HERE Maps API - How to add map markers

Introduction

In this article we will go through how to add one or more `Markers` to the map. A `Marker` defines an `Object` that may display an icon on the map view.

Prerequisites

A supported web browser (basically any modern web browser)

The example assumes you have already added the HERE Maps to your web page as explained in the previous article HERE Maps API - Add Maps To Any Web Page

Important note about maps credentials

Nokia provides several services options within the Maps API offering. The service is free to use, but you must obtain and use authentication and authorization credentials to use the services. Please read the Terms and Conditions and check the Pricing Plans page to decide which business model best fits your needs. Authentication requires unique Maps API credentials, namely an AppId and a token. You can get these credentials free for free following the instructions here

Implementation

Defining a Marker

First we will need to define the `Marker`. This will be done by specifying the location as a `geo.Coordinate` and then defining the `Marker`:

```
var center =    new nokia.maps.geo.Coordinate(40.738728,-73.99236);
var marker = new nokia.maps.map.StandardMarker(center);
```

The `Marker` needs geocoordinates as parameters, that is, its latitude and longitude on the map. In our example we are using 40.738728 for the latitude and -73.99236 for the longitude This will place a marker in Central New York. `marker` now contains a new `StandardMarker` object.

Adding a Marker directly

We may place the `StandardMarker` on the map by adding it directly as shown:

```
map.objects.add(marker);
```

Adding a Marker via a Container

As an alternative to adding a marker directly, we can add it to a `Container` and then add the `Container` to the map. A `Container` is used to group `Objects`, and thus allows them all to be operated on at once without the need to iteratively visit each one (e.g. for changing the visibility).

First create a `Container`:

```
var container = new nokia.maps.map.Container();
```

Now that we have the `Container` ready, we can add the `StandardMarker` into it:

(C) Copyright Nokia 2014. All rights reserved.

```
    container .objects.add(marker);
    //  Alternatively:
    //   container.objects.add(
    //      new nokia.maps.map.StandardMarker(
    //         new nokia.maps.geo.Coordinate(40.738728,-73.99236))
    //    );
```

And then when the `StandardMarker` is in the `Container`, we will need to add the `Container` to the map objects collection, to be able to show the content on it:

```
    map.objects.add(container);
```

Manipulating the Marker

The properties of the `StandardMarker` can be further manipulated, e.g. you can make the `Marker` `draggable` (it is static by default) if you wish:

```
    marker.set("draggable",true);
```

And to make the `StandardMarker` static again, you would give:

```
    marker.set("draggable",false);
```

For a `StandardMarker` you can set the `text` property as shown:

```
    marker.set("text","Your marker!")
```

Retrieving information about the marker

Sometimes it may be useful to know if a `Container` has been added to the `Display`. Earlier we could have initiated the `Display` object e.g. like this:

```
...
    var map = nokia.maps.map.Display(document.getElementById("map"),
...
```

Now, with the reference to `"map"`, we can request:

```
    test = marker.isAdded (map)
```

Our case test would result in `true`.

With the `isVisible (display) : Boolean` we will get true if the object's visibility as well as all parent's visibilities are true and if the object is attached to a map (i.e. parent chain ends with a map). The returned value is only valid for the given `nokia.maps.map.Display` object.

Parameters:

{nokia.maps.map.Display} display The display to check.

In our case we could ask:

```
    test = marker.isVisible(map);
```


(C) Copyright Nokia 2014. All rights reserved.

Again, this would result in “true”.

With the `getParent (display)` we can get the parent of the map object.

Parameters:

{nokia.maps.map.Display} display The display for which to return the parent object.

Returns: {nokia.maps.map.Object} The `parent` object of this object for the given display or `undefined` if this map object has no parent.

```
test = marker.getParent(map);
```

If we want to get the `Display` names that the `Container` is attached to, this could be queried as follows:

```
test = marker.getDisplays();
```

In this case we would know already that there is the `"map" Display`, so now asking the question `marker.isAdded (test)` would give us `true`, because `marker` is attached to the `"map"`.

Example code

A working example can be found at:

http://rawgithub.com/heremaps/examples/master/maps_api_for_javascript/simple-examples/simple-marker.html

For more on the HERE Maps API

Please check out the HERE Maps API full documentation and API reference here:

- HERE Maps API

You may also access the interactive API explorer

- API explorer

And additional examples on GitHub

- HERE Maps on Github

(C) Copyright Nokia 2014. All rights reserved.