SiteWatch API

Version 2.0

Samsyn ehf.

Table of Contents

Part I Copyright 6

Part II SiteWatch API 8

Part III SiteWatch_API Class 10

1 Projection ... 10

ISN93_To_WGS84 ... 10

WGS84_To_ISN93 ... 10

Part IV SWPoint Class 13

1 create ... 13

2 Static Methods ... 13

scale ... 13

subtract ... 14

sum ... 14

Part V SWMap Class 16

1 Bundled Images ... 16

2 Types of Listeners ... 16

3 create ... 18

4 Map Object ... 20

CustomCanvas Object ... 20

forceRedraw ... 20

getConfig ... 20

getId ... 20

getType ... 21

hide ... 21

isHidden ... 21

isTainted ... 21

remove ... 21

setDrawHandler ... 22

show ... 22

updateConfig ... 22

xToWinX ... 23

yToWinY ... 23

DivMarker Object ... 23

getConfig ... 23

getDom ... 23

getId ... 24

getPoint ... 24

getType ... 24

hide ... 24

isHidden ... 25

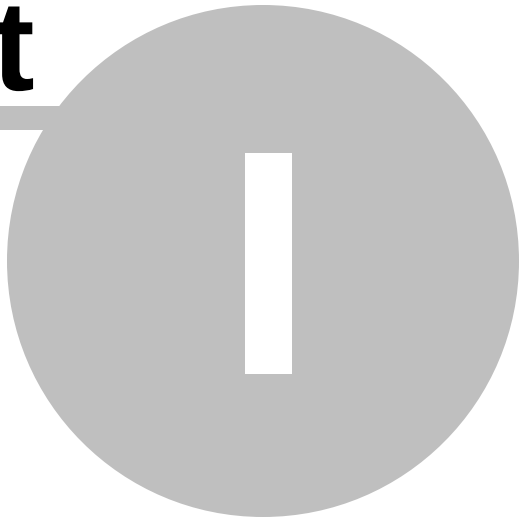
isPosInside ... 25

Remove ... 25
remove ... 25
show ... 25
Update ... 25
update ... 25
EventItem Object ... 26
remove ... 26
Marker Object ... 26
getId ... 26
getImgConfig ... 26
getMarkerConfig ... 26
getPoint ... 27
getTitleConfig ... 27
getType ... 27
hide ... 27
isHidden ... 27
Remove ... 27
remove ... 28
show ... 28
Update ... 28
update ... 28
updatePoint ... 29
VectorCircle Object ... 29
getConfig ... 29
getId ... 30
getPoint ... 30
getRadius ... 30
getType ... 30
hide ... 30
isHidden ... 30
remove ... 31
show ... 31
update ... 31
updateConfig ... 32
updatePoint ... 32
updateRadius ... 32
VectorLine Object ... 33
getConfig ... 33
getId ... 33
getPoint ... 33
getType ... 33
hide ... 34
isHidden ... 34
Remove ... 34
remove ... 34
show ... 34
update ... 34
updateConfig ... 35
updatePoint ... 35
addCustomCanvas ... 36
addDivMarker ... 36
addEventItem ... 37
addListener ... 38
addMarker ... 38

addMarker ... 39
addVectorCircle ... 41
addVectorLine ... 42
clearEventItem ... 44
exportMap ... 44
getCenter ... 44
getCustomCanvas ... 44
getDatasets ... 45
getDivMarker ... 45
getEventItem ... 46
getMaplayers ... 46
getMarker ... 47
getZoom ... 47
panDir ... 47
removeAllCustomCanvases ... 48
removeAllDivMarkers ... 48
removeAllMarkers ... 48
removeAllVectorCircles ... 48
removeAllVectorLines ... 49
removeCustomCanvas ... 49
removeDivMarker ... 49
removeListener ... 49
removeMarker ... 50
removeVector ... 50
removeVectorCircle ... 50
removeVectorLine ... 50
repaint ... 51
route ... 51
routeCheckPoint ... 52
search ... 53
setCenter ... 55
setDataset ... 55
setDataset ... 55
setMaplayer ... 56
setMaplayer ... 56
setZoom ... 56
showMyLocation ... 57

Part VI Examples **60**

Part



1 Copyright

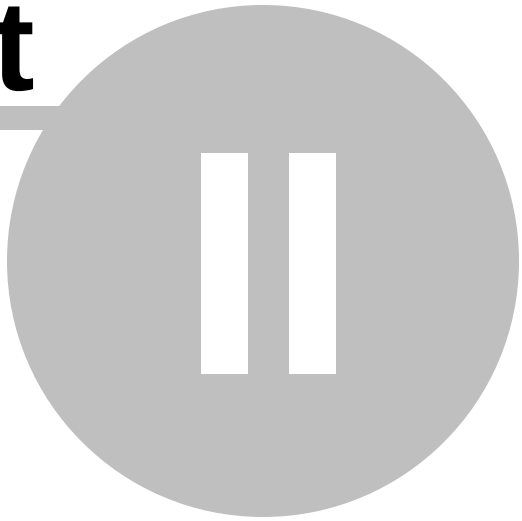
SiteWatch API Help.

Please note:
If you find any bugs or something wrong in the API or in this help please let us know.
Also if you want to suggest new features please let us also know.

Your feedback is welcome and important to us so we can make sure the API is at its best.

You can send us e-mail at: support@samsyn.is with the subject: SiteWatch API bugs/new features.

Part



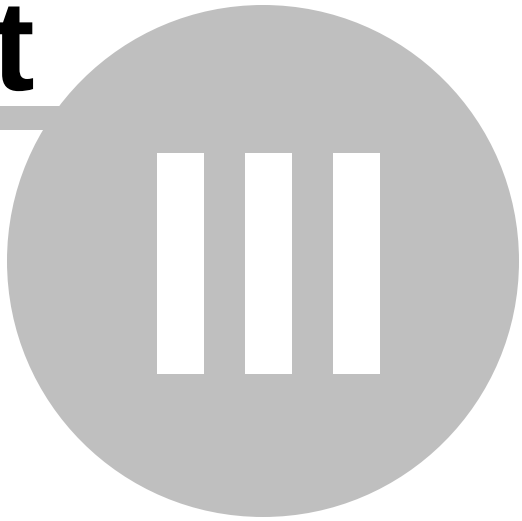
2 SiteWatch API

SiteWatch Application Programming Interface (SWAPI) is a programming development environment that allows programmers to include mapping capabilities to their websites and customize it to their needs.

Map	Description
Lower right corner:	Map scale (5.000.000 = highest, 2.000 = lowest).
Lower left corner:	Zoom level (0 = highest, 10 = lowest).
Upper right corner:	Pan direction.
Blue line:	Example of a Vector line.
Image of house and text 'Samsyn':	Example of a Marker.
Center text 'Hello World':	Example of a DivMarker.

Zoom in and Zoom out and Pan can also be done with the mouse; use mouse scroll up and down and mouse click and drag the map to any direction.

Part



3 SiteWatch_API Class

A static class.

3.1 Projection

A collection of methods to project map coordinates between mapping systems.

3.1.1 ISN93_To_WGS84

ISN93 is a medium and small scale topographic mapping for Iceland.

A method to project map coordinates between the Icelandic ISN93 system to the world WGS84 system.

Parameter	Value
x: number	isn93 x-coordinate.
y: number	isn93 y-coordinate.

Return	Value
object (lat, lon):	A point set object (**not SWPoint**) in WGS84 projection.

Example:
```
var wgs84 = SiteWatch_API.ISN93_To_WGS84(300000, 400000);
```

Or to go full circle:

```
var wgs84 = SiteWatch_API.ISN93_To_WGS84(300000, 400000);
var isn93 = SiteWatch_API.WGS84_To_ISN93(wgs84.lat, wgs84.lon);
```

3.1.2 WGS84_To_ISN93

WGS84 is a horizontal component of 3D system and is used by the GPS satellite navigation system and for NATO military geodetic surveying of the World.

