Showing a map in Qt using Nokia Maps

This code snippet demonstrates how to fetch a map tile in Qt using the Location module of Qt Mobility and Nokia Maps.

Introduction

This code snippet demonstrates how to fetch a map tile in Qt using the Location module of Qt Mobility and Nokia Maps. It is assumed here that you have set up Qt Mobility in your development environment and on your device. For more information, see Setting up Qt Mobility.

Qt project file

Add the WebKit and the Location modules into the project:

```
QT += webkit
CONFIG += mobility
MOBILITY = location
```

This snippet requires the following capabilities:

```
symbian: {
    TARGET.CAPABILITY = Location \
                        NetworkServices
}
```

Source: main.cpp

```
#include "mainwindow.h"

#include <QtGui/QApplication>

int main(int argc, char* argv[])
{
    QApplication app(argc, argv);
    MainWindow mainWindow;
    mainWindow.showMaximized();
    return app.exec();
}
```

Header: mainwindow.h

```
#include <QtCore/QPointer>
#include <QtCore/QUrl>
#include <QtGui/QMainWindow>
#include <QtWebKit/QWebView>

// Location API
#include <QGeoPositionInfo>
#include <QGeoPositionInfoSource>

// QtMobility namespace
QTM_USE_NAMESPACE
```


(C) Copyright Nokia 2014. All rights reserved.

```
class MainWindow : public QMainWindow
{
    Q_OBJECT

public:
    /**
     * Constructor.
     */
    MainWindow(QWidget* parent = 0);

    /**
     * Destructor.
     */
    ~MainWindow();

public slots:
    /**
     * Called when the current position is updated.
     */
    void positionUpdated(QGeoPositionInfo geoPositionInfo);

private:
    /**
     * Obtains the location data source and starts listening for position
     * changes.
     */
    void startGPS();

    /**
     * Creates the URL with which the map tile can be fetched.
     */
    QUrl createMapURL(const QSize& size, qreal latitude,
                      qreal longitude);

private:
    QPointer<QGeoPositionInfoSource> locationDataSource;
    QWebView* webView;
};
```

Source: mainwindow.cpp

```
#include "mainwindow.h"

#include <QGeoCoordinate>

MainWindow::MainWindow(QWidget* parent) : QMainWindow(parent)
{
    setWindowTitle("Nokia Maps Example");

    webView = new QWebView(this);
    setCentralWidget(webView);

    // Start the GPS
    startGPS();
```


(C) Copyright Nokia 2014. All rights reserved.

```
}

MainWindow::~MainWindow()
{
    webView->deleteLater();
}

void MainWindow::startGPS()
{
    // Location fetching omitted here for brevity. Refer to the See also
    // section for more information.

    // Whenever the current position is updated, the positionUpdated
    // function is called.
}

void MainWindow::positionUpdated(QGeoPositionInfo geoPositionInfo)
{
    if (geoPositionInfo.isValid())
    {
        // Stop regular position updates, because a valid position has been
        // obtained
        locationDataSource->stopUpdates();

        // Get the current location as latitude and longitude
        QGeoCoordinate location = geoPositionInfo.coordinate();
        qreal latitude = location.latitude();
        qreal longitude = location.longitude();

        // Fetch the map using the display size and the coordinates
        QUrl url = createMapURL(size(), latitude, longitude);
        webView->stop();     // Stop the ongoing request
        webView->load(url);  // Load the map
    }
}

QUrl MainWindow::createMapURL(const QSize& size, qreal latitude,
                              qreal longitude)
{
    const QString MAP_URL_TEMPLATE =
        "qrc:/map.html?centerx=%1&centery=%2&zoom=11&sizex=%3&sizey=%4";
    QUrl url = QUrl(MAP_URL_TEMPLATE.arg(
        QString::number(latitude), QString::number(longitude),
        QString::number(size.width()), QString::number(size.height())));
    return url;
}
```

Source: map.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <!-- Add the JavaScript file resource into the document -->
```


(C) Copyright Nokia 2014. All rights reserved.

```
        <script src="http://api.maps.ovi.com/jsl.js" type="text/javascript"
charset="utf-8"></script>
    </head>
    <body>
        <div id="mapContainer">
        </div>

        <script type="text/javascript">
        // Retrieve the GET parameters from the query string
        var oGetVars = {};

        function buildValue(sValue) {
            if (/^\s*$/.test(sValue)) {
                return(null);
            }
            if (/^(true|false)$/i.test(sValue)) {
                return(sValue.toLowerCase() === "true");
            }
            if (isFinite(sValue)) {
                return(parseFloat(sValue));
            }
            if (isFinite(Date.parse(sValue))) {
                return(new Date(sValue));
            }
            return(sValue);
        }

        if (window.location.search.length > 1) {
            var iCouple;
            var aCouples = window.location.search.substr(1).split("&");
            for (var iCouplId = 0; iCouplId < aCouples.length; iCouplId++) {
                iCouple = aCouples[iCouplId].split("=");
                oGetVars[unescape(iCouple[0])] = iCouple.length > 1 ?
                    buildValue(unescape(iCouple[1])) :
                    null;
            }
        } else {
            throw "No GET parameters.";
        }

        // Set the size of the map element
        var mapContainerElement = document.getElementById("mapContainer");
        var styleAttr = "width:" + oGetVars["sizex"] + "px; " +
                        "height:" + oGetVars["sizey"] + "px;"
        mapContainerElement.setAttribute("style", styleAttr);

        // Create the map and display it in mapContainerElement
        var centerCoords = [oGetVars["centerx"], oGetVars["centery"]];
        var map = new ovi.mapsapi.map.Display(
            mapContainerElement,
            {
                'zoomLevel': oGetVars["zoom"],
                'center': centerCoords
            }
        );

        // Create a marker and add it to the map
```


(C) Copyright Nokia 2014. All rights reserved.

```
        var marker = new ovi.mapsapi.map.StandardMarker(
            centerCoords,
            {
                text: "Y"
            }
        );
        map.objects.add(marker);
        </script>
    </body>
</html>
```

Add the map.html file to the resource file:

```
<RCC>
    <qresource prefix="/">
        <file>map.html</file>
    </qresource>
</RCC>
```

Summary

This code snippet demonstrated how to fetch a map from Nokia Maps, based on the current location, and show it on the display.

See also

- Setting up Qt Mobility
- Getting the location in Qt
- http://doc.qt.nokia.com/qtmobility/location-overview.html

(C) Copyright Nokia 2014. All rights reserved.