



Interactive Google Maps in Oracle Application Express

Jeffrey Kemp

Principal Software Engineer, Oracle

AUSOUG Webinar Series, 17 February 2021



JavaScript required...

Which of the following do NOT require you to write a single line of JavaScript?

1. Show pins on the map based on a **SQL query**
2. When a pin is clicked, have an **Info Window** pop up
3. Show **driving directions** between a number of points on the map
4. Apply the **Marker Clustering** javascript library
5. Apply the **Spiderfier** javascript library
6. Show the **Satellite** or **Terrain** map types
7. Change the **Zoom** level
8. Show or hide any of the Google map **controls**
9. Load a large number of pins in **batches** using AJAX
10. Show a geo **heatmap**
11. Search the map by **address**
12. Get the user's **device location** and zoom to it
13. Allow the user to drag-and-drop a **geoJSON** document onto the map



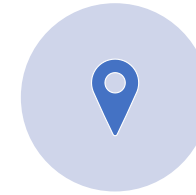
Today's Demo



Install



Searchable Map



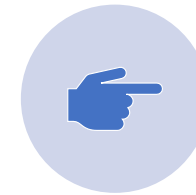
Map Styles



Integrate Report



Custom Icons



Directions



Device Location



Q & A

1. Install JK64 ReportMap Plugin

<https://github.com/jeffreykemp/jk64-plugin-reportmap>

INSTALLATION INSTRUCTIONS

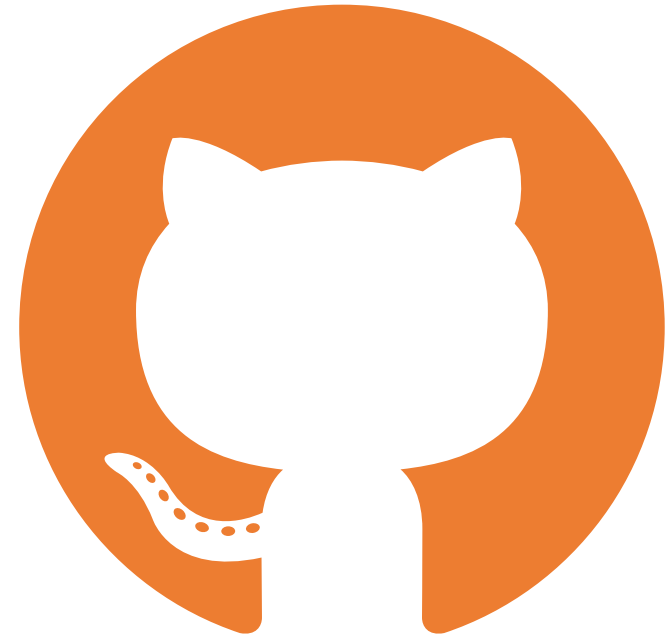
NOTE: if you are upgrading from any version prior to 1.0 (i.e. 0.x), some changes will be required to your application to support it as a number of attributes have been removed or changed.

1. Download the [latest release](#)
2. In your application, go to **Shared Components** -> **Plug-ins** and click **Import**
3. Import the region plugin: `region_type_plugin_com_jk64_report_google_map_r1.sql`
4. Supply your public **Google API Key** (Component Settings)
5. Add a region to the page, select type **JK64 Report Google Map R1 [Plug-In]**
6. For **SQL Source**, enter a query with at least 4 columns, for example:

```
SELECT lat, lng, name, id FROM mydata
```

(if you just want a map with no data, enter a dummy query (e.g. `select 1 from dual`); then in the map region properties, set **Source** -> **Location** to `- Select -`)
7. (OPTIONAL) To add dynamic action features, import the companion Dynamic Action plugins:
`dynamic_action_plugin_com_jk64_report_google_map_da_r1.sql`
`dynamic_action_plugin_com_jk64_report_google_map_directions_da.sql`

IF YOU ARE UPGRADING from any prior release, refer to the [Upgrade Notes](#).