A method to project map coordinates between the world WGS84 system to Icelandic ISN93 system.

Parameter	Value
x: number	wgs84 x-coordinate.
y: number	wgs84 y-coordinate.

Return	Value
object (x, y):	A point set object (**not SWPoint**) in ISN93 projection.

Example:
```
var isn93 = SiteWatch_API.WGS84_To_ISN93(64.1000, 21.5700);
```

Or to go full circle:

```
var isn93 = SiteWatch_API.WGS84_To_ISN93(64.1000, 21.5700);
var wgs84 = SiteWatch_API.ISN93_To_WGS84(isn93.x, isn93.y);
```

Part

4 SWPoint Class

A point class.

Properties	Description
x: number	x-coordinate.
y: number	y-coordinate.

Methods	Description
create(x: number, y: number): SWPoint	Creates a SWPoint object.

Static Methods	Description
sum(point1: SWPoint, point2: SWPoint): SWPoint	Adds two SWPoints together.
subtract(point1: SWPoint, point2: SWPoint): SWPoint	Subtracts one SWPoint from another SWPoint.
scale(point: SWPoint, n: integer): SWPoint	Multiplies SWPoints to a number.

4.1 create

A method to create a point object.

Parameter	Value
x: number	x-coordinate.
y: number	y-coordinate.

Return	Value
object:	A point set object (**SWPoint**).

Example:
```
var point = SWPoint.create(359583, 406481);
```

4.2 Static Methods

Static methods to manipulate a point set object (**SWPoint**).

4.2.1 scale

A method to multiply SWPoints to a number.

Parameter	Value
point: a point set object (**SWPoint**)	isn93 map point locations.
scale: integer	Scale in pixels.

Return	Value
object:	A point set object (**SWPoint**).

Example:
```
var pointToScale = SWPoint.create(359583, 406481);

var pointScaled = SWPoint.subtract(pointToScale, 5);
```

4.2.2 subtract

A method to subtract one SWPoint from another SWPoint.

Parameter	Value
point1: a point set object (**SWPoint**)	isn93 map point locations.
point2: a point set object (**SWPoint**)	isn93 map point locations.

Return	Value
object:	A point set object (**SWPoint**).

Example:
```
var point1 = SWPoint.create(359583, 406481);
var point2 = SWPoint.create(359603, 406471);

var point = SWPoint.subtract(point1, point2);
```

4.2.3 sum

A method to add two SWPoints together.

Parameter	Value
point1: a point set object (**SWPoint**)	isn93 map point locations.
point2: a point set object (**SWPoint**)	isn93 map point locations.

Return	Value
object:	A point set object (**SWPoint**).

Example:
```
var point1 = SWPoint.create(359583, 406481);
var point2 = SWPoint.create(359603, 406471);

var point = SWPoint.sum(point1, point2);
```

Part V

5 SWMap Class

A map class.

Methods	Description
create(id, config): SWMap	Creates a SWMap object.

5.1 Bundled Images

Bundled images from `http://kort.samsyn.is/api/img/` that can be used with the `url` parameter of marker objects or you can create and use your own images from your own site location.

Image	Filename
	house.png
	tip.png
	key.png
	info/find.png
	info/help.png
	info/globe.png
	info/history.png
	info/view.png
	pin/pin_bent.png -or- /pin/pin_bent_2.png
	pin/pin_black.png -or- /pin/pin_black_2.png
	pin/pin_blue.png -or- /pin/pin_blue_2.png
	pin/pin_red.png -or- /pin/pin_red_2.png
	pin/pin_yellow.png -or- /pin/pin_yellow_2.png
	location/bluedot.png
	location/blue-dot.png

5.2 Types of Listeners

Map type events:
Returns a reference to the map and browser events that may be different depending on what browser is used.

Map Type	Description
`zoomchange(map, zoom)`	Occurs when zooming of the map is changed. Returns a reference to the map changed, and the new zoom level value.
`mousedown(map, e)`	Occurs when clicking and holding the mouse button down on the map.
`mousemove(map, e)`	Occurs when moving the mouse over the map.
`mouseup(map, e)`	Occurs when releasing a holding mouse button down on the map.
`dblclick(map, e)`	Occurs when double-clicking on the map.
`mouseout(map, e)`	Occurs when the mouse leaves the map.
`mouseover(map, e)`	Occurs when the mouse enters the map.
`click(map, e)`	Occurs when clicking on the map.
`move(map)`	Occurs when the map is panned.
`movestart(map)`	Occurs when the map panning starts.
`moveend(map)`	Occurs when the map panning ends.
`touchstart(map)`	Occurs when your finger enters the screen of your smartphone.
`touchend(map)`	Occurs when your finger leaves the screen of your smartphone.

Marker type events:
Returns a reference to the map where the marker is, list of markers on the map and browser events that may be different depending on what browser is used.

Marker Type	Description
`markers_click(map, idlist, e)`	Occurs when clicking on a marker on the map.
`markers_dblclick(map, idlist, e)`	Occurs when double-clicking on a marker on the map.
`markers_mousedown(map, idlist, e)`	Occurs when clicking and holding the mouse button down on a marker on the map.
`markers_mousemove(map, idlist, e)`	Occurs when moving the mouse over a marker on the map.
`markers_mouseout(map, idlist, e)`	Occurs when the mouse leaves a marker on the map.
`markers_mouseover(map, idlist, e)`	Occurs when the mouse enters a marker on the map.
`markers_touchstart(map, idlist, e)`	Occurs when your finger enters the screen of your smartphone over a marker.
`markers_touchend(map, idlist, e)`	Occurs when your finger leaves the screen of your smartphone from a marker.

DivMarker type events:
Returns a reference to the map where the divmarker is, list of divmarkers on the map and browser events that may be different depending on what browser is used.

DivMarker Type	Description
`divmarkers_click(map, idlist, e)`	Occurs when clicking on a divmarker on the map.
`divmarkers_dblclick(map, idlist, e)`	Occurs when double-clicking on a divmarker on the map.
`divmarkers_mousedown(map, idlist, e)`	Occurs when clicking and holding the mouse button down on a divmarker on the map.
`divmarkers_mousemove(map, idlist, e)`	Occurs when moving the mouse over a divmarker on the map.
`divmarkers_mouseout(map, idlist, e)`	Occurs when the mouse leaves a divmarker on the map.
`divmarkers_mouseover(map, idlist, e)`	Occurs when the mouse enters a divmarker on the map.
`divmarkers_touchstart(map, idlist, e)`	Occurs when your finger enters the screen of your smartphone over a divmarker.
`divmarkers_touchend(map, idlist, e)`	Occurs when your finger leaves the screen of your smartphone from a divmarker.

EventItem type events:
Returns a reference to the map where the eventitem is, list of eventitems on the map and browser events that may be different depending on what browser is used.

EventItem Type	Description
`eventitems_click(map, idlist, e)`	Occurs when clicking on an eventitem on the map.
`eventitems_dblclick(map, idlist, e)`	Occurs when double-clicking on an eventitem on the map.
`eventitems_mousedown(map, idlist, e)`	Occurs when clicking and holding the mouse button down on an eventitem on the map.
`eventitems_mousemove(map, idlist, e)`	Occurs when moving the mouse over an eventitem on the map.
`eventitems_mouseout(map, idlist, e)`	Occurs when the mouse leaves an eventitem on the map.
`eventitems_mouseover(map, idlist, e)`	Occurs when the mouse enters an eventitem on the map.
`eventitems_touchstart(map, idlist, e)`	Occurs when your finger enters the screen of your smartphone over an eventitem.
`eventitems_touchend(map, idlist, e)`	Occurs when your finger leaves the screen of your smartphone from an eventitem.

5.3 create

A method to create a map object.

Parameter	Value
id: string	Identity of object.
config (optional):	Configuration.

