
2 * Copyright 2020-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
```

```
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QString>
24 #include <QUrl>
25
26 namespace quentier {
27
31 [[nodiscard]] QString QUENTIER_EXPORT getCurrentUserName();
32
36 [[nodiscard]] QString QUENTIER_EXPORT getCurrentUserFullName();
37
41 void QUENTIER_EXPORT openUrl(const QUrl & url);
42
43 } // namespace quentier
```

6.112 TagSortByParentChildRelations.h

```
1 /*
2 * Copyright 2017-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QList>
24
25 namespace qevercloud {
26
27 class Tag;
28
29 } // namespace qevercloud
30
31 namespace quentier {
32
33 class ErrorString;
34
46 bool QUENTIER_EXPORT sortTagsByParentChildRelations(
47     QList<qevercloud::Tag> & tagList, ErrorString & errorDescription);
48
49 } // namespace quentier
```

6.113 MockIKeychainService.h

```
1 /*
2 * Copyright 2022-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
```

```
18
19 #include <quentier/utility/IKeychainService.h>
20
21 #include <gmock/gmock.h>
22
23 namespace quentier::utility::tests::mocks {
24
25 class MockIKeychainService :  public IKeychainService
26 {
27 public:
28     MOCK_METHOD(
29         QFuture<void>, writePassword,
30         (QString service, QString key, QString password), (override));
31
32     MOCK_METHOD(
33         QFuture<QString>, readPassword, (QString service, QString key),
34         (const, override));
35
36     MOCK_METHOD(
37         QFuture<void>, deletePassword, (QString service, QString key),
38         (override));
39 };
40
41 } // namespace quentier::utility::tests::mocks
```

6.114 UidGenerator.h

```
1 /*
2 * Copyright 2016-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <quentier/utility/Linkage.h>
22
23 #include <QString>
24 #include <QUuid>
25
26 namespace quentier {
27
28 class QUENTIER_EXPORT UidGenerator
29 {
30 public:
31     [[nodiscard]] static QString Generate();
32     [[nodiscard]] static QString UidToString(const QUuid & uid);
33 };
34
35 } // namespace quentier
```

6.115 Unreachable.h

```
1 /*
2 * Copyright 2022-2024 Dmitry Ivanov
3 *
4 * This file is part of libquentier
5 *
6 * libquentier is free software; you can redistribute it and/or modify
7 * it under the terms of the GNU Lesser General Public License as published by
8 * the Free Software Foundation, version 3 of the License.
9 *
10 * libquentier is distributed in the hope that it will be useful,
11 * but WITHOUT ANY WARRANTY; without even the implied warranty of
12 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
13 * GNU Lesser General Public License for more details.
14 *
```

```
15 * You should have received a copy of the GNU Lesser General Public License
16 * along with libquentier.  If not, see <http://www.gnu.org/licenses/>.
17 */
18
19 #pragma once
20
21 #include <QtGlobal>
22
23 #ifdef _MSC_VER
24 #define UNREACHABLE                                                    \
25 do {                                                                   \
26 Q_ASSERT(false);                                                       \
27 __assume(0);                                                           \
28 } while (false)
29 #else
30 #define UNREACHABLE                                                    \
31 do {                                                                   \
32 Q_ASSERT(false);                                                       \
33 __builtin_unreachable();                                               \
34 } while (false)
35 #endif
```

Index