Terminology

Term	Definition
Region	an APEX region, e.g. a form, report, or plugin region
ReportMap Region	a region based on the ReportMap plugin
Visualisation	defines how data is to be represented in the map
Dynamic Action	defines some client-side behavior in response to an event
Event	a trigger that causes a Dynamic Action to fire
Map Event	an event that the ReportMap plugin can raise
Action	what to do when a Dynamic Action has fired
ReportMap Action	an action that does something on a map region



2. Create a searchable map

1. Add a map region
2. Set the SQL Query, e.g. `select lat, lng, name, id from...`
3. Set Static ID
4. Add items P1_SEARCH, P1_ID, P1_INFO
5. Dynamic Action on change of P1_SEARCH, set P1_ID using PL/SQL
6. Dynamic Action on change of P1_ID, call **click** API
`$("#map_mymap").reportmap("click", $v("P1_ID"));`
7. Modify SQL Query to add filter on :P1_ID
8. Dynamic Action on map event **markerClick**, set P1_INFO to `this.data.name`



3. Style the map

- Visualisation: **Spiderfier**
- Visualisation: **Marker Clustering**
- Set **Map Style**



4. Integrate report with map

- Add an APEX report to the page
- Add link, set Target to URL `javascript:showHospital(#ID#)`
- Function and Global Variable Declaration:

```
function showHospital(id) {  
    $("#map_mymap").reportmap("click",id);  
}
```
- Add Hidden item P2_ID
- Dynamic Action on map **markerClick**, set value of P2_ID to JavaScript Expression `this.data.id`
- Dynamic Action on change of P2_ID, refresh report region
- Add filter on report based on P2_ID



5. Show custom icons

- Method 1: get icon path/filename from query

- SQL Query

```
select lat, lng, name, id, info,  
       'https://domain/path/file.png' icon  
from ...
```

- Method 2: set **iconBasePath**

- JavaScript Initialization Code:

```
this.options.iconBasePath = 'https://domain/path/';
```

- SQL Query

```
select lat, lng, name, id, info,  
       'file.png' icon  
from ...
```



<https://github.com/jeffreykemp/jk64-plugin-reportmap/wiki/Plugin-Attributes-Reference>

<https://github.com/jeffreykemp/jk64-plugin-reportmap/wiki/Map-Icons>

5. Show custom icons (cont'd.)

- Method 3: use **markerFormatFn**

- JavaScript Initialization Code:

```
this.options.markerFormatFn =  
  function(marker) {  
    var iconUrl = 'https://maps.google.com/mapfiles/ms/icons/'  
      + ((marker.data.id % 3 == 0)  
        ? 'info.png'  
        : 'hospitals.png');  
    marker.icon = {  
      url: iconUrl  
    };  
  };
```



<https://developers.google.com/maps/documentation/javascript/markers>

https://apex.oracle.com/pls/apex/apex_pm/r/ut/icons

5. Show custom icons (cont'd.)

- Method 4: **Custom HTML**
 - Visualisation = Info Layer
 - SQL Query:

```
select lat, lng, name, id,  
       '<i class="fa fa-h-square"/>'  
       as info  
from...
```
 - May be formatted with custom JavaScript by setting `this.options.infoLayerFormatFn`
 - Make clickable by setting `layer.options.onClickHandler`



6. Show directions

- Set Visualisation to Directions
- Items P3_DISTANCE, P3_DURATION
- Dynamic Action on map event **directions**,
execute Javascript:

```
$s("P3_DISTANCE", (this.data.distance/1000) + "km");  
$s("P3_DURATION", (this.data.duration/3600) + "hr");
```



<https://github.com/jeffreykemp/jk64-plugin-reportmap/wiki/Plugin-Events-Reference#directions>

<https://github.com/jeffreykemp/jk64-plugin-reportmap/wiki/Plugin-API-Reference#showdirections>

7. Show device location

- Create button defined by Dynamic Action
- On click, run plugin **JK64 Report Google Map R1 Action**
 - Action = **Go to Device Location**
 - Selection Type = the map region
- Dynamic Action on map event **geolocate**
 - Set item to JavaScript Expression
`this.data.lat+", "+this.data.lng`



More demo

https://apex.oracle.com/pls/apex/jk64/r/jk64_report_map

The screenshot displays the 'JK64 Report Map Plugin' interface. At the top, there is a header with the title 'jk64 Report Map Plugin Demo' and a link to 'Download the source from GitHub'. Below the header is a navigation sidebar with icons for home, eye, location, checkmark, settings, edit, copy, and warning. The main content area features a map of the world in satellite view, with 'Map' and 'Satellite' tabs at the top left. The map includes a search bar, a person icon, and zoom controls. Below the map is a grid of six feature cards, each with an icon and a description:

- Report Pins**: Show a set of data as pins on the map.
- Marker Clustering**: When many pins are very close together, dynamically cluster them into a single marker showing the number of pins within it.
- Route Map**: Show route directions between two points with up to 8 waypoints on the way.
- Directions**: (Icon: two arrows forming a square)
- Geo Heatmap**: (Icon: globe)
- Spiderfier**: (Icon: asterisk)



Q & A

<https://jeffkemponoracle.com>

