Geoconcept LBS Platform : HTC JavaScript API Documentation

GEOCONCEPT

1. Audience

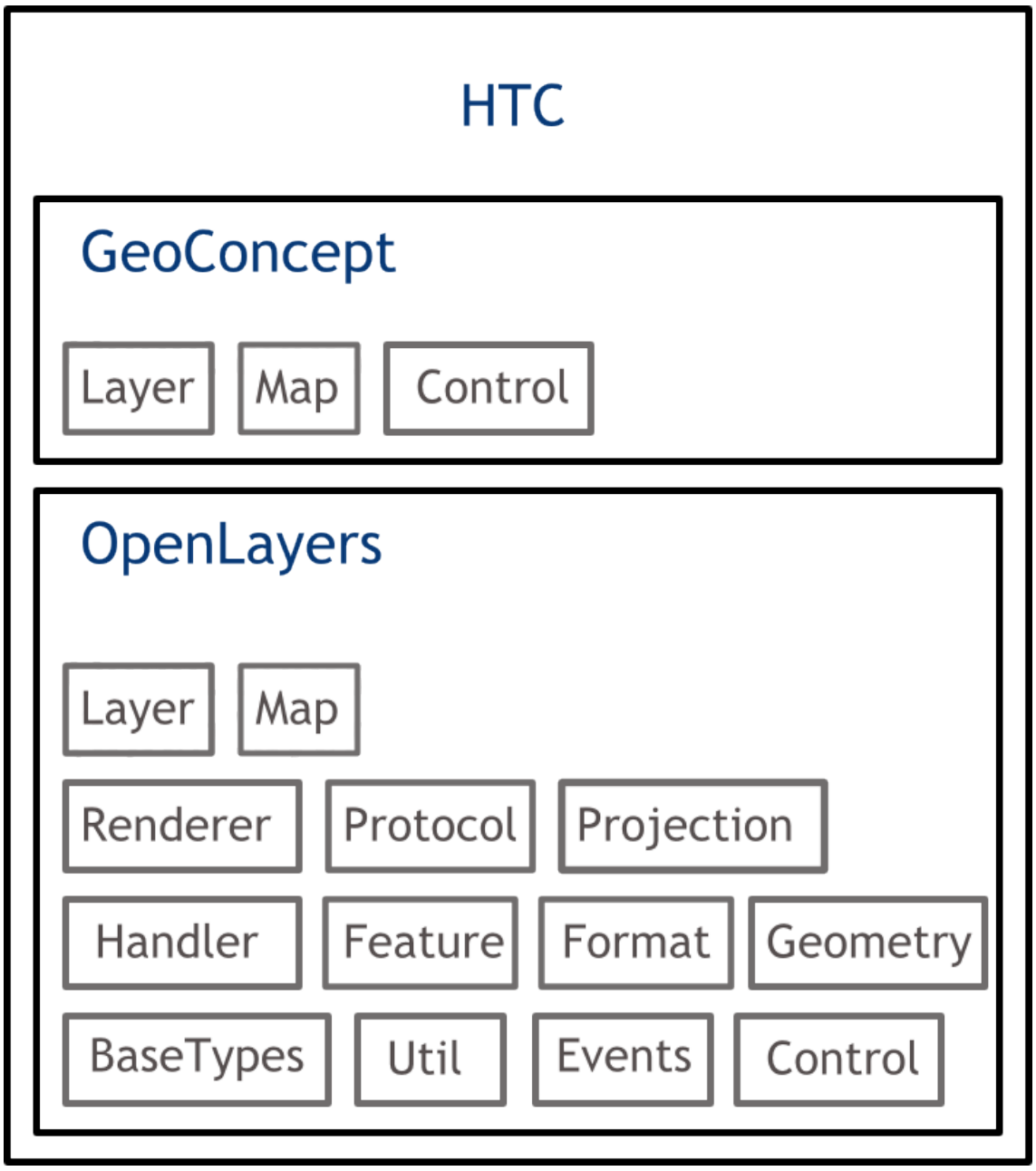
This documentation is designed for people familiar with *JavaScript* [http://www.ecma-international.org/publications/standards/Ecma-262.htm] programming and object-oriented programming concepts. There are many *JavaScript tutorials* [http://www.google.com/search?q=javascript+tutorials] available on the Web.

2. Introduction

The **HTC** API lets you embed interactive Geoconcept maps in your own web pages with JavaScript.

HTC is the name for High Traffic Client, a web client designed especially for web sites with a heavy traffic. This DHTML/JavaScript client can be used to develop very reactive user interfaces and is compatible with all browsers (no ActiveX, Java…).

The **HTC** API is based over the **OpenLayers** JavaScript library (Release 2.13.1, see: *openlayers.org* [http://openlayers.org/]). HTC uses a subset of OpenLayers classes and adds a set of Geoconcept classes.



2.1. Using HTC with OpenLayers

If you already use the OpenLayers library, just include in your web page the htc-lite.js file (in the /lib folder) after OpenLayers.js and call the `GCUI.Layer.GeoConcept` class to create a new Geoconcept HTC layer. Then, you can add it to your OpenLayers map object (with the map `addLayer` method like other OpenLayers layers).

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="viewport"
        content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-
scalable=0">
<meta name="apple-mobile-web-app-capable" content="yes">
<title>Layer : Geoconcept</title>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<style>
.smallmap {
        width: 512px;
        height: 256px;
        border: 1px solid #ccc;
}

h1 {
        color: #003a6b;
        background-color: transparent;
        font: 100% 'Lucida Grande', Verdana, Geneva, Lucida, Arial, Helvetica,
                sans-serif;
        margin: 0;
        padding-top: 0.5em;
}
</style>
<script src="http://api.geoconcept.net/htc/OpenLayers.js"></script>
<script src="http://api.geoconcept.net/htc/htc-lite.js"></script>
<script>
        function init() {
                var map = new OpenLayers.Map('map', {
                        projection : new OpenLayers.Projection("EPSG:27582"),
                        maxExtent : new OpenLayers.Bounds(5000, 1620000, 1198000,
 2678000)
                });
                map.addControl(new OpenLayers.Control.LayerSwitcher());

                var urls = [ "http://gcweb.geoconcept.com/gws/wmts" ];
                var layer = new GCUI.Layer.GeoConcept("GeoConcept Layer", urls, {
                        layer : 'France'
                });
```

```
                layer.events.on({
                        'init' : function() {
                                map.zoomToMaxExtent();
                        }
                });
                map.addLayer(layer);

                var wms = new OpenLayers.Layer.WMS("Géosignal WMS",
                                "http://www.geosignal.org/cgi-bin/wmsmap?", {
                                        layers : "Regions,Departements"
                                });
                map.addLayer(wms);