Parameter	Value
eventItem: boolean (default=false)	Enable event items.
overLayer (optional):	Overlayer.
name: string	Name of overlayer.
function: name	An asynchronous JavaScript function that implements the logic when the function call returns. Needs to return { url: "<image>", opacity: <number> } for the overlay.
scope: name (optional)	Owner of the current object.
panButton: boolean (default=false)	Show the pan button on the map.
zoomButton: boolean (default=false)	Show the zoom button on the map.
hideScaleText: boolean (default=false)	Hide the map scale on the map.
disableZoom: boolean (default=false)	Disable zoom with mouse.
disablePan: boolean (default=false)	Disable pan with mouse.
disableZoomWheel: boolean (default=false)	Disable zoom with mouse wheel.
defaultZoom: integer (default=0)	Default zoom level (0 = highest, 10 = lowest).
defaultCenterPoint: a point set (default={x: 500000, y: 500000})	Default map center location (isn93 map point locations).
showMyLocation: boolean (default=false) (optional):	Track your GPS location on the map.

Return	Value
object:	A map object.

Example:

```
var map = SWMap.create("map",
{
   panButton: true,
   zoomButton: true,
   defaultZoom: 6,
   defaultCenterPoint: { x: 358762, y: 407377 }
});
```

Or

```
var point = SWPoint.create(358762, 407377);
var map = SWMap.create("map",
{
```

```
    panButton: true,
    zoomButton: true,
    defaultZoom: 6,
    defaultCenterPoint: point,
    showMyLocation: true
});
```

5.4 Map Object

A map object.

Contains the loaded map and methods that can be used to interact with it.

5.4.1 CustomCanvas Object

A custom canvas object.

Custom canvas differs from other drawing object in that sense that the user has full control of what is going to be drawn.

Animation can be used if the user prefers.

Canvas support is required for this object to work as the user is in full control.

5.4.1.1 forceRedraw

A method that forces a redraw of the custom convas in a map location.

Parameter	Value
timeout: integer	Timeout in millisecond how long we want to delay redrawing.

Example:
```
customCanvas.forceRedraw(1000);
```

5.4.1.2 getConfig

A method that returns a custom canvas config in a map location.

Return	Value
object:	A custom canvas config object.

Example:
```
var config = customCanvas.getConfig();
```

5.4.1.3 getId

A method that returns a custom canvas id in a map location.

Return	Value
string:	A custom canvas id.

Example:
```
var id = customCanvas.getId();
```

5.4.1.4 getType

A method that returns a custom canvas type in a map location.

Return	Value
string:	A custom canvas type.

Example:
```
var type = customCanvas.getType();
```

5.4.1.5 hide

A method to hide a custom canvas in a map location.

Example:
```
customCanvas.hide();
```

5.4.1.6 isHidden

A method that returns a custom canvas visibility in a map location.

Return	Value
boolean:	A custom canvas visibility.

Example:
```
var visible = customCanvas.isHidden();
```

5.4.1.7 isTainted

A method that returns a custom canvas taintedness in a map location.

If we have been using image data from external servers without correct settings the canvas is tainted.
More information at https://developer.mozilla.org/en-US/docs/Web/HTML/CORS_enabled_image

Return	Value
boolean:	A custom canvas tainted.

Example:
```
var tainted = customCanvas.isTainted();
```

5.4.1.8 remove

A method to remove a custom canvas from a map location.

```
customCanvas.remove();
```

5.4.1.9 setDrawHandler

A method to change the drawing handler of the custom canvas after it has been created on the map.

Parameter	Value
function: name	An asynchronous JavaScript function that implements the logic when the function call returns.

Example:

```
function drawHandler(map, context, item, status)
{
   //context = Canvas context.
   //item = The custom canvas object.
   //status = object having information about the drawing status.

   var s = status.animation; //Drawing function is being called from
   animation frame (true) or not (false).
}

customCanvas.setDrawHandler(drawHandler);
```

5.4.1.10 show

A method to show a custom canvas in a map location.

Example:

```
customCanvas.show();
```

5.4.1.11 updateConfig

A method to update a custom canvas configuration in a map location.

Parameter	Value
config:	Configuration.
animated: boolean (default=false)	Defines whether the custom canvas is animated or not. If it is set as true then the canvas will be animated and the drawHandler of the custom canvas will be called for every frame.
zIndex: integer	Defines the z-index of the canvas.

Example:

```
var newConf = customCanvas.GetConfig();

newConfig.animated = true;
newConfig.zIndex = 1;

customCanvas.updateConfig(newConf);
```

5.4.1.12 xToWinX

A method that returns a x pixel position on custom canvas representing the lat location from ISN93.

Parameter	Value
x: number	isn93 x-coordinate.

Return	Value
position: integer	A x pixel position on canvas relating to input x.

Example:
```
var position = customCanvas.xToWinX();
```

5.4.1.13 yToWinY

A method that returns a y pixel position on custom canvas representing the lon location from ISN93.

Parameter	Value
y: number	isn93 y-coordinate.

Return	Value
position: integer	A y pixel position on canvas relating to input y.

Example:
```
var position = customCanvas.yToWinY();
```

5.4.2 DivMarker Object

A divmarker object.

This is an advanced do-it-yourself object that gives more control of the web page than the marker object.

5.4.2.1 getConfig

A method that returns a divmarker config in a map location.

Return	Value
object:	A divmarker config object.

Example:
```
var config = divmarker.getConfig();
```

5.4.2.2 getDom

A method that returns a divmarker dom in a map location.

Return	Value
object:	A divmarker dom object.

Example:
```
var divmarker = map.getDivMarker("hello");

var dom = divmarker.getDom();
dom.innerHTML = "Hello World";

divmarker.update(); //Update the divmarker so the location is updated.
```

5.4.2.3 getId

A method that returns a divmarker id in a map location.

Return	Value
string:	A divmarker id.

Example:
```
var id = divmarker.getId();
```

5.4.2.4 getPoint

A method that returns a divmarker point in a map location.

Return	Value
Point location { x: <x>, y: <y>}:	isn93 map point locations.

Example:
```
var point = divmarker.getPoint();
```

5.4.2.5 getType

A method that returns a divmarker type in a map location.

Return	Value
string:	A divmarker type.

Example:
```
var type = divmarker.getType();
```

5.4.2.6 hide

A method to hide a divmarker in a map location.

Example:
```
divmarker.hide();
```

5.4.2.7 isHidden

A method that returns a divmarker visibility in a map location.

Return	Value
boolean:	A divmarker visibility.

Example:
```
var visible = divmarker.isHidden();
```

5.4.2.8 isPosInside

A method that returns a divmarker position inside in a map location.

Return	Value
boolean:	A divmarker pos inside.

Example:
```
var posInside = divmarker.isPosInside();
```

5.4.2.9 Remove

This method has been deprecated use **remove** instead.

5.4.2.10 remove

A method to remove a divmarker from a map location.

```
divmarker.remove();
```

5.4.2.11 show

A method to show a divmarker in a map location.

Example:
```
divmarker.show();
```

5.4.2.12 Update

This method has been deprecated use **update** instead.

5.4.2.13 update

A method to update a divmarker configuration in a map location.

Parameter	Value
config (optional):	Configuration.
offset: a point set	Point locations { x: <x>, y: <y>} in pixels from the center point of the divmarker the text should be moved to. This is useful so that the text doesn't appear directly inside the center point.
updateAll: boolean (default=false)	Update all parameter values or merge with current parameter values.

Example:
```
divmarker.update( { offset: {x: 0, y: 0} } );
```

Or

```
divmarker.update();
```

5.4.3 EventItem Object

An event item object.

5.4.3.1 remove

A method to remove an event item from a map location.

```
eventitem.remove();
```

5.4.4 Marker Object

A marker object.

5.4.4.1 getId

A method that returns a marker id in a map location.

Return	Value
string:	A marker id.

Example:
```
var id = marker.getId();
```

5.4.4.2 getImgConfig

A method that returns a marker img config in a map location.

Return	Value
object:	A marker img config object.

Example:
```
var config = marker.getImgConfig();
```

5.4.4.3 getMarkerConfig

A method that returns a marker config in a map location.

