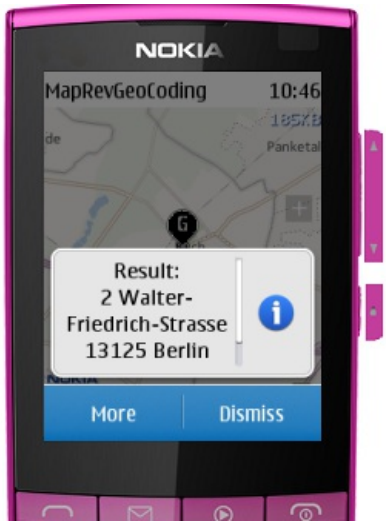
Reverse Geocoding an address with Nokia Maps API for JavaME

This article explains how to use reverse geocoding service with the Map API for Java ME to get address for selected Geo location.

Introduction



The HERE Maps API for Java ME facilitates the easy embedding of the HERE Maps service into Java ME applications. The API includes reverse geocode functionality, which makes it possible to find the address of the given geographical coordinates.

SearchFactory and ReverseGeocodeRequestListener

The SearchFactory class (available in the 'com.nokia.maps.search ' package) offers various functionalities, including geocode, reverse geocode, and search. These operations are performed in an asynchronous manner, for each service there is a listener interface defined for result callback, which is used when the service is used in asynchronous manner. This interface implements methods for informing of the successful completion of the request as well as there is separate method to be called in error situations.

Implementing reverse geo coding midlet

To be able to utilizing the reverse geo service, there need to be class which is implementing the required PlaceSearchRequestListener interface, this interface defines following 2 methods:

- onRequestComplete(ReverseGeocodeRequest request, Location[] result)
- onRequestError(ReverseGeocodeRequest request, java.lang.Throwable error)

And to include this interface in your midlet implementation you need to use the *implements* clause to, basic code that handles this is shown below:

```
public class MapRevGeoCoding extends MIDlet
implements ReverseGeocodeRequestListener, ...
{
    public void onRequestComplete(ReverseGeocodeRequest request, Location[] result){
    ...
    }

    public void onRequestError(ReverseGeocodeRequest request, java.lang.Throwable error)
{
    ...
    }
}
```


(C) Copyright Nokia 2014. All rights reserved.

Starting the query

Before the service can be used you need to get the instance of search factory, which implements the service, this is handled by simply calling the getInstance method from SearchFactory:

```
sm = SearchFactory.getInstance();
```

Then you need to create the ReverseGeocodeRequest instance. This is handled by calling createReverseGeocodeRequest method for the search factory, which after you can use the methods implemented in the reverse geo coding request to handle the actual reverse geo coding request. Note that this request class is implementing multiple functions for making the request, and if you choose you can use the synchronous versions, which would allow you to handle the reply right after the function cal. With this example we are using asynchronous methods, and as we implemented the ReverseGeocodeRequestListener interface we can use the this with with the method call, the other argument used with this method call, is geo the location to be used with reverse geo coding, in this example the center of the map is used as the geo location for the service.

```
ReverseGeocodeRequest revG = sm.createReverseGeocodeRequest();
revG.reverseGeocode(mapCanvas.getMapDisplay().getCenter(),this);
```

Handling the result

Successful completion of the asynchronous reverse geo coding will be notified by calling the onRequestComplete method implemented for the ReverseGeocodeRequestListener interface. The onRequestComplete will be called with ReverseGeocodeRequest argument, as well as Location array holding the results for the request.

```
public void onRequestComplete(ReverseGeocodeRequest request, Location[] result){
 if(result.length > 0){
         Location firstLocation = result[0];
                 Address address = firstLocation.getAddress();
 ...
 }
}
```

The query could also be finished with error, and to handle those you should also implement the onRequestError method, in this example this method simply informs the user of the possible problem.

```
public void onRequestError(ReverseGeocodeRequest request, java.lang.Throwable error){
 Alert alertView = new Alert("Search request error: " + error.getMessage());
 display.setCurrent(alertView);
}
```

Resources

Full source code of the MapRevGeoCoding is available here: File:Java MapRevGeoCoding.zip

Summary

The Map API for Java ME offers rich functionalities that allows to integrate all the main HERE Maps features in a Java ME application, with just a few lines of code.

(C) Copyright Nokia 2014. All rights reserved.

 (C) Copyright Nokia 2014. All rights reserved.