        }
</script>
</head>
<body onload="init()">
        <h1 id="title">Geoconcept HTC (High Traffic Client) Example</h1>
        <p id="shortdesc">Demonstrates a Geoconcept HTC layer that loads
                tiles from a web accessible disk-based cache.</p>
        <div id="map" class="smallmap"></div>
</body>
</html>
```

See this example [examples/layer-geoconcept.html]

2.2. OpenLayers classes in HTC

HTC includes a subset of OpenLayers classes (in the following order) :

- SingleFile.js
- BaseTypes/Class.js
- Util.js
- BaseTypes/Bounds.js
- BaseTypes/Element.js
- BaseTypes/LonLat.js
- BaseTypes/Pixel.js
- BaseTypes/Size.js
- BaseTypes.js
- Console.js
- Lang.js
- Events.js
- Control.js
- Handler.js
- Handler/Box.js

- Handler/MouseWheel.js
- Handler/Click.js
- Handler/Drag.js
- Control/DragPan.js
- Control/ZoomBox.js
- Control/Navigation.js
- Control/PanZoom.js
- Control/ArgParser.js
- Control/Attribution.js
- Tween.js
- Projection.js
- Map.js
- Layer.js
- Renderer.js
- Popup.js
- Format.js
- Geometry.js
- Geometry/Point.js
- Geometry/Collection.js
- Geometry/MultiPoint.js
- Geometry/MultiLineString.js
- Geometry/Curve.js
- Geometry/LineString.js
- Geometry/LinearRing.js
- Geometry/Polygon.js
- Geometry/MultiPolygon.js
- Feature.js
- Feature/Vector.js
- Style.js
- StyleMap.js
- Renderer/Canvas.js
- Renderer/Elements.js
- Renderer/SVG.js
- Renderer/VML.js
- Layer/Vector.js
- Format/WKT.js
- Protocol.js

- Format/JSON.js
- Format/GeoJSON.js
- Protocol/Script.js
- Tile.js
- Tile/Image.js
- Layer/HTTPRequest.js
- Layer/Grid.js
- Layer/Markers.js
- Control/DrawFeature.js
- Control/Measure.js
- Handler/Point.js
- Handler/Path.js
- Handler/Polygon.js
- Handler/RegularPolygon.js
- Icon.js
- Marker.js
- Handler/Feature.js
- Control/SelectFeature.js
- Popup/Anchored.js
- Popup/Framed.js
- Popup/FramedCloud.js
- Filter.js
- Filter/Comparison.js
- Rule.js
- Handler/Keyboard.js
- Control/DragFeature.js
- Control/ModifyFeature.js
- Handler/Pinch.js
- Control/PinchZoom.js
- Handler/Hover.js

3. Getting started

3.1. Setting an app id and a token

If the access to Geoconcept Web services is protected, you need to specify your application id and token to use the API.

Before creating your map, specify your credentials using this JavaScript code :

```
GCUI.Settings = {app_id:"REPLACE_WITH_YOUR_APP_ID",
 token:"REPLACE_WITH_YOUR_TOKEN"};
```

3.2. Creating a map

This first code example shows how to create a basic map in a HTML page.

```
<!DOCTYPE html>
<html>
<head>
<title>Basic map</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}
</style>
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France'
                };
                var map = new GCUI.Map('map', options);
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 100%; height: 100%;"></div>
</body>
</html>
```

Let’s analyze this simple code :

1. Include the htc.js API using a script tag
2. Add a `<div>` with an identifier. This block will contain the map.
3. Create a map object using the *GCUI.Map* [../../../../../jsdoc/files/GeoConcept/Map-js.html#GCUI.Map.GCUI.Map] constructor (with the div identifier and map options).
4. Initialize the map object from the body tag's onload event.

See this example [examples/basic-map.html]

3.3. Map options

Option	Type	Description
server	String	Url of the LBS Platform tile server
mapName	String	Name of the GeoConcept map (file *.gcm)
tab	String	Name of the visibility *[tab]* of the map
x	Float	X-coordinate of the map center
y	Float	Y-coordinate of the map center
scale	Integer	Logical *[scale]* of the map
showSlider	Boolean	Display or not the zoom slider (default is true)

The following code example creates a map centered on Paris (using x, y *[coordinates]* and scale options) :

```
<!DOCTYPE html>
<html>
<head>
<title>Basic map : center</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}
</style>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France',
                        x : 601106,
```

```
                        y : 2428909,
                        scale : 6
                };
                var map = new GCUI.Map('map', options);
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 100%; height: 100%;"></div>
</body>
</html>
```

This example declares an HTML5 doctype :

```
<!DOCTYPE html>
```

Note that if you want to use percentage-based sizes in this standard mode, you must include the following <style> declaration :

```
<style type="text/css">
  html { height: 100% }
  body { height: 100%; margin: 0; padding: 0 }
</style>
```

The code shows also how to use CSS files to specify the style of the DOM elements of the map (for example the gcgrey1.css file renders a grey zoom slider) :

```
<link type="text/css" rel="stylesheet" href="../scripts/htc/skins/htc.css">
<link type="text/css" rel="stylesheet" href="../scripts/htc/skins/sliders/
gcgrey1.css">
```

See this example [examples/basic-map-center.html]

4. Map

4.1. Getting a map

After creating a map, you can get the map object from its id :

```
function init() {
    ...
    var map = new GCUI.Map('map', options);
    ...
}
function doSomethingWithMap() {
    var map = GCUI.getMap('map');
    ...
}
```

4.2. Loading the map

The *GCUI.Map* [../../../../../jsdoc/files/GeoConcept/Map-js.html#GCUI.Map.GCUI.Map] constructor with *`server`*, *`mapName`* and *`tab`* options arguments gets asynchronously map informations (extent, resolutions, ...) from the LBS Platform server. The map events system trigger a *load* event to notify that the map is loaded :

```
<!DOCTYPE html>
<html>
<head>
<title>Basic map : load</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}
</style>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France',
                        x : 601106,
                        y : 2428909,
```

```
                        scale : 6
                };
                var map = new GCUI.Map('map', options);
                map.onEvent("load", onMapLoaded);
        }
        function onMapLoaded() {
                var map = GCUI.getMap('map');
                alert("Map is loaded. Map extent is : " + map.getExtent());
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 100%; height: 100%;"></div>
</body>
</html>
```

The following code registers a callback function that will be called when map is ready :

```
map.onEvent("load",onMapLoaded);
```

In this example, the callback function display the map extent :

```
function onMapLoaded() {
    var map = GCUI.getMap('map');
    alert("Map is loaded. Map extent is : "+ map.getExtent());
}
```

See this example [examples/basic-map-load.html]

4.3. Zooming the map

To get the current zoom level of the map, use :

```
var zoomLevel = map.getZoom();
```

Note : The OpenLayers zoom level **0** corresponds to a map fully zoomed out and the zoom level **11** corresponds to a map fully zoomed in. It's the reverse of the Geoconcept logical scales (in Geoconcept, logical *[scale]* **12** corresponds to a map fully zoomed out and logical *[scale]* **1** corresponds to a map fully zoomed in).

To get the current Geoconcept logical *[scale]* of the map, use :

```
var logicalScale = map.getLogicalScale();
```

So, to convert Geoconcept logical *[scale]* to OpenLayers zoom level :

```
var zoomLevel = map.getNumZoomLevels() - logicalScale;
```

You can specify the minimum (default value is 1) and maximum (default value is 12) logical scales of the map :

```
map.setMinLogicalScale(3);
map.setMaxLogicalScale(10);
```

Several API methods (from OpenLayers API) allow to modify the zoom level of the map :

- `zoomTo` : Zoom to a specific zoom level
- `zoomIn` : Zoom in
- `zoomOut` : Zoom out
- `zoomToExtent` : Zoom to a specific bounding-box
- `zoomToMaxExtent` : Zoom to the map extent

```
<!DOCTYPE html>
<html>
<head>
<title>Basic map : zoom</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}

.btn {
        float: left;
        background-color: rgba(0, 60, 136, 0.701961);
        width: auto;
        height: 22px;
        color: white;
        margin: 2px;
        padding: 2px;
        text-align: center;
        cursor: pointer;
        border: 1px solid #ccc;
        text-align: center;
}
</style>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
```