Return	Value
object:	A marker config object.

Example:
```
var config = marker.getMarkerConfig();
```

5.4.4.4 getPoint

A method that returns a marker point in a map location.

Return	Value
Point location { x: <x>, y: <y>}:	isn93 map point locations.

Example:
```
var point = marker.getPoint();
```

5.4.4.5 getTitleConfig

A method that returns a marker title config in a map location.

Return	Value
object:	A marker title config object.

Example:
```
var config = marker.getTitleConfig();
```

5.4.4.6 getType

A method that returns a marker type in a map location.

Return	Value
string:	A marker type.

Example:
```
var type = marker.getType();
```

5.4.4.7 hide

A method to hide a marker in a map location.

Example:
```
marker.hide();
```

5.4.4.8 isHidden

A method that returns a marker visibility in a map location.

Return	Value
boolean:	A marker visibility.

Example:
```
var visible = marker.isHidden();
```

5.4.4.9 Remove

This method has been deprecated use **remove** instead.

5.4.4.10 remove

A method to remove a marker from a map location.

```
marker.remove();
```

5.4.4.11 show

A method to show a marker in a map location.

Example:
```
marker.show();
```

5.4.4.12 Update

This method has been deprecated use **update** instead.

5.4.4.13 update

A method to update a marker configuration in a map location.

Parameter	Value
titleConfig (optional):	Title configuration.
text: string	Text of marker.
fgColor: string	Foreground color of marker text.
bgColor: string	Background color of marker text.
css: url (to css file)	Cascading style sheet of marker text. Can be used to format the title, add bold, increase font size etc.
border: string	Border of marker text.
padding: string	Padding of marker text.
imgConfig (Optional)	Image configuration.
url (to image file)	Image of marker.
bgColor: string	Background color of marker image.
offset: a point set	Point locations { x: <x>, y: <y>} in pixels from the center point of the marker the image should be moved to. This is useful so that the image doesn't appear directly inside the center point.
css: url (to css file)	Cascading style sheet of marker image. Can be used to format the title, add bold, increase font size etc.
border: string	Border of marker image.
padding: string	Padding of marker image.
markerConfig	Marker configuration.
bgColor: string	Background color of marker.
css: url (to css file)	Cascading style sheet of marker. Can be used to format the title, add bold, increase font size etc.
border: string	Border of marker.
padding: string	Padding of marker.

Parameter	Value
updateAll: boolean (default=false)	Update all parameter values or merge with current parameter values.

Example:
```
marker.update( { text: "Company" } );
```

Or

```
var newTitleConfig = { padding: "10px", border: "12px", text: "Update
example", fgColor: "green", bgColor: "yellow", css: "" };

var newImgConfig = { padding: "1px", url: "http://kort.samsyn.is/api/img/
key.png",
  offset: { x: 10, y: 12 }, bgColor: "red", border: "1px", css: "" };

var newMarkerConfig = { bgColor: "orange", border: "2px", padding: "2px",
display: "block", css: "" };

marker.update(newTitleConfig, newImgConfig, newMarkerConfig);
```

5.4.4.14 updatePoint

A method to update a marker point in a map location.

Parameter	Value
point: a point set object (**SWPoint**) -or- point: a point set array	isn93 map point locations.

Example:
```
var point = SWPoint.create(359583, 406481);

marker.updatePoint(point);
```

5.4.5 VectorCircle Object

A vector circle object.

Circles consisting of a point and radius can be drawn on the map. The circles can be of different radius, size and color.

5.4.5.1 getConfig

A method that returns a vector circle config in a map location.

Return	Value
object:	A vector circle config object.

Example:
```
var config = circle.getConfig();
```

5.4.5.2 getId

A method that returns a vector circle id in a map location.

Return	Value
string:	A vector circle id.

Example:
```
var id = circle.getId();
```

5.4.5.3 getPoint

A method that returns a vector circle point in a map location.

Return	Value
Point location { x: <x>, y: <y>}:	isn93 map point locations.

Example:
```
var point = circle.getPoint();
```

5.4.5.4 getRadius

A method that returns a vector circle radius in a map location.

Return	Value
string:	A vector circle radius.

Example:
```
var radius = circle.getRadius();
```

5.4.5.5 getType

A method that returns a vector circle type in a map location.

Return	Value
string:	A vector circle type.

Example:
```
var type = circle.getType();
```

5.4.5.6 hide

A method to hide a vector circle in a map location.

Example:
```
circle.hide();
```

5.4.5.7 isHidden

A method that returns a vector circle visibility in a map location.

Return	Value
boolean:	A vector circle visibility.

Example:
```
var visible = circle.isHidden();
```

5.4.5.8 remove

A method to remove a vector circle from a map location.

```
circle.remove();
```

5.4.5.9 show

A method to show a vector circle in a map location.

Example:
```
circle.show();
```

5.4.5.10 update

A method to update a vector circle configuration in a map location.

Parameter	Value
point: a point set object (**SWPoint**) -or- point: a point set array	isn93 map point locations.
radius: integer	Radius of circle from point in meters.
config:	Configuration.

lineColor: string	Line color of circle.
	Hex #ffffff
	rgb(0,0,0)
	rgba(0,0,0,0.1)
	colornames (red, blue)

fillColor: string	Fill color of circle.
	Hex #ffffff
	rgb(0,0,0)
	rgba(0,0,0,0.1)
	colornames (red, blue)

lineWidth: integer	Width of circle.

Example:
```
var point = { x: 339587, y: 361482 };

var newRadius = 50000;

var newConf = { lineColor: 'ff0000', fillColor: '#6dd341', lineWidth: 10 };

circle.update(point, newRadius, newConf);
```

5.4.5.11 updateConfig

A method to update a vector circle configuration in a map location.

Parameter	Value
config:	Configuration.

lineColor: string	Line color of circle.
	Hex #ffffff
	rgb(0,0,0)
	rgba(0,0,0,0.1)
	colornames (red, blue)

fillColor: string	Fill color of circle.
	Hex #ffffff
	rgb(0,0,0)
	rgba(0,0,0,0.1)
	colornames (red, blue)

lineWidth: integer	Width of circle.
minSize: integer	Minimum width in pixels.

Example:

```
var newConf = { lineColor: 'ff0000', fillColor: '#6dd341', lineWidth: 10,
minSize: 5 };

circle.updateConfig(newConf);
```

5.4.5.12 updatePoint

A method to update a vector circle point in a map location.

Parameter	Value
point: a point set object (**SWPoint**) -or- point: a point set array	isn93 map point locations.

Example:

```
var point = SWPoint.create(359583, 406481);

circle.updatePoint(point);
```

5.4.5.13 updateRadius

A method to update a vector circle radius in a map location.

Parameter	Value
radius: integer	Radius of circle from point in meters.

Example:

```
var newRadius = 50000;
```

```
circle.updateRadius(newRadius);
```

5.4.6 VectorLine Object

A vector line object.

Lines consisting of at least two points can be drawn on the map. The lines can be of different size and color.

5.4.6.1 getConfig

A method that returns a vector line config in a map location.

Return	Value
object:	A vector line config object.

Example:
```
var config = line.getConfig();
```

5.4.6.2 getId

A method that returns a vector line id in a map location.

Return	Value
string:	A vector line id.

Example:
```
var id = line.getId();
```

5.4.6.3 getPoint

A method that returns a vector line point in a map location.

Return	Value
Point location { x: <x>, y: <y>}:	isn93 map point locations.

Example:
```
var point = line.getPoint();
```

5.4.6.4 getType

A method that returns a vector line type in a map location.

Return	Value
string:	A vector line type.

Example:
```
var type = line.getType();
```

5.4.6.5 hide

A method to hide a vector line in a map location.

Example:
```
line.hide();
```

5.4.6.6 isHidden

A method that returns a vector line visibility in a map location.

Return	Value
boolean:	A vector line visibility.

Example:
```
var visible = line.isHidden();
```

5.4.6.7 Remove

This method has been deprecated use **remove** instead.

