Данное руководство к библиотеке QtReports. Она предназначена для генерации, экспорта отчетов.

Подготовка:

Соберите библиотеку по файлу INSTALL

Создайте проект с .pro файлом, скопируйте паки qtreport и qtreportslib в папку с проектом.

В .pro файле пропишите:

QT += sql printsupport widgets

CONFIG +=c++11

INCLUDEPATH +="путь до\qtreportslib"

LIBS += "путь до\qtreport\libqtreports.dll"

Теперь подключением #include "engine.hpp" в файле проекта, вы можете использовать генератор отчетов.

Пример использования библиотеки:

С подключением к базе данных.

qtreports::Engine engine;//Создаем обьект движка отчета

engine.open( "file.xml" );//Открываем .xml файл отчета

|  |
| --- |
|  |
|  |
|  |

QSqlDatabase db = QSqlDatabase::addDatabase( "QSQLITE" ); //Создаем подключение  
db.setDatabaseName( "testDB" );//Выбираем базу данных  
db.open();//Открываем соединение

engine.setConnection( db );//Передаем соединение с базой данных

engine.createHTML("test.html");//Создаем отчет в формате html engine.createPDF("test.pdf");//Или создаем в формате pdf

db.close();//Закрываем соединение с базой данных

В случаи если результат не совпадает с запланированным или программа работает не корректно, можно воспользоваться функцией engine.getLastError(), которая вернет, в случаи возникновения, ошибку.

Помимо базы данных в качестве источника данных, библиотека может принимать абстрактные типы данных.

Пример:

qtreports::Engine engine;//Создаем обьект движка отчета

engine.open( "file.xml" );//Открываем .xml файл отчета

Создать собственный источник данных (QMap<QString, QVector < QVariant ») и заполнить его

engine.setDataSource( map );

engine.createHTML("test.html");//Создаем отчет в формате html engine.createPDF("test.pdf");//Или создаем в формате pdf

db.close();//Закрываем соединение с базой данных