```
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France',
                        x : 601106,
                        y : 2428909,
                        scale : 6
                };
                var map = new GCUI.Map('map', options);
        }
        function zoomIn() {
                var map = GCUI.getMap('map');
                map.zoomIn();
        }
        function zoomOut() {
                var map = GCUI.getMap('map');
                map.zoomOut();
        }
        function zoomTo(z) {
                var map = GCUI.getMap('map');
                map.zoomTo(z);
        }
        function zoomExtent() {
                var map = GCUI.getMap('map');
                map
                                .zoomToExtent(new OpenLayers.Bounds(570173, 2404000,
 635023,
                                                2451600));
        }
        function zoomMapExtent() {
                var map = GCUI.getMap('map');
                map.zoomToMaxExtent();
        }
</script>
</head>
<body onload="init()">
        <div style="position: absolute; z-index: 5000; left: 35px;">
                <div class="btn" onclick="zoomIn()">zoom in</div>
                <div class="btn" onclick="zoomOut()">zoom out</div>
                <div class="btn" onclick="zoomTo(0)">zoom to 0</div>
                <div class="btn" onclick="zoomTo(11)">zoom to 11</div>
                <div class="btn" onclick="zoomExtent()">zoom to bounds</div>
                <div class="btn" onclick="zoomMapExtent()">map extent</div>
        </div>
        <div id="map" style="width: 100%; height: 100%;"></div>
</body>
</html>
```

See this example [examples/basic-map-zoom.html]

4.4. Centering the map

To get the current center of the map, use :

```
var center = map.getCenter();
```

To change the center of the map, use the `setCenter` method :

```
map.setCenter(new OpenLayers.LonLat(601000,2428657));
```

4.5. Sizing the map

To get the current size (in pixels) of the map div, use :

```
var size = map.getSize();
```

To modify the map size, use the `setSize` method :

```
map.setSize('80%','80%');
```

or

```
map.setSize('400px','400px');
```

Code example :

```
<!DOCTYPE html>
<html>
<head>
<title>Basic map : size</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}
</style>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
```

```
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France',
                        x : 601106,
                        y : 2428909,
                        scale : 6
                };
                var map = new GCUI.Map('map', options);
                map.onEvent("load", onMapLoaded);
        }
        function onMapLoaded() {
                var map = GCUI.getMap('map');
                map.setSize('80%', '80%');
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 100%; height: 100%;"></div>
</body>
</html>
```

See this example [examples/basic-map-size.html]

You can also ask to maximize the map specifying a right margin and bottom margin :

```
<!DOCTYPE html>
<html>
<head>
<title>Basic map : maximize</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}
</style>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France',
                        x : 601106,
                        y : 2428909,
```

```
                        scale : 6
                };
                var map = new GCUI.Map('map', options);
                map.onEvent("load", onMapLoaded);
        }
        function onMapLoaded() {
                var map = GCUI.getMap('map');
                map.maximize(100, 50);
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 100%; height: 100%;"></div>
</body>
</html>
```

See this example [examples/basic-map-maximize.html]

5. Control

5.1. Adding a control to the map

To add a control to the map, use the `addControl` method :

```
map.addControl(mycontrol);
```

5.2. Graphical scale

Create a `GCUI.Control.GraphicScale` control and add it to the map to display a graphical scale.

```
<!DOCTYPE html>
<html>
<head>
<title>Control : graphical scale</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}
</style>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France',
                        x : 601106,
                        y : 2428909,
                        scale : 6
                };
                var map = new GCUI.Map('map', options);
                map.addControl(new GCUI.Control.GraphicScale({
                        posx : 0,
                        posy : -1
                }));
        }
</script>
</head>
<body onload="init()">
```

```
        <div id="map" style="width: 100%; height: 100%;"></div>
</body>
</html>
```

See this example [examples/control-graphicalScale.html]

5.3. Zoom slider

Create a `GCUI.Control.ScaleSlider` control and add it to the map to display a scale slider.

```
<!DOCTYPE html>
<html>
<head>
<title>Control : scale slider</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}
</style>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France',
                        x : 601106,
                        y : 2428909,
                        scale : 6,
                        showSlider : false
                // by default, GCUI.Map add a scale slider control
                };
                var map = new GCUI.Map('map', options);
                map.addControl(new GCUI.Control.ScaleSlider());
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 100%; height: 100%;"></div>
</body>
</html>
```

See this example [examples/control-scaleSlider.html]

5.4. Overview map

Create a `GCUI.Control.GlobalView` control and add it to the map to display an overview map control.

```
<!DOCTYPE html>
<html>
<head>
<title>Control : overview map</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}

.olAlphaImg {
        display: none;
}

.olControlOverviewMap {
        right: 0px;
}

.olControlOverviewMapExtentRectangle {
        opacity: .3;
        filter: alpha(opacity = 30);
        background-color: black;
        border: solid 2px yellow
}
</style>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France',
                        x : 601106,
                        y : 2428909,
                        scale : 6
                };
                var map = new GCUI.Map('map', options);
```

```
                map.onEvent('load', onMapLoaded);
        }
        function onMapLoaded() {
                var map = GCUI.getMap('map');
                var ovlayer = new GCUI.Layer.GeoConcept("ovLayer",
                                'http://gcweb.geoconcept.com/gws/wmts', {
                                        layer : '@Minimap@France',
                                        eventListeners : {
                                                "init" : function() {
                                                        var overview = new
 OpenLayers.Control.OverviewMap({
                                                                maximized : true,
                                                                size : {
                                                                        w : 150,
                                                                        h : 150
                                                                },
                                                                minRatio : 10,
                                                                maxRatio : 10,
                                                                layers : [ ovlayer ]
                                                        });
                                                        map.addControl(overview);
                                                }
                                        }
                                });
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 100%; height: 100%;"></div>
</body>
</html>
```

See this example [examples/control-overviewMap.html]

5.5. Layer switcher

Create a `GCUI.Control.LayerSwitcher` control and add it to the map to display a layer switcher control.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Control : layer switcher</title>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<style>
html {
        height: 100%;
}
```

```
body {
        height: 100%;
        margin: 0;
        padding: 0;
}

.mapGroupInfo {
        width: 100%;
}

.mapLayerInfo {
        width: 100%;
}
</style>
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France'
                };
                var map = new GCUI.Map('map', options);
                map.onEvent('load', onMapLoaded);
        }
        function onMapLoaded() {
                var map = GCUI.getMap('map');
                var ls = new GCUI.Control.LayerSwitcher({
                        div : document.getElementById('layers')
                });
                map.addControl(ls);
                var json = {
                        layers : [ {
                                name : "Groupe B",
                                internalName : "null",
                                depth : 0,
                                expanded : false
                        }, {
                                label : "Vues aériennes",
                                name : "ORTHO",
                                depth : 1,
                                opacity : 100,
                                visibility : 0
                        }, {
                                label : "Fond de carte",
                                name : "France",
                                depth : 1,
                                opacity : 100,
                                visibility : 0
```

```
                        }, {
                                name : "Groupe A",
                                internalName : "null",
                                depth : 0,
                                expanded : true
                        }, {
                                label : "Vues aériennes",
                                name : "ORTHO",
                                depth : 1,
                                opacity : 100,
                                visibility : 0
                        }, {
                                label : "Fond de carte",
                                name : "France",
                                depth : 1,
                                opacity : 100,
                                visibility : 1
                        } ]
                };
                ls.initFromJson(json);
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 70%; height: 100%;"></div>
        <div id="layers"
                style="position: absolute; top: 0px; width: 30%; right: 0px;"></div>
</body>
</html>
```

