

InfoPortugal MAPS Core

Author: Bruno Monteiro
Version: 1.1
Copyright: InfoPortugal S.A.
Date: 19 Mar 2013



Contents

1	Introduction	5
2	Basic usage	5
3	IP.Maps	6
3.1	IP.Maps.setImagesLocation	6
3.2	IP.Maps.getImagesLocation	6
3.3	IP.Maps.setTheme	6
3.4	IP.Maps.fromMercatorToDegrees	6
3.5	IP.Maps.fromDegreesToMercator	6
3.6	IP.Maps.lonLatFromMercatorToDegrees	6
3.7	IP.Maps.lonLatFromDegreesToMercator	7
3.8	IP.Maps.lonLatFromString	7
3.9	IP.Maps.requestJSONP	7
4	IP.Maps.Map	8
4.1	IP.Maps.Map	8
4.2	IP.Maps.Map.init	8
4.3	IP.Maps.Map.addDefaultAerialLayer	8
4.4	IP.Maps.Map.addDefaultTrafficLayer	8
4.5	IP.Maps.Map.setBaseLayer	9
4.6	IP.Maps.Map.setOption	9
4.7	IP.Maps.Map.addControl	9
4.8	IP.Maps.Map.addLayer	9
4.9	IP.Maps.Map.removeLayer	9
4.10	IP.Maps.Map.addMarkers	9
4.11	IP.Maps.Map.zoomToExtent	10
4.12	IP.Maps.Map.zoomToMaxExtent	10
4.13	IP.Maps.Map.setCenter	10
4.14	IP.Maps.Map.getCenter	10
4.15	IP.Maps.Map.getCenterArray	10
4.16	IP.Maps.Map.getZoom	10
4.17	IP.Maps.Map.getScale	10
4.18	IP.Maps.Map.panToLonLat	11
4.19	IP.Maps.Map.panTo	11
4.20	IP.Maps.Map.zoomTo	11
4.21	IP.Maps.Map.zoomIn	11
4.22	IP.Maps.Map.zoomOut	11

4.23	IP.Maps.Map.getExtent	11
4.24	IP.Maps.Map.getLonLatFromPixel	11
4.25	IP.Maps.Map.getLayerPxFromLonLat	12
4.26	IP.Maps.Map.getPixelFromLonLat	12
4.27	IP.Maps.Map.getViewPortPxFromLonLat	12
4.28	IP.Maps.Map.calculateBuffer	12
4.29	IP.Maps.Map.bufferContainsExtent	12
4.30	IP.Maps.Markers.activateDragging	12
4.31	IP.Maps.Map.Markers.deactivateDragging	13
4.32	IP.Maps.Map.isDraggingActivated	13
5	IP.Maps.Markers	14
5.1	IP.Maps.Markers	14
5.2	IP.Maps.Markers.destroy	14
5.3	IP.Maps.Markers.addMarker	14
5.4	IP.Maps.Markers.addMarkers	14
5.5	IP.Maps.Markers.removeMarker	14
5.6	IP.Maps.Markers.removeMarkers	14
5.7	IP.Maps.Markers.getMarkers	15
5.8	IP.Maps.Markers.showMarkerPopup	15
5.9	IP.Maps.Markers.hideMarkerPopup	15
5.10	IP.Maps.Markers.toggleMarkerPopup	15
5.11	IP.Maps.Markers.isDraggingActivated	15
5.12	IP.Maps.Markers.activateDragging	15
5.13	IP.Maps.Markers.deactivateDragging	15
5.14	IP.Maps.Markers.hide	16
5.15	IP.Maps.Markers.show	16
6	IP.Maps.Marker	17
6.1	IP.Maps.Marker	17
6.2	IP.Maps.Marker.destroy	17
6.3	IP.Maps.Marker.setPopupContent	17
6.4	IP.Maps.Marker.showPopup	17
6.5	IP.Maps.Marker.hidePopup	17
6.6	IP.Maps.Marker.togglePopup	17
6.7	IP.Maps.Marker.getLonLat	17
6.8	IP.Maps.Marker.move	18
7	IP.Maps.Geocoding	19

