

105 Qt的HTTP编程

105.1 Qt的HTTP类

为支持基于HTTP协议的网络编程，Qt提供了以下类：

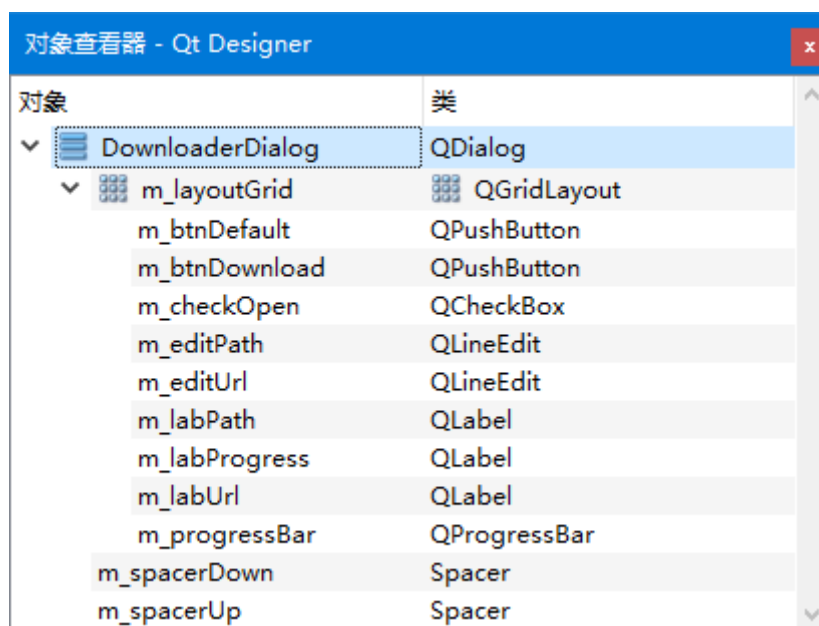
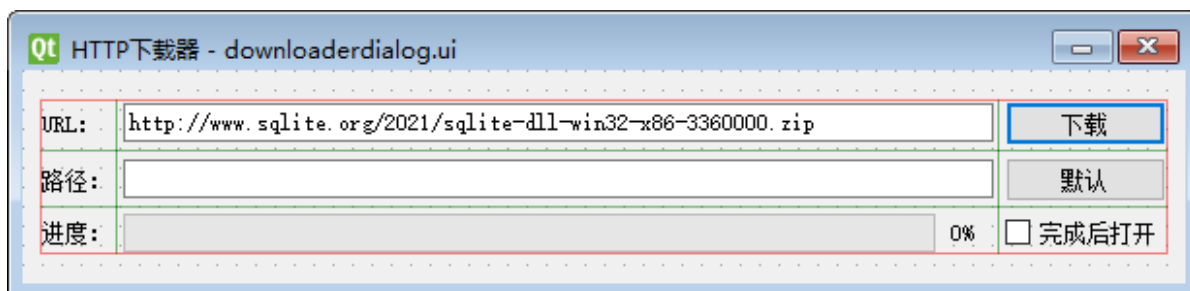
与HTTP有关的类	功能描述
QNetworkRequest	面向URL的网络请求，包括请求的内容，目前支持HTTP、FTP等
QNetworkAccessManager	协调网络操作，实际执行发送请求和接收响应的动作
QNetworkReply	表示网络响应，借助downloadProgress、readyRead和finished等信号，监测网络响应的传输情况，并执行相应操作

105.2 案例

105.2.1 创建项目

通过QtCreator，在C:\Users\Minwei\Projects\Qt路径下，创建名为Downloader的项目。

105.2.2 设计界面



C:\Users\Minwei\Projects\Qt\Downloader\downloaderdialog.ui:

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <ui version="4.0">
3   <class>DownloaderDialog</class>
4   <widget class="QDialog" name="DownloaderDialog">
```

```
5 <property name="geometry">
6 <rect>
7 <x>0</x>
8 <y>0</y>
9 <width>600</width>
10 <height>100</height>
11 </rect>
12 </property>
13 <property name="windowTitle">
14 <string>HTTP下载器</string>
15 </property>
16 <layout class="QVBoxLayout" name="m_layoutVer">
17 <item>
18 <spacer name="m_spacerUp">
19 <property name="orientation">
20 <enum>Qt::Vertical</enum>
21 </property>
22 <property name="sizeHint" stdset="0">
23 <size>
24 <width>20</width>
25 <height>74</height>
26 </size>
27 </property>
28 </spacer>
29 </item>
30 <item>
31 <layout class="QGridLayout" name="m_layoutGrid">
32 <item row="0" column="0">
33 <widget class="QLabel" name="m_labUrl">
34 <property name="text">
35 <string>URL: </string>
36 </property>
37 </widget>
38 </item>
39 <item row="0" column="1">
40 <widget class="QLineEdit" name="m_editUrl">
41 <property name="text">
42 <string>http://www.sqlite.org/2021/sqlite-dll-win32-x86-
3360000.zip</string>
43 </property>
44 </widget>
45 </item>
46 <item row="0" column="2">
47 <widget class="QPushButton" name="m_btnDownload">
48 <property name="text">
49 <string>下载</string>
50 </property>
51 <property name="default">
52 <bool>true</bool>
53 </property>
54 </widget>
55 </item>
56 <item row="1" column="0">
57 <widget class="QLabel" name="m_labPath">
58 <property name="text">
59 <string>路径: </string>
```