See this example [examples/control-layerSwitcher.html]

5.6. Getting map object information from Geoconcept

Create a `GCUI.Control.GeoConceptGetFeatureInfo` control and add it to the map to get information from objects in the Geoconcept map.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="viewport"
        content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-
scalable=0">
<meta name="apple-mobile-web-app-capable" content="yes">
<title>Control : Geoconcept getFeatureInfo</title>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
```

```
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        var map, popups = {};
        var control = null;

        function init() {
                map = new GCUI.Map('map', {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'DemoThematic1'
                });

                updateControl();

        }

        function addControl(h, infobox) {
                if (control) {
                        control.destroy();
                }
                control = new GCUI.Control.GeoConceptGetFeatureInfo(
                                {
                                        hover : h,
                                        //queryVisible: true,
                                        infoMode : infobox ? 'infoboxfields' :
 'sheetfields',
                                        eventListeners : {
                                                getfeatureinfo : function(evt) {
                                                        var text = '';
                                                        if (evt.text) {
                                                                if ((typeof
 evt.text) == 'string') {
                                                                        text =
 evt.text + "<br>";
                                                                } else {
                                                                        for ( var
 field in evt.text) {
                                                                                if
 (/^(ftp|http|https):\/\/.*/

            .test(evt.text[field])) {

    text += "<a target='blank' href='"+evt.text[field]+"'>"

                    + field + '</a><br>';
                                                                                }
 else {

    text += field + ' : '
```

```
                    + evt.text[field] + '<br>';
                                                                                }
                                                                            }
                                                                        }
                                                                } else {
                                                                        text = "No feature
 in that area.<br>";
                                                                }
                                                                if (text != '') {
                                                                        var popupId =
 evt.xy.x + "," + evt.xy.y;
                                                                        var popup =
 popups[popupId];
                                                                        if (!popup || !
popup.map) {
                                                                                popup = new
 GCUI.Popup.Anchored(popupId,

    map.getLonLatFromPixel(evt.xy),

    null, " ", true);

 popup.autoSize = true;

 popups[popupId] = popup;

 map.addPopup(popup, true);
                                                                        }

 popup.setContentHTML(popup.contentHTML + text);
                                                                }
                                                        },
                                                        exception : function(evt) {

 GCUI.Console.error(evt.request.statusText);
                                                        }
                                                }
                                        });
                        map.addControl(control);
                        control.activate();
        }

        function updateControl() {
                addControl(document.getElementById("hover").checked, document
                                .getElementById("infobox").checked);
        }
</script>
</head>
```

```
<body onload="init()">
        <div id="map" class="smallmap" style='width: 800px; height: 600px;'></div>
        <ul id="control">
                <li><input type="radio" name="controlType" value="click"
                        id="click" onclick="updateControl();" checked="checked" />
 <label
                        for="click">Click</label></li>
                <li><input type="radio" name="controlType" value="hover"
                        id="hover" onclick="updateControl();" /> <label
 for="hover">Hover</label>
                </li>
        </ul>
        <ul>
                <li><input type="radio" name="controlMode" value="infobox"
                        id="infobox" onclick="updateControl();" checked="checked" />
 <label
                        for="infobox">infobox</label></li>
                <li><input type="radio" name="controlMode" value="sheet"
                        id="sheet" onclick="updateControl();" /> <label
 for="sheet">sheet</label>
                </li>
        </ul>

</body>
</html>
```

See this example [examples/control-getFeatureInfo.html]

5.7. Selecting map objects from Geoconcept

Create a `GCUI.Control.GeoConceptSelectFeature` control and add it to the map to select objects in the Geoconcept map.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="viewport"
        content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-
scalable=0">
<meta name="apple-mobile-web-app-capable" content="yes">
<title>Control : select features</title>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">

<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        var server = 'http://gcweb.geoconcept.com/gws/wmts';
        var layer = 'DemoThematic1';
```

```
        var control;
        function init() {
                var options = {
                        server : server,
                        layer : layer,
                };
                var map = new GCUI.Map('map', options);
                map.onEvent("load", onMapLoaded);
                map.addControl(new GCUI.Control.LayerSwitcher());

        }

        function onMapLoaded() {
                var map = GCUI.getMap('map');

                var layer = new GCUI.Layer.GeoConcept("Selection", server, {
                        layer : "DemoThematic2",
                        singleTile : true,
                        isBaseLayer : false,
                        transparent : true
                });
                map.addLayer(layer);

                control = new GCUI.Control.GeoConceptSelectFeature({
                        drillDown : true,
                        dataFields : [ 'Type', 'Sous-type', 'id', 'Nom' ],
                        layers : [ layer ],
                        eventListeners : {
                                featureselected : function(evt) {
                                        var selectionData = evt.text;
                                        var format = new OpenLayers.Format.JSON();
                                        var json = format.write(selectionData,
 true);
                                        document.getElementById('output').value =
 json;
                                        var info = '<table>';
                                        for ( var i in selectionData[0]) {
                                                info += '<tr><td>' + i + '</td><td>'
                                                                + selectionData[0]
 [i] + '</td></tr>';
                                        }
                                        info += '</table>';

                                        if (evt.xy && evt.text != '') {
                                                var popup = new
 GCUI.Popup.Anchored("myPopupId", map

 .getLonLatFromPixel(evt.xy), null, info, true);
                                                popup.autoSize = true;
```

```
                                                map.addPopup(popup, true);
                                        }
                                }
                        },
                        selectOperation : 'replace',
                        //strictInclusion: false,
                        //box:true,
                        //polygon : true,
                        // for selecting by ids :
                        type : "PYR",
                        subtype : 'DEPT',
                        keyField : 'id'
                });
                map.addControl(control);
                control.activate();
                map.zoomToMaxExtent();
                //control.events.on({'zoomOnSelection':function(o)
{console.log(o)}});
        }
        function selectByIds() {

 control.selectByIds(document.getElementById('myids').value.split(','));
        }
</script>
</head>
<body onload="init()">
        <div id="map" class="smallmap" style='width: 800px; height: 600px;'></div>
        <div id="selectedObjectsList"></div>
        <div id="buttons">
                <button onclick="control.clearSelection()">Clear</button>
                <button onclick="control.zoomOnSelection()">Zoom</button>
                <input id="myids" type="text" value="29,35">
                <button onclick="selectByIds()">Select</button>
                <br>
                <textarea id='output' rows="6" cols="50"></textarea>
        </div>
</body>
</html>
```

See this example [examples/control-selectFeature.html]

By default, this control selects by point (on map simple click). You can select by rectangle setting the *box* option to `true`. You can select by polygon setting the *polygon* option to `true`.