7.1	IP.Maps.Geocoding	19
7.2	IP.Maps.Geocoding.load	19
8	IP.Maps.Routing	20
8.1	IP.Maps.Routing	20
8.2	IP.Maps.Routing.setFrom	20
8.3	IP.Maps.Routing.setTo	20
8.4	IP.Maps.Routing.addStop	20
8.5	IP.Maps.Routing.load	21
8.6	IP.Maps.Routing.draw	21
8.7	IP.Maps.Routing.clear	21
8.8	IP.Maps.Routing.drawManeuversAsUL	21
8.9	IP.Maps.Routing.getTotalDuration	21
8.10	IP.Maps.Routing.getTotalDistance	21
8.11	IP.Maps.Routing.getManeuvers	21
8.12	IP.Maps.Routing.addManeuverMarker	22
8.13	IP.Maps.Routing.showManeuver	22
8.14	IP.Maps.Routing.hideManeuver	22
9	IP.Maps.Pois	23
9.1	IP.Maps.Pois	23
9.2	IP.Maps.Geocoding.search	23
9.3	IP.Maps.Geocoding.detail	23
9.4	IP.Maps.Geocoding.getCategories	24
9.5	IP.Maps.Geocoding.getSubcategories	24
10	IP.Maps.StaticMarkers	25
10.1	IP.Maps.StaticMarkers	25

1 Introduction

The IP.Maps uses the [OpenLayers](https://openlayers.org/) framework.

API: <http://dev.openlayers.org/releases/OpenLayers-2.8/doc/apidocs/files/OpenLayers-js.html>

2 Basic usage

Example

```
var map = new IP.Maps.Map("map", ["http://map1.infoportugal.info/"]);
map.init(-8.088044, 39.640607, 0); // Lon, Lat, Zoom(0-11)
```

Instantiation

```
var map = IP.Maps.Map(<html_element_id>, <url>);
```

Setting options

```
map.setOptions(<option>, <value>);
```

Adding controls

```
map.addControl(<OpenLayers.Control>);
```

Initialization

```
map.init(<Lon>, <Lat>, <zoom 0-11>);
```

3 IP.Maps

3.1 IP.Maps.setImagesLocation

Description:

Static method. Sets the image URL location.

Required arguments:

url <String>: URL location.

3.2 IP.Maps.getImagesLocation

Description:

Static method. Gets the image URL location

Returns:

The image URL location

3.3 IP.Maps.setTheme

Description

Static method. Sets the stylesheet theme.

Required arguments:

url <String>: URL for the stylesheets.

3.4 IP.Maps.fromMercatorToDegrees

Description:

Static method. Lon/Lat object conversion from Mercator to Degrees.

Required arguments:

lonLat <Openlayers.LonLat>: In Google Mercator.

Returns:

OpenLayers.LonLat object in Degrees.

3.5 IP.Maps.fromDegreesToMercator

Description:

Static method. Lat/Lon object conversion from Degrees to Mercator.

Arguments:

lonLat <Openlayers.LonLat>: In Degrees.

Returns:

OpenLayers.LonLat object in Google Mercator.

3.6 IP.Maps.lonLatFromMercatorToDegrees

Description:

3.7 IP.Maps.lonLatFromDegreesToMercator

Static method. Lat/Lon object conversion from Google Mercator to Degrees.

Required arguments:

lon <Float>: Longitude.

lat <Float>: Latitude.

Returns:

OpenLayers.LonLat object in Degrees.

3.7 IP.Maps.lonLatFromDegreesToMercator

Description:

Static method. Lat/Lon object conversion from Degrees to Google Mercator.

Required arguments:

lon <Float>: Longitude.

lat <Float>: Latitude.

Returns:

OpenLayers.LonLat object in Google Mercator.

3.8 IP.Maps.lonLatFromString

Description:

Static method. Creates a OpenLayers.LonLat from a string in the format <lon>,<lat>.

Required arguments:

lonLat <String>: String in the <lon>,<lat> format.

Returns:

OpenLayers.LonLat object in Degrees.

