Archived:Google Maps With QWebView

Archived: This article is archived because it is not considered relevant for third-party developers creating commercial solutions today. If you think this article is still relevant, let us know by adding the template {{ReviewForRemovalFromArchive|user=~~~~|*write your reason here*}}.

Qt Quick should be used for all UI development on mobile devices. The approach described in this article (using C++ for the Qt app UI) is deprecated.

The QtWebKit module provides a web browser engine as well as classes to render and interact with web content. This article shows how to consume Google Maps using Qt WebKit module.

Introduction

29 May 2011

Qt's WebKit integration is a fantastic technology. Being able to add web content into our application is very simple and easy now, so that there just about anything out in the web that we can show with out compromising the look and feel. A nice way to look for this is there is a quite a number of lot of people know HTML, so there will be plenty of data / content that out application can consume.

Qt WebKit module is so much more than just a way to show HTML in our app. We can actually use this module to make an application that can do anything a browser can – and much more (with some more own logic added to it)

In this example we will try consuming the Google's Maps.

Prerequisites

Google Maps API requires that you sign up for a key:

```
<maps:Map xmlns:maps="com.google.maps.*" id="map" mapevent_mapready="onMapReady(event)" width="100%" height="100%"
  key="YOUR_GOOGLE_MAP_KEY"/>
```

Procedure

We will consider 2 ways of putting a marker on top of Google Map

1. With Name Only
2. With Geocode [Latitude / Longitude]

Step 01: Derive a Class from QWebView

```
class GoogleMapView : public QWebView
{
...
}
```

Step 02: Create QNetworkAccessManager member variable

```
m_netManager = new QNetworkAccessManager(this);
```

Step 03: Connect `QNetworkAccessManager` finished SIGNAL to GoogleMapView own SLOT

```
connect(m_netManager, SIGNAL(finished(QNetworkReply*)),
            this, SLOT(replyFinished(QNetworkReply*)));
```

Step 04: Request Coordinates for a Location Name (for example: "Citi Bank, Singapore") by using

```
QString requestStr( tr("http://maps.google.com/maps/geo?q=%1&output=%2&key=%3")
            .arg("YOUR_LOCATION")
            .arg("csv")
            .arg("GOOGLE_MAP_KEY")
```

Step 05: Catch the response from "200,1,1.3520830,103.8198360" we can find the Coordinates as (1.3520830,103.8198360)

```
QString replyStr( reply->readAll() );
    QStringList coordinateStrList = replyStr.split(",");

    if( coordinateStrList.size() == 4) {
        QPointF coordinate( coordinateStrList[2].toFloat(),coordinateStrList[3].toFloat() );
        coordinates << coordinate;
    }
```

Step 06: Now we can use the coordinates to put a marker on the map using Javascript

(C) Copyright Nokia 2013. All rights reserved.

```
var map = new GMap2(document.getElementById(\"map\"));"
var bounds = new GLatLngBounds;"
var markers = [];"
map.setCenter( new GLatLng(0,0),1 );";
markers[0] = new GMarker(new GLatLng(point.x(), point.y()));
for( var i=0; i<markers.length; ++i ) {
 bounds.extend(markers[i].getPoint());
 map.addOverlay(markers[i]);
 }
map.setCenter(bounds.getCenter());
```

Step 07: Now we will send this script to web page for evalute

```
 this->page()->mainFrame()->evaluateJavaScript( scriptStr );
```

Step 08: you can call Step04 with new location name

you can play with other parameters from http://code.google.com/apis/maps/documentation/javascript/v2/services.html

GoogleMapView.h

```
class QGoogleMapView : public QWebView
{
    Q_OBJECT

public:
    QGoogleMapView(QWidget *parent=0);

public slots:
    void replyFinished(QNetworkReply*);
    void loadCoordinates();
    void geoCode(const QString &address);
    void clearCoordinates();

signals:
    void reloadMap();

private:
    QNetworkAccessManager *manager;
    QList<QPointF> coordinates;
    int pendingRequests;
};
```

GoogleMapView.cpp

```
QGoogleMapView::QGoogleMapView(QWidget *parent) : QWebView(parent), pendingRequests(0)
{
    manager = new QNetworkAccessManager(this);
    connect(manager, SIGNAL(finished(QNetworkReply*)),
            this, SLOT(replyFinished(QNetworkReply*)));
    connect(this,SIGNAL(reloadMap()),
            this,SLOT(loadCoordinates()));

    //load(QUrl("http://chaos.troll.no/~hhartz/visualaccess/index.html"));
    geoCode("");
}

void QGoogleMapView::geoCode(const QString& address)
{
    clearCoordinates();
    QString requestStr( tr("http://maps.google.com/maps/geo?q=%1&output=%2&key=%3")
            .arg("citibank, singapore")
            .arg("csv")
            .arg("ABQIAAAADX-GI9AcnNpiPCPpHEOvyBTsWThlcmnAmd0F51atdZmM43x6-RR4YCYPrtwhf8O9wSWFJ3XkwjOHIw") );

    manager->get( QNetworkRequest(requestStr) );
    ++pendingRequests;
}

void QGoogleMapView::replyFinished(QNetworkReply *reply)
{
    if( reply->error() != QNetworkReply::NoError ) {
       qDebug () << reply->errorString();
       /* And probably we should do something in UI to reflect error state */
       return;
    }

    QString replyStr( reply->readAll() );
    QStringList coordinateStrList = replyStr.split(",");

    if( coordinateStrList.size() == 4 ) {
        QPointF coordinate( coordinateStrList[2].toFloat(),coordinateStrList[3].toFloat() );
        coordinates << coordinate;
    }

    --pendingRequests;
    if( pendingRequests<1 )
        emit( reloadMap() );
}

void QGoogleMapView::loadCoordinates()
{
    QStringList scriptStr;
    scriptStr
            << "var map = new GMap2(document.getElementById(\"map\"));"
            << "var bounds = new GLatLngBounds;"
            << "var markers = [];"
            << "map.setCenter( new GLatLng(0,0),1 );";

    int num=-1;
    foreach( QPointF point, coordinates ) {
        scriptStr << QString("markers[%1] = new GMarker(new GLatLng(%2, %3));")
                            .arg(++num)
                            .arg(point.x())
                            .arg(point.y());
    }

    scriptStr
            << "for( var i=0; i<markers.length; ++i ) {"
```


(C) Copyright Nokia 2013. All rights reserved.

```
            << "    bounds.extend(markers[i].getPoint());"
            << "    map.addOverlay(markers[i]);"
            <<     "}"
            <<     "map.setCenter(bounds.getCenter());";


    this->page()->mainFrame()->evaluateJavaScript( scriptStr.join("\n") );
}

void QGoogleMapView::clearCoordinates()
{
    coordinates.clear();
}
```

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.