5.8. Printing

Create a `GCUI.Control.Print` control and add it to the map to print a map.

```
<!DOCTYPE html>
```

```
<html>
<head>
<title>Control : print</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}

.btn {
        float: left;
        background-color: rgba(0, 60, 136, 0.701961);
        width: auto;
        height: 22px;
        color: white;
        margin: 2px;
        padding: 2px;
        text-align: center;
        cursor: pointer;
        border: 1px solid #ccc;
        text-align: center;
}
</style>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        var print;
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France'
                };
                var map = new GCUI.Map('map', options);
                print = new GCUI.Control.Print();
                map.addControl(print);
                print.activate();
        }
</script>
</head>
<body onload="init()">
        <div style="position: absolute; z-index: 5000; left: 25px;">
                <div class="btn" onclick="print.print()">Print the map</div>
        </div>
```

```
        <div id="map" style="width: 500px; height: 300px;"></div>
</body>
</html>
```

See this example [examples/control-print.html]

5.9. Geocoding

Create a `GCUI.Control.Geocode` control and add it to the map to geocode an address.

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<title>Control : geocode</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}

.odd {
        background-color: #F5F5F5;
}
</style>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        var geoc;
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France',
                        x : 601106,
                        y : 2428909,
                        scale : 6
                };
                var map = new GCUI.Map('map', options);

                geoc = new GCUI.Control.GeoCode();
                map.addControl(geoc);
        }
```

```
        function geocode() {
                geoc.geocode({
                        addressLine : document.getElementById('address').value,
                        postalCode : document.getElementById('postalCode').value,
                        city : document.getElementById('city').value,
                        callback : function(resp) {
                                document.getElementById('results').innerHTML = '';
                                for (var i = 0, l = resp.geocodedAddresses.length; i
 < l; i++) {
                                        var add = resp.geocodedAddresses[i];
                                        var div = document.createElement('div');
                                        if (i % 2 == 0) {
                                                div.className = 'odd';
                                        }
                                        div.innerHTML = add.addressLine + ' ' +
 add.postalCode
                                                        + ' ' + add.city;
                                        div.onclick =
 OpenLayers.Function.bind(function() {
                                                GCUI.getMap('map').setCenter(
                                                                new
 OpenLayers.LonLat(this.x, this.y));
                                        }, add);

 document.getElementById('results').appendChild(div);
                                }
                        },
                        url : 'http://gcweb.geoconcept.com/gws/api/lbs/geocode/json'
                });
        }
</script>
</head>
<body onload="init()">
        <form>
                Address : <input type="text" id="address"> Postal code : <input
                        type="text" id="postalCode"> City : <input type="text"
                        id="city"> <input type="button" onclick="geocode()"
                        value="ok">
        </form>
        <div id="results"
                style="position: absolute; width: 30%; cursor: pointer;"></div>
        <div id="map"
                style="position: absolute; right: 0; width: 70%; height: 90%;"></
div>
</body>
</html>
```

See this example [examples/control-geocode.html]

5.10. Routing

Create a `GCUI.Control.Route` control and add it to the map to get a route from A to B

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="viewport"
        content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-
scalable=0">
<meta name="apple-mobile-web-app-capable" content="yes">
<title>Control : route</title>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}

#results {
        background-color: white;
        position: absolute;
        width: 300px;
        right: 0px;
        overflow: auto;
        z-index: 15000;
        top: 0px;
        font-size: 11px;
        height: 250px;
        cursor: pointer;
}

span {
        display: block;
        width: 26px;
        height: 26px;
}

.tableRoute {
        width: 99%;
}
```

```
.tdNav {
        background-color: #5a87dd;
        width: 27px;
}

.navF {
        background: url(../scripts/img/signs.png) no-repeat -98px -1px;
}

.navFR {
        background: url(../scripts/img/signs.png) no-repeat -194px -1px;
}

.navFL {
        background: url(../scripts/img/signs.png) no-repeat -164px -1px;
}

.navBL {
        background: url(../scripts/img/signs.png) no-repeat -292px -1px;
}

.navBR {
        background: url(../scripts/img/signs.png) no-repeat -324px -1px;
}

.navL {
        background: url(../scripts/img/signs.png) no-repeat -34px -1px;
}

.navR {
        background: url(../scripts/img/signs.png) no-repeat -65px -1px;
}

.navround_about_entry {
        background: url(../scripts/img/signs.png) no-repeat -450px -1px;
}

.navround_about_exit {
        background: url(../scripts/img/signs.png) no-repeat -482px -1px;
}

.tdDist {
        white-space: nowrap;
}

.tdDur {
        white-space: nowrap;
}
```

```
.tdSpeed {
        white-space: nowrap;
}

tr:hover,.gcui-route tr:hover.odd {
        background-color: #D7E1F1;
}

tr.odd {
        background-color: #EBF0F8;
}
</style>
<script src="http://api.geoconcept.net/htc/proj4js.js"></script>
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France'
                };
                var map = new GCUI.Map('map', options);
                map.onEvent("load", onMapLoaded);
        }
        function onMapLoaded() {
                var routeLayer = new OpenLayers.Layer.Vector('routeLayer',

 OpenLayers.Util.extend(OpenLayers.Feature.Vector.style, {
                                        'default' : {
                                                strokeWidth : 4,
                                                strokeColor : "#1BE01B"
                                        },
                                        'select' : {
                                                strokeWidth : 4,
                                                strokeColor : "blue"
                                        }
                                }));
                var currentXY = null;
                var map = GCUI.getMap('map');
                map.events.register('mousemove', null, function(evt) {
                        currentXY = evt.xy;
                });
                routeLayer.events.on({
                        'featureselected' : function(feature) {
                                feature = feature.feature;
                                var name = feature.attributes.name ?
 feature.attributes.name
                                                + '<br>' : '';
                                var info = "<div style='font-size:.8em'>" + name +
 'Durée: '
```

```
                                                + feature.attributes.duration +
 '<br>Distance: '
                                                + feature.attributes.distance + "</
div>";
                                var popup = new
 OpenLayers.Popup.FramedCloud("chicken", GCUI

 .getMap('map').getLonLatFromPixel(currentXY), null,
                                                info, null, true, null);
                                feature.popup = popup;
                                GCUI.getMap('map').addPopup(popup);
                        },
                        'featureunselected' : function(feature) {
                                feature = feature.feature;
                                GCUI.getMap('map').removePopup(feature.popup);
                                feature.popup.destroy();
                                feature.popup = null;
                        }
                });
                map.addLayer(routeLayer);
                var route = new GCUI.Control.Route({
                        autoActivate : true,
                        layer : routeLayer
                });
                map.addControl(route);
                route.route({
                        origin : new OpenLayers.LonLat(0.18, 48), // Le Mans
                        destination : new OpenLayers.LonLat(2.35, 48.86), // Paris
                        waypoints : [ new OpenLayers.LonLat(1.91, 47.90) ], //
 Orléans
                        callback : function(resp, options) {
                                var features = this.displayRoute(resp); // this ==
 the route control
                                displayRouteSheet(resp); // display route sheet in
 #results div
                        }
                });

                var select = new OpenLayers.Control.SelectFeature(routeLayer, {
                        hover : true,
                        autoActivate : true
                });
                map.addControl(select);
        }
        function displayRouteSheet(route) {
                var legs = route.legs;
                var routeSheet = '<table>';
                for (var i = 0; i < legs.length; i++) {
                        var leg = legs[i];
```

```
                        var steps = leg.steps;
                        for (var j = 0; j < steps.length; j++) {
                                var step = steps[j];
                                step.speed = Math.round((3600 * step.distance)
                                                / (1000 * step.duration));
                                step.distance = step.distance < 1000 ? step.distance
 + ' m'
                                                : (step.distance / 1000).toFixed(1)
 + ' km';
                                step.duration = step.duration < 60 ? step.duration +
 ' s'
                                                : Math.round(step.duration / 60) + '
 mn';
                                var bounds = new OpenLayers.Bounds();
                                for (var k = 0; k < step.points.length; k++) {
                                        var pt = step.points[k].split(',');
                                        bounds.extendXY(parseFloat(pt[0]),
 parseFloat(pt[1]));
                                }
                                routeSheet += '<tr'
                                                + ((j % 2 == 0) ? ' class="odd"' :
 "")
                                                + ' onclick="zoomOn('
                                                + bounds.bottom
                                                + ','
                                                + bounds.left
                                                + ','
                                                + bounds.top
                                                + ','
                                                + bounds.right
                                                + ')"><td class="tdNav"><span
 class="nav'+step.navInstruction+'"></span></td><td class="tdName">'
                                                + step.name + '</td><td
 class="tdDist">'
                                                + step.distance + '</td><td
 class="tdDur">'
                                                + step.duration + '</td><td
 class="tdSpeed">'
                                                + step.speed + ' km/h</td></tr>';
                        }
                }
                document.getElementById('results').innerHTML = routeSheet + '</
table>';
        }
        function zoomOn(bottom, left, top, right) {
                var map = GCUI.getMap('map');
                map.zoomToExtent(new OpenLayers.Bounds(left, bottom, right, top));
        }
</script>
```

```
</head>
<body onload="init()">
        <div id="map" style="width: 100%; height: 100%; position: relative;"></div>
        <div id="results"></div>
</body>
</html>
```