5.4.6.8 remove

A method to remove a vector line from a map location.

```
line.remove();
```

5.4.6.9 show

A method to show a vector line in a map location.

Example:
```
line.show();
```

5.4.6.10 update

A method to update a vector line configuration in a map location.

Parameter	Value
point: a point set objects (**SWPoint**) -or- point: a point set array	isn93 map point locations.
config:	Configuration.

lineColor: string	Color of line.
	Hex #ffffff
	rgb(0,0,0)
	rgba(0,0,0,0.1)
	colornames (red, blue)

lineWidth: integer	Width of line.

Example:
```
var point = { x: 339587, y: 361482 };
```

```
var newPtn = line1.getPoint();

newPtn[0].x = 644088;
newPtn[0].y = 467236;
newPtn[1].x = 644088;
newPtn[1].y = 467236;

var newConf = line.getConfig();

newConfig.lineColor = "blue";
newConfig.lineWidth = 10;

line.update(newPtn, newConf);
```

5.4.6.11 updateConfig

A method to update a vector line configuration in a map location.

Parameter	Value
config:	Configuration.

lineColor: string	Color of line.
	Hex #ffffff
	rgb(0,0,0)
	rgba(0,0,0,0.1)
	colornames (red, blue)

lineWidth: integer	Width of line.

Example:

```
var newConf = line.GetConfig();

newConfig.lineColor = "blue";
newConfig.lineWidth = 10;

line.updateConfig(newConf);
```

5.4.6.12 updatePoint

A method to update a vector line point in a map location.

Parameter	Value
point: a point set object (**SWPoint**) -or- point: a point set array	isn93 map point locations.

Example:

```
var point = SWPoint.create(359583, 406481);

line.updatePoint(point);
```

5.4.7 addCustomCanvas

A method to add a custom canvas with specific drawHandler to a map location.

Parameter	Value
id: string	Identity of object.
function: name	An asynchronous JavaScript function that implements the logic when the function call returns.
scope: name (optional)	Owner of the current object.
config:	Configuration.
animated: boolean (default=false)	Defines whether the custom canvas is animated or not. If it is set as true then the canvas will be animated and the drawHandler of the custom canvas will be called for every frame.
zIndex: integer	Defines the z-index of the canvas.
repaint: boolean (optional) (default=true)	Redraw line on map.

Return	Value
object:	A custom canvas object.

Example:

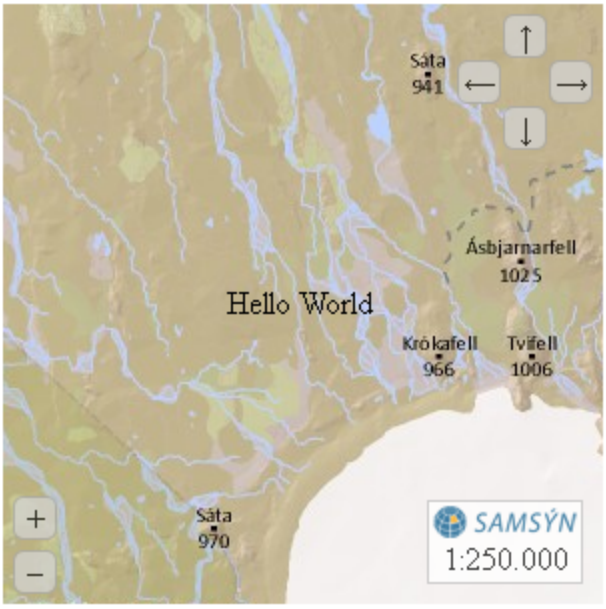
```
var customCanvas = map.addCustomCanvas("canvas", drawHandler);
```

5.4.8 addDivMarker

A method to add a divmarker to a map location.

Parameter	Value
id: string	Identity of object.
point: a point set object (**SWPoint**) -or- point: a point set array	isn93 map point locations.
config (optional):	Configuration.
offset: a point set	Point locations { x: <x>, y: <y>} in pixels from the center point of the divmarker the text should be moved to. This is useful so that the text doesn't appear directly inside the center point.

Return	Value
object:	A divmarker object.



Example:

```
var divmarker = map.addDivMarker("hello",
SWPoint.create(500000, 500000));

var dom = divmarker.getDom();
dom.innerHTML = "Hello World";

divmarker.update(); //Update the divmarker so the location is updated.
```

5.4.9 addEventItem

A method to add an event item to a map location.

Parameter	Value
id: string	Identity of object.
left:	isn93 map left location.
right:	isn93 map right location.
top:	isn93 map top location.
bottom:	isn93 map bottom location.

Return	Value
object:	An event item object.

Example:

```
var eventitem = map.addEventItem("event",
{ left: 359583 - 20, right: 359583 + 20, top: 406481 + 20, bottom: 406481 -
20 } );
```

5.4.10 addListener

A method to add a listener background event.

Parameter	Value
type: string	A type of listener to create. See Types 16.
function: name	An asynchronous JavaScript function that implements the logic when the function call returns.
scope: name (optional)	Owner of the current object.

Example:

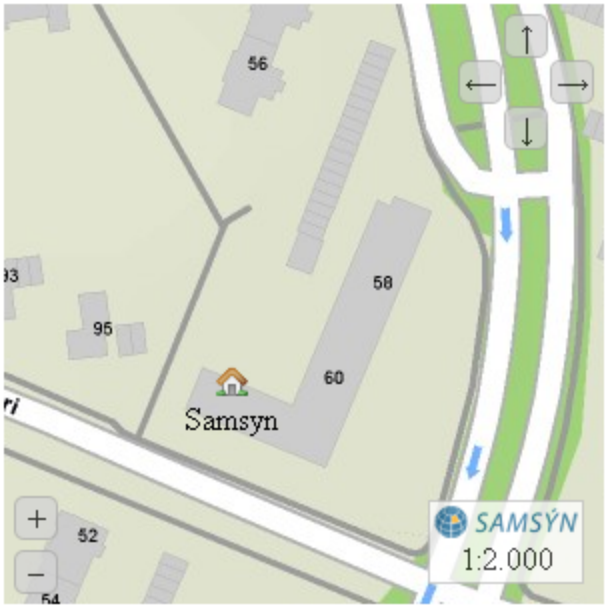
```
function zoomChange(map, zoom)
{
   //Your code here!
   //map.<method>;
}

function mousemoveChange(map, e)
{
   //Your code here!
   //map.<method>;
}

function markersClick(map, idlist, e)
{
   //Your code here!
   //map.<method>;
}

map.addListener("zoomchange", zoomChange, this);
map.addListener("mousemove", mousemoveChange, this);
map.addListener("markers_click", markersClick, this);
```

5.4.11 addMarker

A method to add a marker to a map location.

Parameter	Value
id: string	Identity of object.
point: a point set object (**SWPoint**) -or- point: a point set array	isn93 map point locations.
text: string	Title of marker.
url (to image file)	Image of marker.

Return	Value
object:	A marker object.



Example:

```
var marker = map.addMarker("house",
SWPoint.create(359583, 406481),
"Samsyn", "http://kort.samsyn.is/api/img/house.png");
```

5.4.12 addMarker

A method to add a marker to a map location.

Parameter	Value
id: string	Identity of object.
point: a point set object (**SWPoint**) -or- point: a point set array	isn93 map point locations.
titleConfig (optional):	Title configuration.
text: string	Text of marker.

fgColor: string	Foreground color of marker text.
	Hex #ffffff
	rgb(0,0,0)
	rgba(0,0,0,0.1)
	colornames (red, blue)

bgColor: string	Background color of marker text.
	Hex #ffffff
	rgb(0,0,0)
	rgba(0,0,0,0.1)
	colornames (red, blue)

css: url (to css file)	Cascading style sheet of marker text.

Parameter	Value
	Can be used to format the title, add bold, increase font size etc.
border: string	Border of marker text.
padding: string	Padding of marker text.
imgConfig (Optional)	Image configuration.
url (to image file)	Image of marker.