3.9 IP.Maps.requestJSONP

Description:

Static method.* Creates a JSONP request object.

Required arguments:

url <String>: URL.

elementId <String>: HTML element id.

4 IP.Maps.Map

4.1 IP.Maps.Map

Description:

Constructor - Creates an IP.Maps.Map instance.

Required arguments:

elementId <String>: HTML element id.

url <String>: URL or an array of the webmapping services.

4.2 IP.Maps.Map.init

Description:

Map initialization

Required arguments:

lon <Float>: Longitude.

lat <Float>: Latitude.

zoom <Integer>: Zoom (0..11).

4.3 IP.Maps.Map.addDefaultAerialLayer

Description:

Adds the default Satellite layer to the list of available IP.Maps layers

Optional arguments:

url <String>: URL or an array of the webmapping services (defaults to Map ones)

label <String>: A name with which to identify the new layer (defaults to "Aerial")

isBaseLayer <Bool>: Setups the new layer as a Base Layer (defaults to false)

4.4 IP.Maps.Map.addDefaultTrafficLayer

Description:

Adds the default layer with traffic information to the list of available IP.Maps layers

Optional arguments:

url <String>: URL or an array of the webmapping services (defaults to Map ones)

label <String>: A name with which to identify the new layer (defaults to "Traffic")

isBaseLayer <Bool>: Setups the new layer as a Base Layer (defaults to false)

IP.Maps.Map.addDefaultHybridLayer

Description:

Adds the default Hybrid Layer to the list of available IP.Maps Layers

Optional arguments:

url <String>: URL or an array of the webmapping services (defaults to Map ones)

label <String>: A name with which to identify the new layer (defaults to "Hybrid")

4.5 IP.Maps.Map.setBaseLayer

visibility <Bool>: Setups the new layer so it's visible by default (defaults to false)

4.5 IP.Maps.Map.setBaseLayer

Description:

Sets the base layer.

Required arguments:

layer <OpenLayers.Layer>: A OpenLayers.Layer object.

4.6 IP.Maps.Map.setOption

Description:

Sets an option on the map

Required arguments:

key <String>: Key.

value <Misc>: Value.

4.7 IP.Maps.Map.addControl

Description:

Adds a control to the map

Required arguments:

control <OpenLayers.Control>: A OpenLayers.Control object.

4.8 IP.Maps.Map.addLayer

Description:

Adds a layer to the map.

Required arguments:

name <String>: The layer name.

layer <OpenLayers.Layer>: A OpenLayers.Layer object.

4.9 IP.Maps.Map.removeLayer

Description:

Removes a layer to the map.

Required arguments:

layer <OpenLayers.Layer>: A OpenLayers.Layer object.

4.10 IP.Maps.Map.addMarkers

Description:

Adds object or an array of IP.Maps.Markers to the map.

Required arguments:

layers <IP.Maps.Markers>: Object or an array of IP.Maps.Markers objects.

4.11 IP.Maps.Map.zoomToExtent

Description:

OpenLayers alias for zoomToExtent().

Required arguments:

bounds <OpenLayers.Bounds>: OpenLayers.Bounds object.

4.12 IP.Maps.Map.zoomToMaxExtent

Description:

OpenLayers alias for zoomToMaxExtent().

4.13 IP.Maps.Map.setCenter

Description:

Centers the map

Required arguments:

lon <Float>: longitude.

lat <Float>: latitude.

zoom <Integer>: zoom (0..11).

4.14 IP.Maps.Map.getCenter

Description:

Gets the lon/lat from the center of the map.

Returns:

OpenLayers.LonLat object.

4.15 IP.Maps.Map.getCenterArray

Description:

Gets the Lon/Lat from the center of the map in an array object.

Returns:

Array object.

4.16 IP.Maps.Map.getZoom

Description:

Gets the current zoom of the map.

Returns:

Integer 0..11.

4.17 IP.Maps.Map.getScale

Description:

Gets the current scale of the map.

Returns:

The scale of the current scale

4.18 IP.Maps.Map.panToLonLat

Description:

Pan the map to a lon/lat.