See this example [examples/control-route.html]

6. Layer

6.1. Adding a Geoconcept layer

To add a Geoconcept layer to the map, use the `GCUI.Layer.GeoConcept` constructor :

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="viewport"
        content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-
scalable=0">
<meta name="apple-mobile-web-app-capable" content="yes">
<title>Layer : Geoconcept</title>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<style>
.smallmap {
        width: 512px;
        height: 256px;
        border: 1px solid #ccc;
}

h1 {
        color: #003a6b;
        background-color: transparent;
        font: 100% 'Lucida Grande', Verdana, Geneva, Lucida, Arial, Helvetica,
                sans-serif;
        margin: 0;
        padding-top: 0.5em;
}
</style>
<script src="http://api.geoconcept.net/htc/OpenLayers.js"></script>
<script src="http://api.geoconcept.net/htc/htc-lite.js"></script>
<script>
        function init() {
                var map = new OpenLayers.Map('map', {
                        projection : new OpenLayers.Projection("EPSG:27582"),
                        maxExtent : new OpenLayers.Bounds(5000, 1620000, 1198000,
 2678000)
                });
                map.addControl(new OpenLayers.Control.LayerSwitcher());

                var urls = [ "http://gcweb.geoconcept.com/gws/wmts" ];
                var layer = new GCUI.Layer.GeoConcept("GeoConcept Layer", urls, {
                        layer : 'France'
                });
                layer.events.on({
```

```
                        'init' : function() {
                                map.zoomToMaxExtent();
                        }
                });
                map.addLayer(layer);

                var wms = new OpenLayers.Layer.WMS("Géosignal WMS",
                                "http://www.geosignal.org/cgi-bin/wmsmap?", {
                                        layers : "Regions,Departements"
                                });
                map.addLayer(wms);

        }
</script>
</head>
<body onload="init()">
        <h1 id="title">Geoconcept HTC (High Traffic Client) Example</h1>
        <p id="shortdesc">Demonstrates a Geoconcept HTC layer that loads
                tiles from a web accessible disk-based cache.</p>
        <div id="map" class="smallmap"></div>
</body>
</html>
```

See this example [examples/layer-geoconcept.html]

6.2. Adding a thematic layer

To add a Geoconcept thematic layer, use the `GCUI.Layer.Thematic` constructor :

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="viewport"
        content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-
scalable=0">
<meta name="apple-mobile-web-app-capable" content="yes">
<title>Layer : thematic</title>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<style>
.mapLayerControl {
        left: 25px;
}

#legends img {
        float: left;
}
```

```
</style>
<script>
        var server = "http://gcweb.geoconcept.com/gws/wmts";
        var layer = "DemoThematic1";

        function init() {
                var options = {
                        server : server,
                        layer : "DemoThematic1"
                };
                var map = new GCUI.Map('map', options);
                map.onEvent("load", onMapLoaded);
        }
        function showLegend() {
                var options = {
                        //items : this.legendTitle + (this.type==='color' ? '||' :
 '|||||')
                        items : this.legendTitle
                                        + '|sub-title|str2|str3|str4|str5|str6|str7|
str8|str9|str10|str11|str12|str13|str14|str15|str16'
                };
                if (this.type !== 'color') {
                        options.items = this.legendTitle
                                        + '|str1|str2|str3|sub-title|str5|str6|str7|
str8|str9|str10|str11|str12|str13|str14 %|str15 %|str16 %';
                }
                var legendUrl = this.getLegendUrl(options);
                var img = document.createElement("img");
                img.src = legendUrl;
                img.id = 'legendImg_' + this.id;
                document.getElementById("legends").appendChild(img);
        }
        function toggleLegend() {
                var img = document.getElementById('legendImg_' + this.id);
                if (img) {
                        img.style.visibility = this.visibility ? 'visible' :
 'hidden';
                }
        }
        function onMapLoaded() {
                var map = GCUI.getMap('map');
                var layerSwitcher = new GCUI.Control.LayerSwitcher();
                map.addControl(layerSwitcher);
                map.zoomToMaxExtent();

                var thematicColor = new GCUI.Layer.Thematic("Couleur", server, {
                        //tabname : tabname,
                        layer : "DemoThematic1",
                        classes : [ {
```

```
                        minValue : 0.0,
                        maxValue : 2.0,
                        color : 'F7F700'
                }, {
                        minValue : 2.0,
                        maxValue : 4.0,
                        color : 'F7B411'
                }, {
                        minValue : 4.0,
                        maxValue : 10.0,
                        color : 'F77122'
                } ],
                gcType : 'PYR',
                gcSubType : 'DEPT',
                data : [ {
                        id : '44',
                        v : 1
                }, {
                        id : '35',
                        v : 3
                }, {
                        id : '22',
                        v : 25
                }, {
                        id : '29',
                        v : ''
                }, {
                        id : '56',
                        v : 7
                } ],
                noValueColor : 'FFFFFF',
                legendTitle : 'My color legend title'
        }, {
                opacity : 0.5
        });
        thematicColor.events.on({
                'init' : showLegend,
                'visibilitychanged' : toggleLegend
        });
        map.addLayer(thematicColor);

        var thematicSymbol = new GCUI.Layer.Thematic("Symbole", server, {
                legendTitle : 'My thematic legend title',
                //tabname : tabname,
                layer : "DemoThematic1",
                type : 'symbol',
                classes : [ {
                        minValue : 0.0,
                        maxValue : 2.0,
```

```
                                color : '1240AB',
                                symbol : 'BLACKCIRCLE',
                                size : 10
                        }, {
                                minValue : 2.0,
                                maxValue : 4.0,
                                color : '4671D5',
                                symbol : 'BLACKCIRCLE',
                                size : 20
                        }, {
                                minValue : 4.0,
                                maxValue : 10.0,
                                color : '6C8CD5',
                                symbol : 'BLACKCIRCLE',
                                size : 30
                        } ],
                        gcType : 'PYR',
                        gcSubType : 'DEPT',
                        data : [ {
                                id : '44',
                                v : 1
                        }, {
                                id : '35',
                                v : 3
                        }, {
                                id : '22',
                                v : 25
                        }, {
                                id : '29'
                        }, {
                                id : '56',
                                v : 7
                        } ]
                });
                thematicSymbol.events.on({
                        'init' : showLegend,
                        'visibilitychanged' : toggleLegend
                });
                map.addLayer(thematicSymbol);
        }
</script>
</head>
<body onload="init()">
        <div id="map" class="smallmap" style="width: 800px; height: 500px;"></div>
        <div id="legends"></div>
</body>
</html>
```