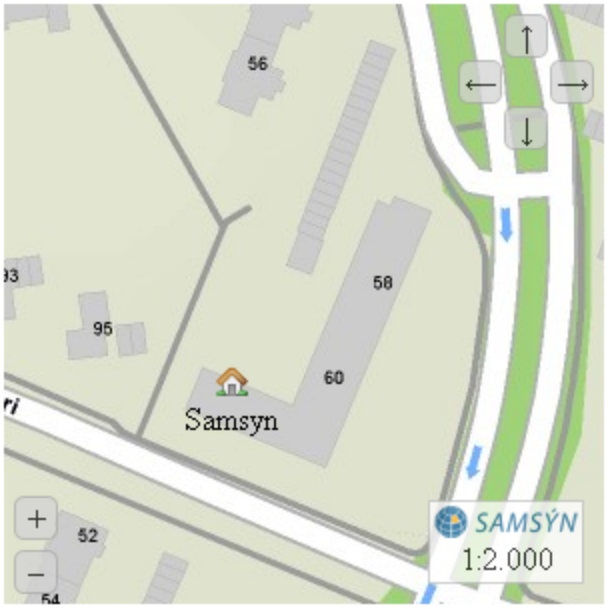
bgColor: string	Background color of marker image.
	Hex #ffffff
	rgb(0,0,0)
	rgba(0,0,0,0.1)
	colornames (red, blue)

offset: a point set	Point locations { x: <x>, y: <y>} in pixels from the center point of the marker the image should be moved to. This is useful so that the image doesn't appear directly inside the center point.
css: url (to css file)	Cascading style sheet of marker image. Can be used to format the title, add bold, increase font size etc.
border: string	Border of marker image.
padding: string	Padding of marker image.
markerConfig	Marker configuration.

bgColor: string	Background color of marker.
	Hex #ffffff
	rgb(0,0,0)
	rgba(0,0,0,0.1)
	colornames (red, blue)

css: url (to css file)	Cascading style sheet of marker. Can be used to format the title, add bold, increase font size etc.
display: string	Cascading style sheet display of marker. Can be used to change the behavior or which type of box a marker is displayed inside.
border: string	Border of marker.
padding: string	Padding of marker.

Return	Value
object:	A marker object.



Example:

```
var marker = map.addMarker("house",
SWPoint.create(359583, 406481),
{ text: "Samsyn" }, { url: "http://kort.samsyn.is/api/img/house.png" });
```

5.4.13 addVectorCircle

A method to add a vector line to a map location.

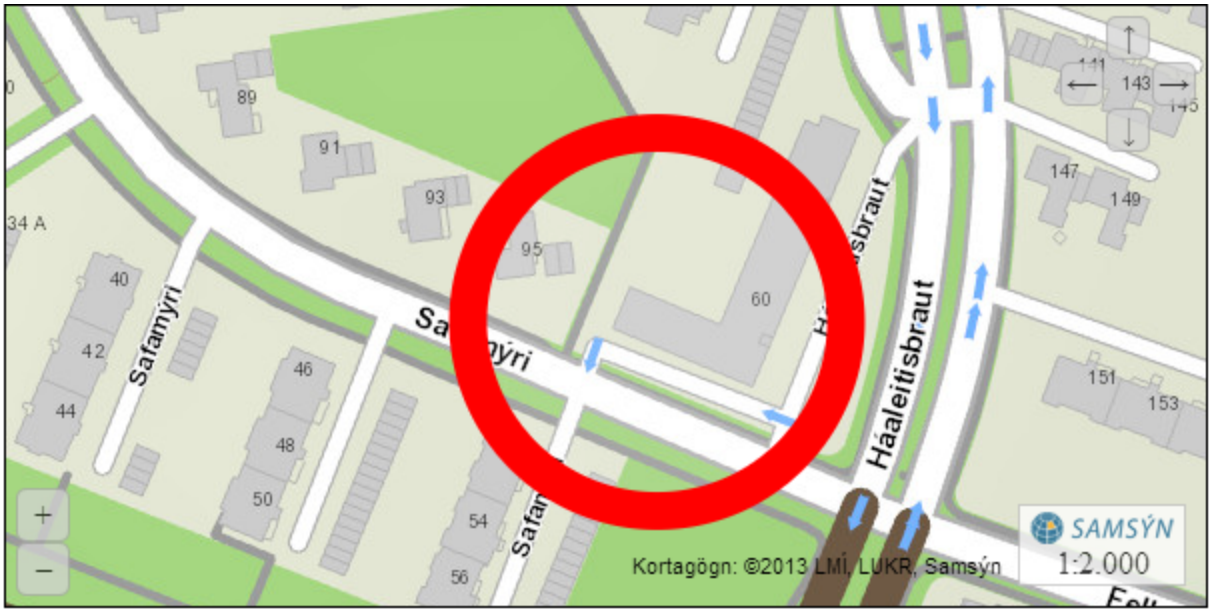
Parameter	Value
id: string	Identity of object.
point: a point set object (**SWPoint**) -or- point: a point set array	isn93 map point locations.
radius: integer	Radius of circle from point in meters.
config:	Configuration.

lineColor: string	Line color of circle.
	Hex #ffffff
	rgb(0,0,0)
	rgba(0,0,0,0.1)
	colornames (red, blue)

fillColor: string	Fill color of circle.
	Hex #ffffff
	rgb(0,0,0)
	rgba(0,0,0,0.1)
	colornames (red, blue)

lineWidth: integer	Width of circle.
repaint: boolean (optional) (default=true)	Redraw circle on map.

Return	Value
object:	A vector circle object.



Example:

```
var circle = map.addVectorCircle("circle",
[{ x: 359583, y: 406481 }], 100,
{ lineColor: "red", lineWidth: 10 });
```

Or

```
var point1 = SWPoint.create(359583, 406481);
var point2 = SWPoint.create(359603, 406471);

var circle1 = map.addVectorCircle("circle1",
[point1], 75,
{ lineColor: "red", fillColor: "rgba(249, 95, 94, 0)", lineWidth: 10 },
false);

var circle2 = map.addVectorCircle("circle2",
[point2], 125,
{ lineColor: "blue", lineWidth: 10 }, true);
```

5.4.14 addVectorLine

A method to add a vector line to a map location.

Parameter	Value
id: string	Identity of object.
point: a point set objects (**SWPoint**) -or- point: a point set array	isn93 map point locations.

Parameter	Value
config:	Configuration.

lineColor: string	Color of line.
	Hex #ffffff
	rgb(0,0,0)
	rgba(0,0,0,0.1)
	colornames (red, blue)

lineWidth: integer	Width of line.
repaint: boolean (optional) (default=true)	Redraw line on map.

Return	Value
object:	A vector line object.

Example:

```
var line = map.addVectorLine("line",
[{ x: 359583, y: 406481 }, { x: 359603, y: 406471 }],
{ lineColor: "red", lineWidth: 3 });
```

Or

```
var point1 = SWPoint.create(359583, 406481);
var point2 = SWPoint.create(359603, 406471);
var point3 = SWPoint.create(359623, 406461);
var point4 = SWPoint.create(359643, 406451);

var line1 = map.addVectorLine("line1",
[point1, point2],
{ lineColor: "blue", lineWidth: 3 }, false);
```

```
var line2 = map.addVectorLine("line2",
[point3, point 4],
{ lineColor: "red", lineWidth: 3 }, true);
```

5.4.15 clearEventItem

A method to remove an event item from a map location.

Parameter	Value
id: string	Identity of object.

Example:
```
map.clearEventItem("event");
```

Or

```
eventitem.remove();
```

5.4.16 exportMap

A method that exports the map.

Parameter	Value
config (optional):	Configuration.
width: integer (optional)	Width of map in pixels.
height: integer (optional)	Height of map in pixels.

Example:
```
map.exportMap();
```

Or

```
map.exportMap(640, 480);
```

5.4.17 getCenter

getCenter is a method that returns the current map center location.