Required arguments:

lon <Float>: longitude.

lat <Float>: latitude.

4.19 IP.Maps.Map.panTo

Description:

Pan the map to a lon/lat.

Required arguments:

lonLat <OpenLayers.LonLat>: OpenLayers.LonLat object.

4.20 IP.Maps.Map.zoomTo

Description:

Zooms to level

Required arguments:

zoom <Integer>: Zoom 0..11.

4.21 IP.Maps.Map.zoomIn

Description:

Zooms in the map.

4.22 IP.Maps.Map.zoomOut

Description:

Zooms out the map.

4.23 IP.Maps.Map.getExtent

Description:

Gets the map extent.

Returns:

OpenLayers.Bounds object which represents the lon/lat bounds of the current viewPort.

4.24 IP.Maps.Map.getLonLatFromPixel

Description:

4.25 IP.Maps.Map.getLayerPxFromLonLat

Gets a OpenLayers.LonLat object from a pixel.

Returns:

OpenLayers.LonLat object.

4.25 IP.Maps.Map.getLayerPxFromLonLat

Description:

Gets a OpenLayers.Layer object from a pixel.

Returns:

OpenLayers.LonLat object.

4.26 IP.Maps.Map.getPixelFromLonLat

Description:

Gets a pixel location given a map location. The map location is translated to an integer pixel location (in viewport pixel coordinates) by the current base layer.

Returns:

OpenLayers.Pixel corresponding to the OpenLayers.LonLat translated into view port pixels by the current base layer.

4.27 IP.Maps.Map.getViewPortPxFromLonLat

Description:

Gets a OpenLayers.Pixel ViewPort from a OpenLayers.LonLat.

Returns:

OpenLayers.Pixel Layer Pixel translated into ViewPort Pixel coordinates.

4.28 IP.Maps.Map.calculateBuffer

Description:

Calculates the current buffer.

Returns:

Array object.

4.29 IP.Maps.Map.bufferContainsExtent

Description:

Returns true if the current buffer contains the current extent.

Returns:

Boolean if the current buffer contains the current extent

4.30 IP.Maps.Markers.activateDragging

Description:

Activate dragging.

4.31 IP.Maps.Map.Markers.deactivateDragging

Required arguments:

object <OpenLayers.Layer> or <Object>: OpenLayers.Layer or Object the has a layer attribute.

4.31 IP.Maps.Map.Markers.deactivateDragging

Description:

Deactivate dragging.

4.32 IP.Maps.Map.isDraggingActivated

Description:

Returns true if the dragging is activated

Returns:

Boolean if the dragging is activated

5 IP.Maps.Markers

5.1 IP.Maps.Markers

Description:

Constructor - Creates a markers container.

Required arguments:

name <String>: The layer name.

Optional arguments:

style <OpenLayers.StyleMap>: The OpenLayers.StyleMap for the icons.

5.2 IP.Maps.Markers.destroy

Description:

Removes the markers, removes layer from map and destroy the layer

5.3 IP.Maps.Markers.addMarker

Description:

Adds a IP.Maps.Marker to the IP.Maps.Markers layer.

Required arguments:

marker <IP.Maps.Marker>: A IP.Maps.Marker object.

5.4 IP.Maps.Markers.addMarkers

Description:

Adds an array of IP.Maps.Marker to the IP.Maps.Markers.

Required arguments:

markers <Array>: Array of IP.Maps.Marker.

5.5 IP.Maps.Markers.removeMarker

Description:

Removes a IP.Maps.Marker from the IP.Maps.Markers layer.

Required arguments:

marker <IP.Maps.Marker>: A IP.Maps.Marker object.

5.6 IP.Maps.Markers.removeMarkers

Description:

Removes an array of IP.Maps.Marker from the IP.Maps.Markers.

Optional arguments:

markers <Array>: Array of IP.Maps.Marker. Without arguments, all markers are removed.

5.7 IP.Maps.Markers.getMarkers

Description:

Gets the IP.Maps.Marker objects associated.

Returns:

Array of IP.Maps.Marker.

