Creating an HTTP network request in Qt

Overview

This code snippet demonstrates how to use `QNetworkAccessManager` for sending an HTTP request.

Note: In order to use this code, you need to have Qt installed on your platform.

Header

```
#include <QNetworkAccessManager>
#include <QUrl>
#include <QNetworkRequest>
#include <QNetworkReply>

QNetworkAccessManager* nam;
```

Source

Create `QNetworkAccessManager` and start listening for its `finished` signal.

```
nam = new QNetworkAccessManager(this);
QObject::connect(nam, SIGNAL(finished(QNetworkReply*)),
         this, SLOT(finishedSlot(QNetworkReply*)));
```

HTTP GET request:

```
QUrl url("http://www.forum.nokia.wiki");
QNetworkReply* reply = nam->get(QNetworkRequest(url));
// NOTE: Store QNetworkReply pointer (maybe into caller).
// When this HTTP request is finished you will receive this same
// QNetworkReply as response parameter.
// By the QNetworkReply pointer you can identify request and response.
```

When the `QNetworkAccessManager::finished` signal is received, the HTTP request is completed.

```
void MyHttpEngine::finishedSlot(QNetworkReply* reply)
{
    // Reading attributes of the reply
    // e.g. the HTTP status code
    QVariant statusCodeV =
    reply->attribute(QNetworkRequest::HttpStatusCodeAttribute);
    // Or the target URL if it was a redirect:
    QVariant redirectionTargetUrl =
    reply->attribute(QNetworkRequest::RedirectionTargetAttribute);
    // see CS001432 on how to handle this

    // no error received?
    if (reply->error() == QNetworkReply::NoError)
    {
        // read data from QNetworkReply here

        // Example 1: Creating QImage from the reply
        QImageReader imageReader(reply);
        QImage pic = imageReader.read();

        // Example 2: Reading bytes form the reply
        QByteArray bytes = reply.readAll();  // bytes
        QString string(bytes); // string
    }
    // Some http error received
    else
    {
        // handle errors here
    }

    // We receive ownership of the reply object
    // and therefore need to handle deletion.
    delete reply;
}
```

See also

- For more information about `QNetworkAccessManager`, see QNetworkAccessManager Class Reference
- Handling an HTTP redirect with QNetworkAccessManager

(C) Copyright Nokia 2013. All rights reserved.

Postconditions

An HTTP response is received.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.