See this example [examples/layer-thematic.html]

6.3. Adding a vector layer to the Map

To add a vector layer, use the `OpenLayers.Layer.Vector` constructor :

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="viewport"
        content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-
scalable=0">
<meta name="apple-mobile-web-app-capable" content="yes">
<title>Layer : vector</title>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France'
                };
                var map = new GCUI.Map('map', options);
                map.onEvent("load", onMapLoaded);
        }
        function onMapLoaded() {
                var map = GCUI.getMap('map');
                map.zoomToMaxExtent();

                var vectorLayer = new OpenLayers.Layer.Vector("Marker");
                var feature = new OpenLayers.Feature.Vector(
                                new OpenLayers.Geometry.Point(596914, 2428874), {
                                        some : 'data'
                                }/*, {
                                                                externalGraphic : '../
pictures/marker.png',
                                                                graphicHeight : 21,
                                                                graphicWidth : 16
                                                        }*/);
                vectorLayer.addFeatures(feature);
                map.addLayer(vectorLayer);
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 800px; height: 500px;"></div>
</body>
</html>
```

See this example [examples/layer-vector.html]

6.4. Adding a WMS layer to the Map

To add a WMS layer, use the `OpenLayers.Layer.WMS` constructor :

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="viewport"
        content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-
scalable=0">
<meta name="apple-mobile-web-app-capable" content="yes">
<title>Layer : Geoconcept</title>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<style>
.smallmap {
        width: 512px;
        height: 256px;
        border: 1px solid #ccc;
}

h1 {
        color: #003a6b;
        background-color: transparent;
        font: 100% 'Lucida Grande', Verdana, Geneva, Lucida, Arial, Helvetica,
                sans-serif;
        margin: 0;
        padding-top: 0.5em;
}
</style>
<script src="http://api.geoconcept.net/htc/OpenLayers.js"></script>
<script src="http://api.geoconcept.net/htc/htc-lite.js"></script>
<script>
        function init() {
                var map = new OpenLayers.Map('map', {
                        projection : new OpenLayers.Projection("EPSG:27582"),
                        maxExtent : new OpenLayers.Bounds(5000, 1620000, 1198000,
 2678000)
                });
                map.addControl(new OpenLayers.Control.LayerSwitcher());

                var urls = [ "http://gcweb.geoconcept.com/gws/wmts" ];
                var layer = new GCUI.Layer.GeoConcept("GeoConcept Layer", urls, {
                        layer : 'France'
                });
                layer.events.on({
```

```
                        'init' : function() {
                                map.zoomToMaxExtent();
                        }
                });
                map.addLayer(layer);

                var wms = new OpenLayers.Layer.WMS("Géosignal WMS",
                                "http://www.geosignal.org/cgi-bin/wmsmap?", {
                                        layers : "Regions,Departements"
                                });
                map.addLayer(wms);

        }
</script>
</head>
<body onload="init()">
        <h1 id="title">Geoconcept HTC (High Traffic Client) Example</h1>
        <p id="shortdesc">Demonstrates a Geoconcept HTC layer that loads
                tiles from a web accessible disk-based cache.</p>
        <div id="map" class="smallmap"></div>
</body>
</html>
```

See this example [examples/layer-geoconcept.html]

7. Event

7.1. Map loading event

The *GCUI.Map* [../../../../../jsdoc/files/GeoConcept/Map-js.html#GCUI.Map.GCUI.Map] constructor with *`server`*, *`mapName`* and *`tab`* options arguments gets asynchronously map informations (extent, resolutions, …) from the LBS Platform server. The map events system trigger a *load* event to notify that the map is loaded :

```
<!DOCTYPE html>
<html>
<head>
<title>Basic map : load</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}
</style>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France',
                        x : 601106,
                        y : 2428909,
                        scale : 6
                };
                var map = new GCUI.Map('map', options);
                map.onEvent("load", onMapLoaded);
        }
        function onMapLoaded() {
                var map = GCUI.getMap('map');
                alert("Map is loaded. Map extent is : " + map.getExtent());
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 100%; height: 100%;"></div>
</body>
```

```
</html>
```

The following code registers a callback function that will be called when map is ready :

```
map.onEvent("load",onMapLoaded);
```

In this example, the callback function display the map extent :

```
function onMapLoaded() {
    var map = GCUI.getMap('map');
    alert("Map is loaded. Map extent is : "+ map.getExtent());
}
```

See this example [examples/basic-map-load.html]

7.2. Map moveend event

This example shows how to listen to the map `moveend` event :

```
<!DOCTYPE html>
<html>
<head>
<title>Event : map moveend</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}
</style>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France',
                        x : 601106,
                        y : 2428909,
                        scale : 6
                };
                var map = new GCUI.Map('map', options);
                map.events.on({
                        'moveend' : function() {
```

```
                                alert('End move, map extent is : ' +
 map.getExtent());
                        }
                });
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 100%; height: 100%;"></div>
</body>
</html>
```

See this example [examples/event-moveend.html]

7.3. Map zoomend event

This example shows how to listen to the map `zoomend` event :

```
<!DOCTYPE html>
<html>
<head>
<title>Event : map zoomend</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}
</style>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France',
                        x : 601106,
                        y : 2428909,
                        scale : 6
                };
                var map = new GCUI.Map('map', options);
                map.events.on({
                        'zoomend' : function() {
                                alert('End zoom, map zoom is : ' + map.getZoom());
```

```
                        }
                });
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 100%; height: 100%;"></div>
</body>
</html>
```

See this example [examples/event-zoomend.html]

You can also register actions on the other map events types : *Map events* [../../../../../jsdoc/files/OpenLayers/Map-js.html#OpenLayers.Map.events]

7.4. Map click event

You can add a control to the map to register and define an action on the click event :

```
<!DOCTYPE html>
<html>
<head>
<title>Event : click</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}
</style>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        OpenLayers.Control.Click = OpenLayers.Class(OpenLayers.Control,
                        {
                                defaultHandlerOptions : {
                                        'single' : true,
                                        'double' : false,
                                        'pixelTolerance' : 0,
                                        'stopSingle' : false,
                                        'stopDouble' : false
                                },

                                initialize : function(options) {
```

```
                                        this.handlerOptions =
 OpenLayers.Util.extend({},
                                                        this.defaultHandlerOptions);

 OpenLayers.Control.prototype.initialize.apply(this,
                                                        arguments);
                                        this.handler = new
 OpenLayers.Handler.Click(this, {
                                                //'dblclick' : this.trigger,
                                                'click' : this.trigger
                                        }, this.handlerOptions);
                                },

                                trigger : function(e) {
                                        var map = GCUI.getMap('map');
                                        var lonlat = map.getLonLatFromPixel(e.xy);
                                        alert("You clicked near x: " + lonlat.lon +
 " , y: "
                                                        + +lonlat.lat);
                                }

                        });
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France',
                        x : 601106,
                        y : 2428909,
                        scale : 6
                };
                var map = new GCUI.Map('map', options);
                var c = new OpenLayers.Control.Click();
                map.addControl(c);
                c.activate();
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 100%; height: 100%;"></div>
</body>
</html>
```