5.8 IP.Maps.Markers.showMarkerPopup

Description:

Shows the marker popup.

Required arguments:

IP.Maps.Marker: A IP.Maps.Marker object.

5.9 IP.Maps.Markers.hideMarkerPopup

Description:

Hides the marker popup.

Required arguments:

marker <IP.Maps.Marker>: A IP.Maps.Marker object.

5.10 IP.Maps.Markers.toggleMarkerPopup

Description:

Toggles the visibility of marker popup.

Required arguments:

marker <IP.Maps.Marker>: A IP.Maps.Marker object.

5.11 IP.Maps.Markers.isDraggingActivated

Description:

Returns true if dragging is activated.

Returns:

Returns true if dragging is activated.

5.12 IP.Maps.Markers.activateDragging

Description:

Activate dragging.

5.13 IP.Maps.Markers.deactivateDragging

Description:

Deactivate dragging.

5.14 IP.Maps.Markers.hide

Description:

Hides markers.

5.15 IP.Maps.Markers.show

Description:

Show markers.

6 IP.Maps.Marker

6.1 IP.Maps.Marker

Description:

Constructor - Creates a Popup Marker.

Required arguments:

lon <Float>: longitude.

lat <Float>: latitude.

Optional arguments:

popupContent <String>: Popup content.

popupClass <OpenLayers.Popup>: Popup class.

showClose <Boolean>: Show close true/false.

allowOverflow <Boolean>: Allow overflow true/false.

6.2 IP.Maps.Marker.destroy

Description:

Removes the marker from the parent(IP.Maps.Markers), destroy popup and destroy the layer.

6.3 IP.Maps.Marker.setPopupContent

Description:

Sets the popup content.

Required arguments:

popupContent <String>: Popup content.

6.4 IP.Maps.Marker.showPopup

Description:

Shows the marker popup.

6.5 IP.Maps.Marker.hidePopup

Description:

Hides the marker popup.

6.6 IP.Maps.Marker.togglePopup

Description:

Toggles the visibility of the marker popup.

6.7 IP.Maps.Marker.getLonLat

Description:

Returns a OpenLayers.LonLat object.

Returns:

OpenLayers.LonLat object.

6.8 IP.Maps.Marker.move

Description:

Moves marker to lon, lat.

Required arguments:

lon <Float>: longitude.

lat <Float>: latitude.

7 IP.Maps.Geocoding

7.1 IP.Maps.Geocoding

Description:

Constructor - Initialize Geocoding.

Required arguments:

url <String>: API URL.

api_key <String>: API Key.

7.2 IP.Maps.Geocoding.load

Description:

Loads data via JSONP.

Optional arguments:

options ``<Object>``: Object with:

address <String>: Address for search.

number <Integer>: Door number.

postal_code_4 <String>: Postal code 4 digits in DDDD format.

postal_code_3 <String>: Postal code 3 digits in DDD format.

locality <String>: Locality.

beforeSend <Callback>: Callback for before send.

onComplete <Callback>: Callback for on complete.

8 IP.Maps.Routing

8.1 IP.Maps.Routing

Description:

Constructor - Initialize Routing.

Required arguments:

url <String>: API URL.

api_key <String>: API Key.

map: <IP.Maps.Map>: IP.Maps.Map object.

Optional arguments:

icons_path: <String>: Icons path.

icons_size: <OpenLayers.Size>: Icons size.

line_style: <Object>: Line layer style.

maneuver_zoom_level: <Integer>: Maneuver zoom level.

8.2 IP.Maps.Routing.setFrom

Description:

Sets lon/lat for initial location.

Required arguments:

lon <Float>: Longitude.

lat <Float>: Latitude.

display: <Boolean>: Boolean for displaying the marker.

8.3 IP.Maps.Routing.setTo

Description:

Sets lon/lat for destination location.

Required arguments:

lon <Float>: Longitude.

lat <Float>: Latitude.

display: <Boolean>: Boolean for displaying the marker.

8.4 IP.Maps.Routing.addStop

Description:

Sets lon/lat for stop.

