

Qt síť

Vytvořil: Jan Svárovský 4.Ai 2017/18

Popis a návrh síťové aplikace

- QT += network

```
#include <QtNetwork>
```

- Třída QtNetwork poskytuje souhrn funkcí pro aplikace, které využívají pro komunikaci standard TCP/IP.
- Operace, jako načítání webových zdrojů, cookies a přenos dat přes HTTP, jsou zprostředkovány pomocí různých tříd C++.

`#include <QNetworkAccessManager>` - Umožňuje odesílat požadavky a přijímat odpovědi na běžně používaných protokolech

Síť ve Qt - HTTP

`#include <QNetworkRequest>` - Vytváří požadavek, který předá na `NetworkAccessManager`

`#include <QNetworkReply>` - Obsahuje data a hlavičky HTTP odpovědi

`#include <QUrl>` - Uchovává a slouží k práci s webovými adresami

mainwindow.h

```
private slots:  
    void downloadFinished(QNetworkReply *reply);  
QNetworkAccessManager manager;
```

mainwindow.cpp

```
ApiClient c;  
c.setMethod("get_user_info");  
c.addQuery("username", ui->lineEditUsername->text());  
QNetworkRequest request;  
qDebug() << c.getUrl();  
request.setUrl(c.getUrl());  
manager.get(request);  
c.clear();  
connect(&manager, SIGNAL(finished(QNetworkReply*)), SLOT(downloadFinished(QNetworkReply*)));  
  
void MainWindow::downloadFinished(QNetworkReply *reply)  
{  
    QByteArray text = reply->readAll();
```

apiclient.h

```
QUrlQuery query;  
QUrl getUrl();  
void clear();
```

apiclient.cpp

```
QUrl url("https://ue.zubro.net/rest/");  
query.addQueryItem("method", getMethod());  
query.addQueryItem("format", "json");  
url.setQuery(query);  
return url;
```

Sít' ve Qt - TCP server

`#include <QTcpServer>` - Umožňuje provoz TCP serveru

`#include <QHostAddress>` - Zpracovává a uchovává IP adresy

`#include <QHostInfo>` - Slouží k překladi adresy na název pomocí DNS

- Umožňuje přijímat příchozí připojení TCP
- Port serveru můžete zadat „napevno“ nebo ho nechat vybrat automaticky
- Příchozí připojení lze naslouchat na konkrétní adrese nebo na všech adresách zařízení

```
QTcpServer *tcpServer = nullptr;
```

```
tcpServer = new QTcpServer(this);  
if (!tcpServer->listen()) {
```

```
QList<QHostAddress> ipAddressesList = QNetworkInterface::allAddresses();
```

```
connect(tcpServer, &QTcpServer::newConnection, this, &Server::sendFortune);
```

```
void Server::sendFortune()  
{  
    QByteArray block;  
    QDataStream out(&block, QIODevice::WriteOnly);  
    out.setVersion(QDataStream::Qt_5_10);  
  
    out << fortunes[QRandomGenerator::global()->bounded(fortunes.size())];
```

```
QTcpSocket *clientConnection = tcpServer->nextPendingConnection();  
connect(clientConnection, &QAbstractSocket::disconnected,  
        clientConnection, &QObject::deleteLater);
```

```
clientConnection->write(block);  
clientConnection->disconnectFromHost();  
}
```

Sít' ve Qt - TCP klient

`#include <QTcpSocket>` - Umožňuje provoz TCP serveru, dědí od `QAbstractSocket`

- Umožňuje připojení k serveru pomocí protokolu TCP
- Zadáte adresu a port serveru a pošlete požadavek na připojení
- Pomocí „connect“ přijímáte příchozí data

```
QTcpSocket *tcpSocket = nullptr;  
QDataStream in;
```

```
connect(tcpSocket, &QIODevice::readyRead, this, &Client::readFortune);
```

```
void Client::readFortune()  
{  
    in.startTransaction();  
  
    QString nextFortune;  
    in >> nextFortune;  
  
    if (!in.commitTransaction())  
        return;
```

Kahoot!