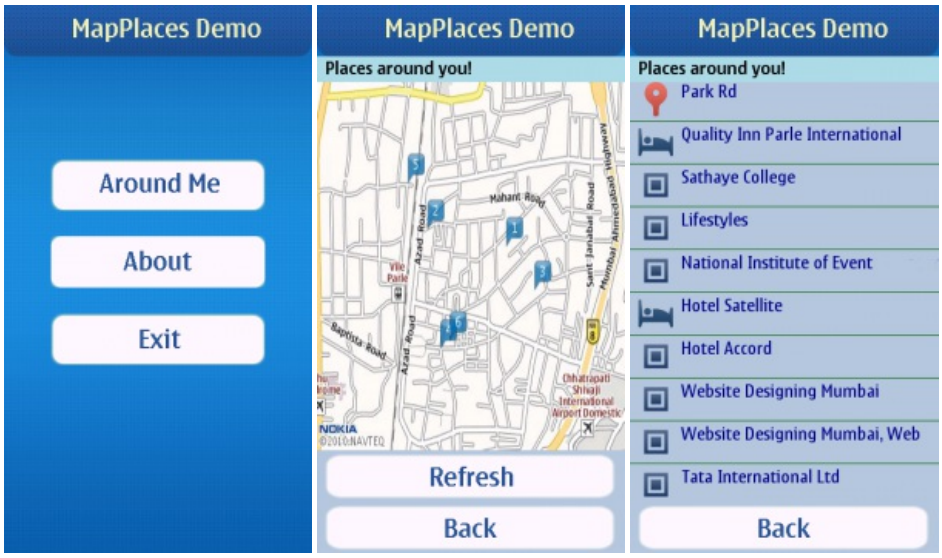
MapPlacesDemo - Plotting multiple POI on Nokia Maps

This article illustrates how to **show multiple points of interests (POI)** around you on Nokia Maps using Qt Quick.



Introduction

This article showcases a demo app - MapPlacesDemo - which shows the user - places around him, on Nokia Map. The app uses the current position of the user (lat-long from the integrated GPS receiver), get the places / points of interest (POI) around the user within 1 km range and displays them on map - using Nokia Map Image API. The app also gives the user more details like type, vicinity, etc of the point of interest; and flexibility to see each POI individually on map.

A video demonstration of MapPlacesDemo is shown below, The media player is loading...

Map Places API

MapPlacesDemo uses the Google Places API for fetching points of interest around the current position. The service provides list of POIs around a specified position (latitude and longitude) and withing a given radius (distance). The format for Google Places API request is:

```
   QString url="https://maps.googleapis.com/maps/api/place/search/"+FORMAT+"?
location="+lat+","+lng+"&radius="+RADIUS+"&sensor="+SENSOR+"&key="+GOOGLE_API_KEY;
```

Where,

- FORMAT: json/xml

json indicates output in JavaScript Object Notation (JSON) xml indicates output as XML MapPlacesDemo uses an xml format.

- SENSOR: Indicates whether or not the Place Details request came from a device using a location sensor (e.g. a GPS). This value must be either true or false.
- GOOGLE_API_KEY: Your application's API key. This key identifies your application for purposes of quota management so that places added from your application are made immediately available to your app. You can visit the Google APIs Console to create an API Project and obtain your key.

A HTTP GET request is done to the above url to get XML response,

```
void MapPlacesRequestor::GetPlaces(QString lat, QString lng)
{

    QString url = "https://maps.googleapis.com/maps/api/place/search/"
            +FORMAT+"?location="
            +lat+","
            +lng+"&radius="
```


(C) Copyright Nokia 2014. All rights reserved.

```
            +RADIUS+
            "&sensor="+
            SENSOR+
            "&key="
            +GOOGLE_API_KEY;
    nam->get(QNetworkRequest(QUrl(url)));
}
```

Once the XML response is received, we use parseResponse function, which uses Qt Dom parser and creates a list of Places. The Places class has attributes like lat, lng, name, vicinity and type of place; which will be presented to the user.

```
class Places{
public:
    Places(QString aname, QString avicinity, QString atype, QString alat, QString alng)
    {
        name = aname;
        vicinity = avicinity;
        lat = alat;
        lng = alng;
        type = atype;
    }
    QString name, vicinity, type;
    QString lat, lng;
};
```

`parseResponse` creates a list of places (`QList<Places*>`) as shown below,

```
int MapPlacesRequestor::parseResponse(QString PlacesData)
{
    iPlacesList.clear();
    QDomDocument doc;

    if( !doc.setContent(PlacesData))
    {
        return -2;
    };

    QDomElement root = doc.documentElement();

    QDomNode n = root.firstChild();

    while( !n.isNull() )
    {
        QDomElement e = n.toElement();

        if (e.tagName()=="result")
        {
   // Attributes of a place
            QString pname, pvicinity, plat, plng, ptype;
            QDomNode result_n = e.firstChild();

            while(!result_n.isNull())
            {
                QDomElement result_e = result_n.toElement();
```


(C) Copyright Nokia 2014. All rights reserved.

```
                if (result_e.tagName() == "name")
                {
                    pname = result_e.text();
                }
                else if (result_e.tagName() == "vicinity")
                {
                    pvicinity = result_e.text();
                }
                else if (result_e.tagName() == "type")
                {
                    if (ptype.length()==0)
                    {
                        ptype= result_e.text();
                    }
                }
                else if( result_e.tagName() == "geometry")
                {
                    QDomNode geometry_n = result_e.firstChild();

                    QDomElement geometry_e = geometry_n.toElement();

                    if(geometry_e.tagName() == "location")
                    {
                        QDomNode location_n = geometry_n.firstChild();

                        while(!location_n.isNull())
                        {
                            QDomElement location_e = location_n.toElement();

                            if(location_e.tagName()== "lat")
                            {
                                plat = location_e.text();
                            }

                            if(location_e.tagName()== "lng")
                            {
                                plng = location_e.text();
                            }
       // Go to the next sibling
                            location_n = location_n.nextSibling();
                        }
                    }
                }
    // Next sibling
                result_n = result_n.nextSibling();
            }
   // Add a place to the list
            iPlacesList.append(new Places(pname, pvicinity,ptype, plat, plng));
        }
        n = n.nextSibling();
    }
    return 0;
}
```

Plotting multiple POIs on Nokia Map

(C) Copyright Nokia 2014. All rights reserved.