Required arguments:

lon <Float>: Longitude.

lat <Float>: Latitude.

8.5 IP.Maps.Routing.load

Description:

Loads data via JSONP.

Optional arguments:

options ``<Object>``: Object with:

type: <Integer>: Routing type 0: fastest, 1: shortest, 2: by foot.

tolls: <Integer>: Enable tolls 0: false, 1: true.

beforeSend: <Callback> Callback for before send.

onComplete: <Callback> Callback for on complete.

8.6 IP.Maps.Routing.draw

Description:

Draws routing on map.

Optional arguments:

zoomToExtend: <Boolean>: Zoom to extent.

8.7 IP.Maps.Routing.clear

Description:

Clears routing on map.

8.8 IP.Maps.Routing.drawManeuversAsUL

Description:

Draws maneuvers on a DOM element as UL.

Required arguments:

elemend_id: <String>: HTML element ID.

8.9 IP.Maps.Routing.getTotalDuration

Description:

Returns the total duration in seconds.

8.10 IP.Maps.Routing.getTotalDistance

Description:

Returns the total distance in Km.

8.11 IP.Maps.Routing.getManeuvers

Description:

Returns the maneuvers in GeoJSON format.

8.12 IP.Maps.Routing.addManeuverMarker

Description:

Adds a maneuver marker.

Required arguments:

index: <Integer>: Maneuver index.

type: <String>: Maneuver type.

lonLat: <OpenLayers.LonLat>: Maneuver lon/lat.

8.13 IP.Maps.Routing.showManeuver

Description:

Shows a maneuver on map.

Required arguments:

index: <Integer>: Maneuver index.

8.14 IP.Maps.Routing.hideManeuver

Description:

Hides a maneuver on map.

Required arguments:

index: <Integer>: Maneuver index.

9 IP.Maps.Pois

9.1 IP.Maps.Pois

Description:

Constructor - Initialize Pois.

Required arguments:

url <String>: API URL.

api_key <String>: API Key.

9.2 IP.Maps.Geocoding.search

Description:

Search point of interest.

Optional arguments:

options ``<Object>``: Object with:

lang: <String>: Language pt or en

name <String>: Name of POI for search.

category <Integer>: Category ID.

subcategory <Integer>: Subcategory ID.

bbox <OpenLayers.Bounds>: Bounding box for searching format.

lon <Float>: Longitude.

lat <Float>: Latitude.

distance <Integer>: Distance in meters from point for searching.

beforeSend <Callback>: Callback for before send.

onComplete <Callback>: Callback for on complete.

Important

The request must provide the lon, lat and distance or the bbox or the name.

9.3 IP.Maps.Geocoding.detail

Description:

Point of interest detail.

Required arguments:

id: <String>: Point of interest ID.

Optional arguments:

options ``<Object>``: Object with:

9.4 IP.Maps.Geocoding.getCategories

lang: <String>: Language pt or en

beforeSend <Callback>: Callback for before send.

onComplete <Callback>: Callback for on complete.

9.4 IP.Maps.Geocoding.getCategories

Description:

Returns the categories available.

9.5 IP.Maps.Geocoding.getSubcategories

Description:

Returns the subcategories available for a category.

Required arguments:

category: <Integer>: Category ID.

Optional arguments:

options ``<Object>``: Object with:

lang: <String>: Language pt or en

beforeSend <Callback>: Callback for before send.

onComplete <Callback>: Callback for on complete.

10 IP.Maps.StaticMarkers

10.1 IP.Maps.StaticMarkers

Description:

Constructor - Initialize Static Markers.

Required arguments:

url <String>: API URL.

apiKey <String>: API Key.

map: <IP.Maps.Map>: IP.Maps.Map object.

Optional arguments:

zoomThreshold: <Integer>: Threshold zoom level where the markers appear.

style <OpenLayers.StyleMap>: The OpenLayers.StyleMap for the icons.

markersLabel <String>: Label for the Markers object.

getMarkerFromFeature <Callback>: Callback that creates a Marker from a feature.

Copyright © 2010, InfoPortugal