Return	Value
object:	A point set object.

Example:
```
var point = map.getCenter();
```

5.4.18 getCustomCanvas

A method that returns a custom canvas in a map location.

Parameter	Value
id: string	Identity of object.

Return	Value
object:	A custom canvas object.

Example:
```
var customCanvas = map.getCustomCanvas("canvas");
```

5.4.19 getDatasets

A method that returns the underlying datasets of the map.

Return	Value
object:	A dataset array.

Example:
```
var ds = map.getDatasets();

for (var i = 0, c = ds.length; i < c; i++)
{
   var my_id = ds[i].id;
   var my_name = ds[i].name;
   var my_visible = ds[i].visible;
   var my_scales = ds[i].scales;

   if (my_visible)
   {
      var ml = ds[i].layers;

      for (var ii = 0, cc = ml.length; ii < cc; ii++)
      {
         var my_ml_id = ml[ii].id;
         var my_ml_name = ml[ii].name;
         var my_ml_visible = ml[ii].visible;

         if (my_ml_visible)
         {
            //Your code here!
         }
      }

      //Your code here!
   }
}
```

5.4.20 getDivMarker

A method that returns a divmarker in a map location.

Parameter	Value
id: string	Identity of object.

Return	Value
object:	A divmarker object.

Example:
```
var divmarker = map.getDivMarker("hello");

var dom = divmarker.getDom();
dom.innerHTML = "Hello World";

divmarker.update(); //Update the divmarker so the location is updated.
```

5.4.21 getEventItem

A method that returns an event item in a map location.

Parameter	Value
id: string	Identity of object.

Return	Value
object:	An event item object.

Example:
```
var eventitem = map.getEventItem("event");
```

5.4.22 getMaplayers

A method that returns the underlying maplayers of the map.

Return	Value
object:	A maplayer array.

Example:
```
var ml = map.getMaplayers();

for (var i = 0, c = ml.length; i < c; i++)
{
   var my_id = ml[i].id;
   var my_name = ml[i].name;
   var my_visible = ml[i].visible;

   if (my_visible)
   {
      //Your code here!
   }
}
```

5.4.23 getMarker

A method that returns a marker in a map location.

Parameter	Value
id: string	Identity of object.

Return	Value
object:	A marker object.

Example:
```
var marker = map.getMarker("house");
```

5.4.24 getZoom

Zoom is the extent at which the map is displayed.

getZoom is a method that returns the current map zoom level.

Return	Value
integer:	Zoom level (0 = highest, 10 = lowest).

Example:
```
var zoom = map.getZoom();
```

---

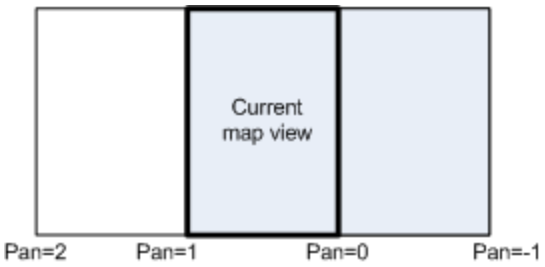
zoomIn and **zoomOut** are methods that change the zoom (increase or decrease map details) of the map to a zoom level relative to the current map center location.

5.4.25 panDir

A method that pans the map to the direction relative to the current map center location.

Parameter	Value
x: number	Direction left or right in half-screens.
y: number	Direction up or down in half-screens.

Image below shows an example of a half-screen in two half's:
When pan=0 then the center of the map is currently displayed.
When pan=1 then the map slides half-screen of the current map view to the left and becomes the new center of the map.
Half of the map was not displayed before (white box on the left) and half of the map was displayed before (light blue box in the middle).
When pan=2 then the map slides half-screen of the current map view to the left and becomes the new center of the map.
Nothing of the map was displayed before (white box on the left and another box on the left (not shown)) when pan=0.



Example:
```
map.panDir(1, 0); //Pans the center 1 left.
map.panDir(-1, 0); //Pans the center 1 right.
map.panDir(0, -1); //Pans the center 1 up.
map.panDir(0, 1); //Pans the center 1 down.
```

Or

```
var point = SWPoint.create(359583, 406481);

map.setCenter(point, map.getZoom());

map.panDir(1, 0); //Pans the center 1 left.
```

5.4.26 removeAllCustomCanvases

A method to remove all custom canvases from the map.

Example:
```
map.removeAllCustomCanvases();
```

5.4.27 removeAllDivMarkers

A method to remove all divmarkers from a map location.

Example:
```
map.removeAllDivMarkers();
```

5.4.28 removeAllMarkers

A method to remove all markers from a map location.

Example:
```
map.removeAllMarkers();
```

5.4.29 removeAllVectorCircles

A method to remove all vector circles from a map location.

Example:
```
map.removeAllVectorCircles();
```

5.4.30 removeAllVectorLines

A method to remove all vector lines from a map location.

Example:
```
map.removeAllVectorLines();
```

5.4.31 removeCustomCanvas

A method to remove a custom canvas from a map location.

Parameter	Value
id: string -or customCanvas: object	Identity of object. A custom canvas object.

Example:
```
map.removeCustomCanvas("line");
map.removeCustomCanvas(line);
```

Or

```
customCanvas.remove();
```

5.4.32 removeDivMarker

A method to remove a divmarker from a map location.

Parameter	Value
id: string -or divMarker: object	Identity of object. A divmarker object.

Example:
```
map.removeDivMarker("hello");
map.removeDivMarker(divmarker);
```

Or

```
divmarker.remove();
```

5.4.33 removeListener

A method to remove a listener background event.

Parameter	Value
type: string	A type of listener to create. See Types 16.
function: name	An asynchronous JavaScript function that implements the logic when the function call returns.

Example:
```
function zoomChange(map, zoom)
{
   //Your code here!
   //map.<method>;
}

map.removeListener("zoomchange", zoomChange);
```

5.4.34 removeMarker

A method to remove a marker from a map location.

Parameter	Value
id: string -or marker: object	Identity of object. A marker object.

Example:
```
map.removeMarker("house");
map.removeMarker(marker);
```

Or

```
marker.remove();
```

5.4.35 removeVector

This method has been deprecated use **removeVectorLine** instead.

5.4.36 removeVectorCircle

A method to remove a vector circle from a map location.

Parameter	Value
id: string -or vectorCircle: object	Identity of object. A vector circle object.

Example:
```
map.removeVectorCircle("circle");
map.removeVectorCircle(circle);
```

Or

```
circle.remove();
```

5.4.37 removeVectorLine

A method to remove a vector line from a map location.

Parameter	Value
id: string -or vectorLine: object	Identity of object. A vector line object.

Example:
```
map.removeVectorLine("line");
map.removeVectorLine(line);
```

Or

```
line.remove();
```

5.4.38 repaint

A method that repaints the map.

Example:
```
map.repaint();
```

5.4.39 route

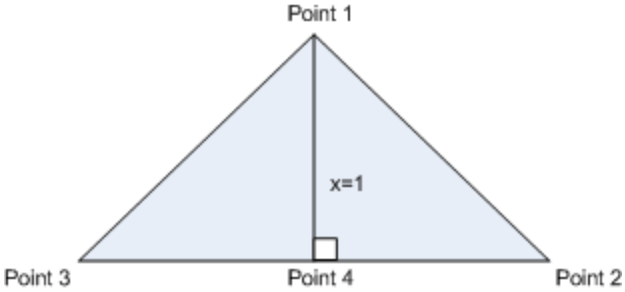
A method to set a route from one map location to another map location.

