**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Отчет

По лабораторной работе №7

На тему: “Исследование способов реализации сетевых запросов в QT-приложениях”

По дисциплине: “Кроссплатформенное программирование”

Выполнил:

ст.гр. ИС/б-17-2

Долженко И.А.

Проверил:

Строганов В.А.

Севастополь

2021

1 ЦЕЛЬ РАБОТЫ

Исследование способов работы Qt-приложения с HTTP GET/POST запросами. Приобретение навыков разработки простейших сетевых приложений в Qt.

2 ПОСТАНОВКА ЗАДАЧИ

2.1. Изучить основные возможности модуля QtNetwork для реализации HTTP-запросов (выполняется в ходе самостоятельной подготовки к лабораторной работе).

2.2. Создать Qt GUI приложение.

2.3. В дизайнере создать форму и реализовать логику приложения по инструкции из методических указаний.

2.4. Исследовать работу программы при отправке GET и POST запросов, проанализировать формат получаемых HTTP-ответов.

3 ТЕКСТ ПРОГРАММЫ

mainwindow.h:

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QtNetwork/QNetworkAccessManager>

#include <QtNetwork/QNetworkReply>

#include <QByteArray>

QT\_BEGIN\_NAMESPACE

namespace **Ui** { class **MainWindow**; }

QT\_END\_NAMESPACE

class **MainWindow** : public QMainWindow

{

Q\_OBJECT

public:

**MainWindow**(QWidget \*parent = nullptr);

~***MainWindow***();

private:

Ui::MainWindow \*ui;

QNetworkAccessManager \*nam;

private slots:

void **hideForm**();

void **showGetForm**();

void **showPostForm**();

void **submit**();

void **reset**();

void **finished**(QNetworkReply \*reply);

};

#endif // MAINWINDOW\_H

mainwindow.cpp:

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::**MainWindow**(QWidget \*parent)

: QMainWindow(*parent*)

, ui(new Ui::MainWindow)

{

ui->setupUi(this);

nam = new QNetworkAccessManager(this);

hideForm();

connect(ui->buttonGet, SIGNAL(clicked()), this, SLOT(showGetForm()));

connect(ui->buttonPost, SIGNAL(clicked()), this, SLOT(showPostForm()));

connect(ui->buttonSubmit, SIGNAL(clicked()), this, SLOT(submit()));

connect(ui->buttonReset, SIGNAL(clicked()), this, SLOT(reset()));

connect(ui->buttonMenu, SIGNAL(clicked()), this, SLOT(hideForm()));

connect(ui->buttonMenu, SIGNAL(clicked()), this, SLOT(reset()));

connect(nam, SIGNAL(finished(QNetworkReply\*)), this, SLOT(finished(QNetworkReply\*)));

}

MainWindow::~***MainWindow***()

{

delete ui;

}

void MainWindow::**hideForm**()

{

ui->buttonGet->setHidden(false);

ui->buttonPost->setHidden(false);

ui->buttonSubmit->setHidden(true);

ui->buttonReset->setHidden(true);

ui->label\_2->setHidden(true);

ui->label\_3->setHidden(true);

ui->label\_4->setHidden(true);

ui->lineEditUrl->setHidden(true);

ui->lineEditData->setHidden(true);

ui->plainTextEdit->setHidden(true);

ui->buttonMenu->setHidden(true);

}

void MainWindow::**showGetForm**()

{

ui->buttonGet->setHidden(true);

ui->buttonPost->setHidden(true);

ui->buttonSubmit->setHidden(false);

ui->buttonReset->setHidden(false);

ui->label\_2->setHidden(false);

ui->label\_4->setHidden(false);

ui->lineEditUrl->setHidden(false);

ui->plainTextEdit->setHidden(false);

ui->buttonMenu->setHidden(false);

}

void MainWindow::**showPostForm**()

{

ui->buttonGet->setHidden(true);

ui->buttonPost->setHidden(true);

ui->buttonSubmit->setHidden(false);

ui->buttonReset->setHidden(false);

ui->label\_2->setHidden(false);

ui->label\_3->setHidden(false);

ui->label\_4->setHidden(false);

ui->lineEditUrl->setHidden(false);

ui->lineEditData->setHidden(false);

ui->plainTextEdit->setHidden(false);

ui->buttonMenu->setHidden(false);

