Service API - Fetch and sort nearby landmarks

Landmarks Service API provides access to landmarks stored on the device to widget developers. One can query landmarks by category, name or maximum distance from the given point. Downside is that there is no built in support to sort fetched landmarks by their distance, which is a common use case for LBS applications.

These functions are only available on S60 5th Edition phones.

Luckily Location Service API includes calculate method, which can be used to calculate the distance between two points expressed in WGS-84 format.

Steps to take:

1. Get Current location (hard coded location used in example for convenience)
2. Get landmarks and filter out ones that are too far away
3. Calculate distance between landmark and current position
4. Sort the list by distance / alphabetically

Step one is omitted since it is just hard coded in the example.

Getting the landmarks is a easy task. Landmarks are filtered by adding modifying filter object properties. MaximumDistance specifies how far away the landmaks can be, and landmarkPosition defines the point which is used as a reference.

```
var currentPos = new Object();
currentPos.Longitude = 23.861260972;
currentPos.Latitude  = 61.447976734;

var maxDistance = 100000;

var landmarkSo = device.getServiceObject("Service.Landmarks", "IDataSource");

var criteria = new Object();
criteria.Type = "Landmark";

var filter = new Object();
criteria.Filter = filter;

criteria.Filter.MaximumMatches = 10;
criteria.Filter.MaximumDistance = maxDistance; //meters
criteria.Filter.CoverageRadiusOption = true;
criteria.Filter.LandmarkPosition = currentPos;

var result = landmarkSo.IDataSource.GetList(criteria,landmarksGot);
```

Asynchronous version of the GetList is used so callback function must be defined.

```
function landmarksGot(transactionID, eventCode, result){
switch(eventCode) {
 case 2:
       {
        var list = result.ReturnValue;

 var landmark = null;
 var landmarks= new Array();
 var i=0;

        while ((landmark = list.getNext()) != undefined) {
    var distance = calclulateDistance(currentPos,landmark.LandmarkPosition);
    landmarks[i]= new Array();
    landmarks[i][0] = landmark;
    landmarks[i][1] = distance;
    i++;
 }
 landmarks.sort(sortLandmarks);

 for(lm in landmarks){
  writeLine(landmarks[lm][0].LandmarkName+" dist "+distanceToHumanReadable(landmarks[lm][1]));
 }
       }
....handle the rest of the possible event codes..
```

Now distance must be calculated from the current position to the landmark position. This is done by the help of the Location Service API. Please note the Altitude must be defined although it is not used in the calculation.

```
locationSo = device.getServiceObject("Service.Location", "ILocation");

function calclulateDistance(currentPos, landmarkPos){

 var criteria = new Object();
 criteria.MathRequest = "FindDistance";
 criteria.DistanceParamSource = currentPos;
 criteria.DistanceParamDestination = landmarkPos;

 //altitude must be present altough it does not affect the result
 criteria.DistanceParamSource.Altitude=0;
 criteria.DistanceParamDestination.Altitude=0;

 result = locationSo.ILocation.Calculate(criteria);

 if (result.ErrorCode != 0) {
  writeLine(result.ErrorCode +": "+result.ErrorMessage);
  return -1;
 }
 else {
  return result.ReturnValue;
 }
```


(C) Copyright Nokia 2013. All rights reserved.

```
}
```

The final step is to create a custom sort function to sort the items in the list by distance. First we sort by the distance and then, if distance is equal, sorting is done by the LandmarkName. Sorting is done by calling landmarks.sort(sortLandmarks);

```
function sortLandmarks(a, b) {

if(a[1]< b[1])
 return -1;

if(a[1] > b[1])
 return 1;

//a[1] == b[1] sort alphabetically
var nameA=a[0].LandmarkName.toLowerCase();
var nameB=b[0].LandmarkName.toLowerCase();

if(nameA < nameB)
 return -1;

if(nameA > nameB)
 return 1;

return 0;
}
```

The last bit is to play around with number formatting. 56.2km looks better than 56200m.

```
function distanceToHumanReadable(distance){
var dipDist="";

if (distance > 1000) {
 distance = Math.round(distance / 100)/10;
 dipDist = distance  + " km";
}
else {
 dipDist = Math.round(distance) + " m";
}
return dipDist;
}
```


(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.