Parameter	Value
fromX: number	isn93 x-coordinate of start location.
fromY: number	isn93 y-coordinate of start location.
toX: number	isn93 x-coordinate of end location.
toY: number	isn93 y-coordinate of end location.
searchDistance: integer (optional)	Radius of area from point in meters to the street.
simplify: integer (optional)	Line simplicity in meters (0 = nothing is simple, n = By how much can the line from x be slided so that points can be removed). Simplification removes extraneous bends and small intrusions and extrusions from a line or polygon boundary without destroying its essential shape. It is efficient for data compression and for eliminating redundant details. The line that results may contain sharp angles and spikes that reduce the cartographic quality of the line. Use for small amounts of data reduction or compression and when high cartographic quality is not needed.
function: name	An asynchronous JavaScript function that implements the logic when the function call returns.
scope: name (optional)	Owner of the current object.

Image below to the right: Simplify=1 then Point 1 is not removed from the route. Simplify=2 then the distance x = 1 from the line (from imaginary Point 4) between

Point 2 and 3 to the Point 1 is < 2 then Point 1 is removed from the route.

Image below to the left: Yellow line is more simplified than the green line and thus results in less detail of the route.



Example:

```
function rResult(map, res)
{
   var r = res.OK; //Search success (true) or failure (false).

   map.addVectorLine("line",
   res.data[0].points,
   { color: "blue", size: 3 });

   var dir = res.data[0].direction;
   var b = dir[0].b; //b = bearing (B = straight ahead, H = to the right, V
   = to the left).
   var n = dir[0].n; //n = streetname.
   var x = dir[0].p[0].x; //x-coordinate.
   var y = dir[0].p[0].y; //y-coordinate.
   var d = dir[0].d; //distance in meters.
   var t = dir[0].t; //time in minutes.

   var tt = res.data[0].time; //Total time in minutes.
   var td = res.data[0].distance; //Total distance length in meters.
}

map.route(358455, 407655, 358852, 407286, 100, 0, rResult, this); //Short.
```

Or

```
map.route(358455, 407655, 540637, 575896, 100, 2, rResult, this); //Long.
```

5.4.40 routeCheckPoint

A method to set a route check point to a map location.

Parameter	Value
x: number	isn93 x-coordinate of start location.

Parameter	Value
y: number	isn93 y-coordinate of start location.
searchDistance: integer (optional)	Radius of area from point in meters to the street.
function: name	An asynchronous JavaScript function that implements the logic when the function call returns.
scope: name (optional)	Owner of the current object.

Example:

```
function rResult(map, res)
{
   var r = res.OK; //Search success (true) or failure (false).
}

var searchRadius = 100;

map.routeCheckPoint(358455, 407655, searchRadius, rResult, this);
```

5.4.41 search

A method to search for addresses on the map.

Please note that the page has to be UTF-8 encoded for the server if using non-English characters.

Parameter	Value
searchString: string	Search string.
function: name	An asynchronous JavaScript function that implements the logic when the function call returns.
scope: name (optional)	Owner of the current object.

Laugavegur 32 Search

Laugavegur 32, 101 Reykjavik point(357560,408003)

Example:

```
function sResult(map, res)
{
   var txt = "";

   var r = res.OK; //Search success (true) or failure (false).

   for (var i = 0, c = res.data.length; i < c; i++)
   {
      //n = streetname, x = x-coordinate, y = y-coordinate.
      txt = txt +
         res.data[i].n + " point(" + res.data[i].p[0].x + ", " +
         res.data[i].p[0].y + ")";

      //c = category (1 = address, 4 = city).
      //var cat = res.data[i].c;

      map.setCenter( {res.data[i].p[0].x, res.data[i].p[0].y}, 10);
   }

   //Do something with the txt object!
}

var text = 'Laugavegur 32';

map.search(text, sResult, this);
```

5.4.42 setCenter

setCenter is a method that sets the map center location to map coordinates and zoom level and does a **zoomIn** or **zoomOut** depending on the previous zoom level.

Parameter	Value
point: a point set object(s) (**SWPoint**) -or- point: a point set array	isn93 map point locations.
zoom: integer (optional)	Zoom level (0 = highest, 10 = lowest).

Example:

```
var point = SWPoint.create(359583, 406481);

map.setCenter(point, 5);
```

Or

```
var point = SWPoint.create(359583, 406481);

map.setCenter(point, map.getZoom());
```

Or to zoom in to many map coordinates around the centermost map coordinate

```
map.setCenter([{ x: 359583, y: 406481 }, { x: 359603, y: 406471 }]);
```

Or

```
var point1 = SWPoint.create(359583, 406481);
var point2 = SWPoint.create(359603, 406471);
var point3 = SWPoint.create(359623, 406461);
var point4 = SWPoint.create(359643, 406451);

map.setCenter([point1, point2, point3, point4]);
```

5.4.43 setDataset

A method to set the active dataset for the map.

Parameter	Value
name: string	Name of object.

Example:

```
map.setDataset("my_dataset_name");
```

5.4.44 setDataset

A method to set the active dataset for the map.

Parameter	Value
id: string	Identity of object.

Example:
```
map.setDataset("my_dataset_identity");
```

5.4.45 setMaplayer

A method to show/hide a maplayer on the map.

Parameter	Value
name: string	Name of object.
show: boolean	Show the maplayer on the map.
opacity: number (optional)	Transparency of the maplayer on the map (0 = No transparency, 1 = Full transparency).

Example:
```
map.setMaplayer("my_maplayer_name", false, 0.5);
```

5.4.46 setMaplayer

A method to show/hide a maplayer on the map.

Parameter	Value
id: string	Identity of object.
show: boolean	Show the maplayer on the map.
opacity: number (optional)	Transparency of the maplayer on the map (0 = No transparency, 1 = Full transparency).

Example:
```
map.setMaplayer("my_maplayer_identity", false, 0.5);
```

5.4.47 setZoom

Zoom is the extent at which the map is displayed.

setZoom is a method that sets the map zoom level and does a **zoomIn** or **zoomOut** depending on the previous zoom level.

Parameter	Value
zoom: integer	Zoom level (0 = highest, 10 = lowest).

Example:
```
map.setZoom(5);
```

---

zoomIn and **zoomOut** are methods that change the zoom (increase or decrease map

details) of the map to a zoom level relative to the current map center location.

5.4.48 showMyLocation

A method that shows your current GPS location on the map.

Parameter	Value
showMyLocation: boolean	Track your GPS location on the map.

Example:
```
map.showMyLocation(true);
```

Or

Parameter	Value
showOnStart: boolean (default=false) (optional)	Show GPS location when map is loaded.
setView: boolean (default=false) (optional)	Go to the current GPS location on the map.
errorFunction: name (optional) Example: function errorResult(e) { alert(e); }	An asynchronous JavaScript function that implements the logic when the function call returns. ERROR CODE: UNKNOWN (0). PERMISSION_DENIED (1) if the user clicks that “Don’t Share” button or otherwise denies you access to their location. POSITION_UNAVAILABLE (2) if the network is down or the positioning satellites can’t be contacted. TIMEOUT (3) if the network is up but it takes too long to calculate the user’s position.
accuracy: integer (default=1000001) (optional)	GPS accuracy in meters.
options: (optional):	HTML5 Geolocation options.
enableHighAccuracy: boolean (default=true)	The enableHighAccuracy property is exactly what it sounds like. If true, and the device can support it, and the user consents to sharing their exact location, then the device will try to provide it. Both iPhones and Android phones have separate permissions for low- and high-accuracy positioning, so it is possible that enableHighAccuracy: true will fail, but calling with enableHighAccuracy: false would succeed.
timeout: integer (default=20000)	The timeout property is the number of milliseconds your web application is willing to wait for a position. This timer doesn’t start counting down until after the user gives permission to even try to calculate their position. You’re not timing the user; you’re timing the network.
maximumAge: integer (default=2000)	The maximumAge property allows the device to answer immediately with a cached position.

Parameter	Value
marker: object (optional):	A marker object. Since a marker has a point parameter it will be discarded and your current GPS location used instead.
vectorCircle: object (optional):	A vector circle object.

Example:

```
map.showMyLocation(
{ showOnStart: true, setView: true, errorFunction: function errorResult(e)
{ alert(e); }
});
```

Part

6 Examples

A collection of usable examples to use in your project.

See the examples web site for detailed and helpful examples.