}

void MainWindow::**submit**()

{

QString url = ui->lineEditUrl->text();

QString data = ui->lineEditData->text();

QByteArray postData;

postData.append(data.toUtf8());

if (postData.isEmpty() == true) {

nam->get(QNetworkRequest(QUrl(url)));

}

else

{

nam->post(QNetworkRequest(QUrl(url)), postData);

}

}

void MainWindow::**reset**()

{

ui->lineEditUrl->setText("");

ui->lineEditData->setText("");

ui->plainTextEdit->setPlainText("");

}

void MainWindow::**finished**(QNetworkReply \*reply)

{

if(reply->error() == QNetworkReply::NoError)

{

ui->plainTextEdit->setPlainText(reply->readAll());

}

else

{

ui->plainTextEdit->setPlainText(reply->errorString());

}

}

4 РЕЗУЛЬТАТЫ ВЫПОЛНЕНИЯ ПРОГРАММЫ

Начальное окно приложения представлено на рисунке 1. На нем располагаются две кнопки: Get и Post, каждая из которых отображает соответствующую форму.

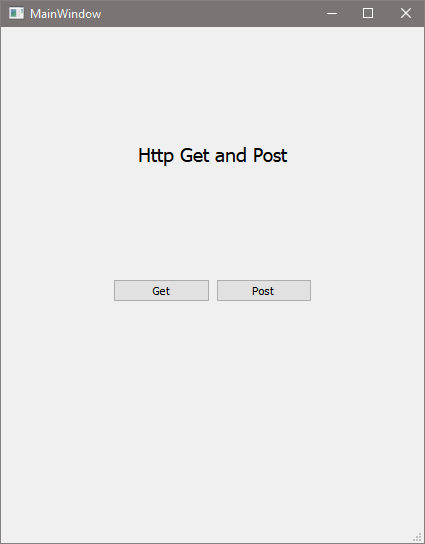


Рисунок 1 – Главное меню приложения

Выполним Get запрос на <https://reqres.in/api/users?page=2>. Результат выполнения запроса представлен на рисунке 2.

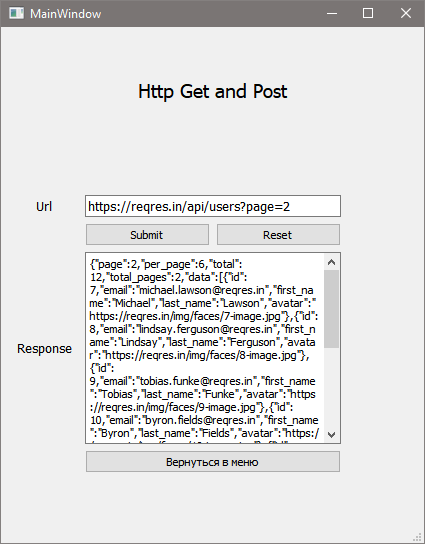


Рисунок 2 – Ответ на Get запрос

Теперь выполним Post запрос на <https://reqres.in/api/users/> с данными “ name="morpheus"&job="leader"”. Результат выполнения запроса представлен на рисунке 3.

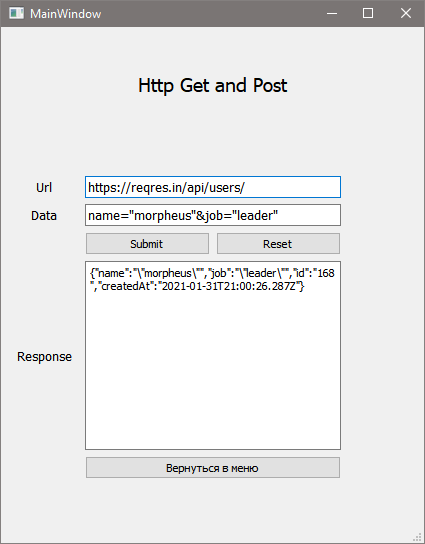


Рисунок 3 – Ответ на Post запрос

ВЫВОДЫ

В ходе выполнения лабораторной работы была написана программа, которая выполняет HTTP GET/POST запросы с помощью класса QNetworkAccessManager. Он содержит общую конфигурацию и настройки для посылаемых запросов. Также был изучен объект QNetworkReply, который используется для получения любых данных, возвращаемых в ответ на соответствующий запрос.