MapPlacesDemo plots first 6 places received from Google Places API on a Nokia Map. A HTTP GET request of a map displaying one or more points of interest is done by providing the corresponding geo-coordinates in the request parameters as shown below:

```
void MapPlacesRequestor::getPlacesImage()
{
    QString url("http://m.nokia.me/?poi=");
    for (int i =0; i<iPlacesList.count(); i++)
    {
        if (i==6)
        {
            break;
        }
        url.append(iPlacesList.at(i)->lat);
        url.append(",");
        url.append(iPlacesList.at(i)->lng);
        url.append(",");
    }
    url.append("&h="
               + MAP_HEIGHT + ",&w="
               + MAP_WIDTH + "&nopoim");
    nam->get(QNetworkRequest(QUrl(url)));
}
```

When the map response is received, it is passed to QML UI for displaying. The QML UI for the AroundMe map view is available in the attached source code.

POI Views

The next screen displays a listview with all the places, received from the Google places API. The model is exposed to QML from getPlacesNameList() function.

```
QStringList MapPlacesRequestor::getPlacesNameList(){
    QStringList list;
    for (int i =0; i<iPlacesList.count(); i++)
    {
        list.append(iPlacesList.at(i)->name);
    }
    return list;
}
```

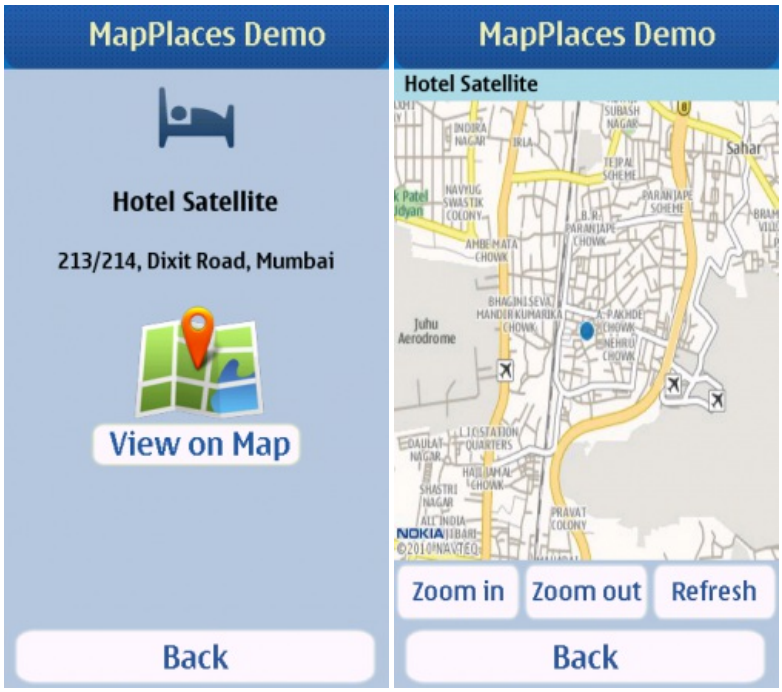
The code for listview component is shown below,

```
// List of places around
ListView{
 id: listview
 width: parent.width
 height: parent.height - backButton_aroundMeListView.height - infoText_AroundMe.height
 clip: true
 anchors.top: infoText_AroundMe.bottom
 model: listmodel
 delegate: Rectangle {
  width: parent.width
  height: 50
  color: "lightsteelblue"
  Image{
```


(C) Copyright Nokia 2014. All rights reserved.

```
    id: icon
    width: 40
    height: 40
    source: MapPlacesRequestor.getIcon(index);
    anchors.left: parent.left
    anchors.leftMargin: 10
   }
   Text {
    id: listItemtext
    text: modelData
    anchors.left: icon.right
    anchors.leftMargin: 10
    font.pixelSize: 20
    color: "darkblue"
   }
   Rectangle{
    width: parent.width
    height: 1
    color: "darkgreen"
    anchors.top: icon.bottom
    anchors.topMargin: 5
   }
   MouseArea{
    anchors.fill: parent
    onClicked: {
     detailView.visible = true;
     selectedIndex = index;
     MapPlacesRequestor.getPoiMap(index);
    }
   }
  }
 }
```

Selecting a place from the list, shows a detail view for the place.



The above view also allows to view that place inidividually on a Nokia map. This is done, as explained in the article MyMapDemo - Using the RESTful Map API with QtQuick. This view also allows the user to zoom in, zoom out and refresh the map as shown in the video/screenshot.

(C) Copyright Nokia 2014. All rights reserved.

Source code

- The complete source for MapPlacesDemo and MyMapDemo is available here and here.

Related Article

- MyMapDemo - Using the RESTful Map API with QtQuick

(C) Copyright Nokia 2014. All rights reserved.