Clearing cookies from a QML WebView

This article shows how to clear cookies from a QML WebView

Overview

In some cases, you need to clear the cookies (or the cache) from a (QML) WebView. In general, the `WebView` doesn't have persistent storage for cookies after the session ended, that is when you quit the application (a solution to implement persistent cookies is described in QTBUG-16449). However, it can be useful to clear cookies while the application is running, for example if you need to do Facebook authentication in a `WebView`.

In Qt C++, cookies and the cache are controlled by the QNetworkAccessManager used by the QWebView. This network access manager doesn't have its direct representation in QML; however, it's possible to get a pointer to the instance of the access manager that QML is using to access the Internet. Through this instance, you can then replace the cookie jar of the access manager with a newly constructed (= empty) object, which essentially cleans out all the cookies. If you want to have more control and only delete specific cookies, you would need to provide your own QNetworkCookieJar subclass; but this is not the focus of this article.

To trigger the replacement of the cookie jar from QML, you need to glue it to C++. This can be done in the standard way how to call a C++ method from QML. In short: create a QObject instance, mark the C++ method as Q_INVOKABLE and set your class instance as a context property to the declarative engine. Then, you can directly call your cookie-jar-replacement-method from within QML.

Example Application

The small example app is based on the standard Qt Quick Application wizard template from Qt Creator and demonstrates the whole process:

- Start the app and wait for the web page to load (by default a 3rd party cookie test page).
- In the web page, click the "Set Test Cookie" button and wait for the page to reload, listing the cookie as set.
- Now, click the red banner at the top of the page to clear the cookies and to reload the page.
- Note that the cookie is gone.

Download the example app: File:WebviewCookies.zip

NetworkAccessManagerInteractor

This class is set up in a way so that it can be triggered from QML, and has a method called `clearCookies()` that replaces the cookie jar from the network access manager used by the web view with a new and empty one, essentially clearing all cookies. The ownership of the new `QNetworkCookieJar` is transferred to the `QNetworkAccessManager`.

Header file:

```
class NetworkAccessManagerInteractor : public QObject
{
    Q_OBJECT
public:
    explicit NetworkAccessManagerInteractor(QDeclarativeEngine *declarativeEngine, QObject *parent = 0);

public slots:
    Q_INVOKABLE void clearCookies();

private:
    QDeclarativeEngine* m_declarativeEngine;
};
```

Code:

```
NetworkAccessManagerInteractor::NetworkAccessManagerInteractor(QDeclarativeEngine *declarativeEngine, QObject *parent) :
    QObject(parent),
    m_declarativeEngine(declarativeEngine)
{
}

void NetworkAccessManagerInteractor::clearCookies()
{
    QNetworkAccessManager * nam = m_declarativeEngine->networkAccessManager();
    if (nam) {
        QNetworkCookieJar * emptyCookieJar = new QNetworkCookieJar();
        // QNetworkAccessManager takes ownership of the cookieJar object.
        nam->setCookieJar(emptyCookieJar);
        qDebug() << "Replaced cookie jar";
    } else {
        qDebug() << "Unable to retrieve Network Access Manager from declarative engine";
    }
}
```

Registering the class with the declarative engine

The new class needs to be registered with the declarative engine at application startup, so that the `clearCookies()` method can be called from the QML environment.

(C) Copyright Nokia 2013. All rights reserved.

```
Q_DECL_EXPORT int main(int argc, char *argv[])
{
    QScopedPointer<QApplication> app(createApplication(argc, argv));

    QmlApplicationViewer viewer;

    NetworkAccessManagerInteractor* nami = new NetworkAccessManagerInteractor(viewer.engine(), &viewer);
    viewer.rootContext()->setContextProperty("nami", nami);

    viewer.setOrientation(QmlApplicationViewer::ScreenOrientationAuto);
    viewer.setMainQmlFile(QLatin1String("qml/WebviewCookies/main.qml"));
    viewer.showExpanded();

    return app->exec();
}
```

Triggering the clear cookies method from QML

The simple QML file contains a webview and a self-made red button; when clicked, it triggers the C++ method to clear the cookies and reloads the page in the webview.

```
// import QtQuick 1.0 // to target S60 5th Edition or Maemo 5
import QtQuick 1.1
import QtWebKit 1.0

Rectangle {
    id: rectangle2
    width: 360
    height: 360


    Rectangle {
        id: myButton
        height: 60
        color: "#ff0000"
        anchors.right: parent.right
        anchors.left: parent.left
        anchors.top: parent.top
        z: 1

        MouseArea {
            id: mouse_area1
            anchors.fill: parent
            onClicked: {
                nami.clearCookies();
                webView.reload.trigger();
            }

            Text {
                id: myButtonText
                color: "#fffb00"
                text: qsTr("Clear Coookies and Reload")
                font.bold: true
                anchors.fill: parent
                verticalAlignment: Text.AlignVCenter
                horizontalAlignment: Text.AlignHCenter
                font.pixelSize: 16
            }

        }
    }

    WebView {
        id: webView
        url: "http://www.html-kit.com/tools/cookietester/"
        anchors.fill: parent
    }

}
```

Download the example app: File:WebviewCookies.zip

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.