```
60     </property>
61   </widget>
62 </item>
63 <item row="1" column="1">
64   <widget class="QLineEdit" name="m_editPath"/>
65 </item>
66 <item row="1" column="2">
67   <widget class="QPushButton" name="m_btnDefault">
68     <property name="text">
69       <string>默认</string>
70     </property>
71   </widget>
72 </item>
73 <item row="2" column="0">
74   <widget class="QLabel" name="m_labProgress">
75     <property name="text">
76       <string>进度: </string>
77     </property>
78   </widget>
79 </item>
80 <item row="2" column="1">
81   <widget class="QProgressBar" name="m_progressBar">
82     <property name="value">
83       <number>0</number>
84     </property>
85   </widget>
86 </item>
87 <item row="2" column="2">
88   <widget class="QCheckBox" name="m_checkOpen">
89     <property name="text">
90       <string>完成后打开</string>
91     </property>
92   </widget>
93 </item>
94 </layout>
95 </item>
96 <item>
97   <spacer name="m_spacerDown">
98     <property name="orientation">
99       <enum>Qt::Vertical</enum>
100    </property>
101    <property name="sizeHint" stdset="0">
102      <size>
103        <width>20</width>
104        <height>0</height>
105      </size>
106    </property>
107  </spacer>
108 </item>
109 </layout>
110 </widget>
111 <resources/>
112 <connections/>
113 </ui>
```

105.2.3 实现功能

C:\Users\Minwei\Projects\Qt\Downloader\downloaderdialog.h:

```
1  #ifndef DOWNLOADERDIALOG_H
2  #define DOWNLOADERDIALOG_H
3
4  #include <QDialog>
5  #include <QNetworkAccessManager>
6  #include <QNetworkReply>
7  #include <QFile>
8
9  QT_BEGIN_NAMESPACE
10 namespace Ui { class DownloaderDialog; }
11 QT_END_NAMESPACE
12
13 class DownloaderDialog : public QDialog
14 {
15     Q_OBJECT
16
17 public:
18     DownloaderDialog(QWidget *parent = nullptr);
19     ~DownloaderDialog();
20
21 private slots:
22     void on_m_btnDownload_clicked();
23     void on_m_btnDefault_clicked();
24
25     void on_m_reply_downloadProgress(
26         qint64 bytesReceived, qint64 bytesTotal);
27     void on_m_reply_readyRead();
28     void on_m_reply_finished();
29
30 private:
31     Ui::DownloaderDialog *ui;
32     QNetworkAccessManager* m_access;
33     QNetworkReply* m_reply;
34     QFile* m_file;
35 };
36
37 #endif // DOWNLOADERDIALOG_H
```

C:\Users\Minwei\Projects\Qt\Downloader\downloaderdialog.cpp:

```
1  #include <QMessageBox>
2  #include <QDir>
3  #include <QDesktopServices>
4  #include <QThread>
5
6  #include "downloaderdialog.h"
7  #include "ui_downloaderdialog.h"
8
9  DownloaderDialog::DownloaderDialog(QWidget *parent)
10     : QDialog(parent)
11     , ui(new Ui::DownloaderDialog)
```

```

12     , m_access(new QNetworkAccessManager(this))
13     , m_reply(Q_NULLPTR)
14     , m_file(Q_NULLPTR)
15 {
16     ui->setupUi(this);
17
18     on_m_btnDefault_clicked();
19 }
20
21 DownloaderDialog::~DownloaderDialog()
22 {
23     delete ui;
24 }
25
26 void DownloaderDialog::on_m_btnDownload_clicked()
27 {
28     QUrl url = QUrl::fromUserInput(ui->m_editUrl->text().trimmed());
29     if(!url.isValid())
30     {
31         QMessageBox::critical(this, "URL错误", url.errorString());
32         return;
33     }
34
35     m_file = new QFile(ui->m_editPath->text().trimmed() + "/" +
url.fileName());
36     if(!m_file->open(QIODevice::WriteOnly))
37     {
38         QMessageBox::critical(this, "文件错误",
39             QString("打开“%1”文件失败! ").arg(m_file->fileName()));
40         return;
41     }
42
43     m_reply = m_access->get(QNetworkRequest(url));
44
45     connect(m_reply, SIGNAL(downloadProgress(qint64, qint64)),
46         this, SLOT(on_m_reply_downloadProgress(qint64,qint64)));
47     connect(m_reply, SIGNAL(readyRead()),
48         this, SLOT(on_m_reply_readyRead()));
49     connect(m_reply, SIGNAL(finished()),
50         this, SLOT(on_m_reply_finished()));
51
52     ui->m_btnDownload->setEnabled(false);
53 }
54
55 void DownloaderDialog::on_m_btnDefault_clicked()
56 {
57     ui->m_editPath->setText(QDir::currentPath());
58 }
59
60 void DownloaderDialog::on_m_reply_downloadProgress(
61     qint64 bytesReceived, qint64 bytesTotal)
62 {
63     ui->m_progressBar->setMaximum(bytesTotal);
64     ui->m_progressBar->setValue(bytesReceived);
65 }
66

```

```
67 void DownloaderDialog::on_m_reply_readyRead()
68 {
69     m_file->write(m_reply->readAll());
70 }
71
72 void DownloaderDialog::on_m_reply_finished()
73 {
74     m_reply->deleteLater();
75     m_reply = Q_NULLPTR;
76
77     m_file->close();
78
79     if (ui->m_checkOpen->isChecked())
80         QDesktopServices::openUrl(QUrl::fromLocalFile(m_file->fileName()));
81
82     delete m_file;
83     m_file = Q_NULLPTR;
84
85     QThread::sleep(1);
86     ui->m_btnDownload->setEnabled(true);
87     ui->m_progressBar->setValue(0);
88 }
```

105.2.4 测试验证

运行效果如图所示：