See this example [examples/event-click.html]

7.5. Feature select event

You can add a control to the map to register and define an action on the feature selected event :

```
<!DOCTYPE html>
```

```
<html>
<head>
<title>Event : feature selected</title>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France'
                };
                var map = new GCUI.Map('map', options);
                map.onEvent("load", onMapLoaded);
        }
        function onMapLoaded() {
                var map = GCUI.getMap('map');
                map.zoomToMaxExtent();

                var vectorLayer = new OpenLayers.Layer.Vector("Marker");
                var feature = new OpenLayers.Feature.Vector(
                                new OpenLayers.Geometry.Point(596914, 2428874), {
                                        mydata : 'my feature data 1'
                                }/*, {
                                                                externalGraphic : '../
pictures/marker.png',
                                                                graphicHeight : 21,
                                                                graphicWidth : 16
                                                        }*/);
                vectorLayer.addFeatures(feature);
                feature = new OpenLayers.Feature.Vector(new
 OpenLayers.Geometry.Point(
                                528914, 2438974), {
                        mydata : 'my feature data 2'
                }/*, {
                                        externalGraphic : '../pictures/marker.png',
                                        graphicHeight : 21,
                                        graphicWidth : 16
                                }*/);
                vectorLayer.addFeatures(feature);
                map.addLayer(vectorLayer);

                vectorLayer.events.on({
                        'featureselected' : function(e) {
                                var feature = e.feature;
                                var geom = feature.geometry;
                                var popup = new GCUI.Popup.Anchored("myPopupId",
                                                new OpenLayers.LonLat(geom.x,
 geom.y), null,
```

```
                                                feature.data.mydata, true);
                                popup.autoSize = true;
                                map.addPopup(popup, true);
                        },
                        'featureunselected' : function(e) {
                        }
                });
                var hoverCtrl = new OpenLayers.Control.SelectFeature(vectorLayer, {
                        hover : true,
                        autoActivate : true
                });
                map.addControl(hoverCtrl);
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 800px; height: 500px;"></div>
</body>
</html>
```

See this example [examples/event-featureselect.html]

7.6. Feature modify event

You can add a control to the map to register and define an action on the feature modified event :

```
<!DOCTYPE html>
<html>
<head>
<title>Event : feature modified</title>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France'
                };
                var map = new GCUI.Map('map', options);
                map.onEvent("load", onMapLoaded);
        }
        function onMapLoaded() {
                var map = GCUI.getMap('map');
                map.zoomToMaxExtent();

                var vectorLayer = new OpenLayers.Layer.Vector("Marker");
                var feature = new OpenLayers.Feature.Vector(
                                new OpenLayers.Geometry.Point(596914, 2428874), {
```

```
                                        mydata : 'my feature data 1'
                                }/*,
                         {externalGraphic: '../pictures/marker.png', graphicHeight:
 21, graphicWidth: 16}*/);
                vectorLayer.addFeatures(feature);
                feature = new OpenLayers.Feature.Vector(new
 OpenLayers.Geometry.Point(
                                528914, 2438974), {
                        mydata : 'my feature data 2'
                }/*,
                                {externalGraphic: '../pictures/marker.png',
 graphicHeight: 21, graphicWidth: 16}*/);
                vectorLayer.addFeatures(feature);
                map.addLayer(vectorLayer);

                vectorLayer.events.on({
                        'featuremodified' : function(evt) {
                                var feature = evt.feature;
                                alert(feature.geometry);
                        }
                });
                var modifyControl = new
 OpenLayers.Control.ModifyFeature(vectorLayer);
                map.addControl(modifyControl);
                modifyControl.activate();
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 800px; height: 500px;"></div>
</body>
</html>
```

See this example [examples/event-featuremodified.html]

8. Projection

8.1. Centering map on WGS84 coordinates

```
<!DOCTYPE html>
<html>
<head>
<title>Basic map : center lonlat</title>
<style>
html {
        height: 100%;
}

body {
        height: 100%;
        margin: 0;
        padding: 0;
}
</style>
<link rel="stylesheet"
        href="http://api.geoconcept.net/htc/htc/skins/htc.css">
<script src="http://api.geoconcept.net/htc/proj4js.js"></script>
<script src="http://api.geoconcept.net/htc/htc.js"></script>
<script>
        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France'
                };
                var map = new GCUI.Map('map', options);
                map.onEvent('load', onMapLoaded);
        }
        function onMapLoaded() {
                var map = GCUI.getMap('map');
                var lonlat = new OpenLayers.LonLat(2.328450697115128,
 48.80814035316188);
                map.moveTo(lonlat.transform(new OpenLayers.Projection('EPSG:4326'),
                                new OpenLayers.Projection(map.getProjection())), 8);
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 100%; height: 100%;"></div>
</body>
</html>
```

See this example [examples/basic-map-center-lonlat.html]

9. Strategy

9.1. Cluster strategy

```
<!DOCTYPE html>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8">
<meta name="viewport"
        content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-
scalable=0">
<meta name="apple-mobile-web-app-capable" content="yes">
<title>Strategy : cluster</title>
<script src="http://api.geoconcept.net/htc/OpenLayers.js"></script>
<script src="http://api.geoconcept.net/htc/htc-lite.js"></script>
<script>
        var map, layer;

        function init() {
                var options = {
                        server : 'http://gcweb.geoconcept.com/gws/wmts',
                        layer : 'France'
                };
                map = new GCUI.Map('map', options);
                map.onEvent("load", onMapLoaded);
        }
        function onMapLoaded() {
                map.zoomToMaxExtent();

                var style = new OpenLayers.Style({
                        pointRadius : "${radius}",
                        fillColor : "#597EC2",
                        fillOpacity : 0.8,
                        strokeColor : "#cc6633",
                        strokeWidth : "${width}",
                        strokeOpacity : 0.8,
                        label : "${label}",
                        fontColor : "white"
                }, {
                        context : {
                                width : function(feature) {
                                        return (feature.cluster) ? 2 : 1;
                                },
                                radius : function(feature) {
                                        var pix = 5;
                                        if (feature.cluster) {
                                                pix =
  Math.min(feature.attributes.count, 7) + 6;
```

```
                                        }
                                        return pix;
                                },
                                label : function(feature) {
                                        return feature.attributes.count || '';
                                }
                        }
                });

                // create a semi-random grid of features to be clustered
                var dx = 500;
                var dy = 500;
                var px, py;
                var features = [];
                for (var x = 594248; x <= 604691; x += dx) {
                        for (var y = 2424401; y <= 2433462; y += dy) {
                                px = x + (2 * dx * Math.random());
                                py = y + (2 * dy * Math.random());
                                features.push(new OpenLayers.Feature.Vector(
                                                new OpenLayers.Geometry.Point(px,
py), {
                                                        x : px,
                                                        y : py
                                                }));
                        }
                }

                var strategy = new OpenLayers.Strategy.Cluster({
                        distance : 50,
                        threshold : 2
                });

                var clusters = new OpenLayers.Layer.Vector("Clusters", {
                        strategies : [ strategy ],
                        styleMap : new OpenLayers.StyleMap({
                                "default" : style,
                                "select" : {
                                        fillColor : "#8aeeef",
                                        strokeColor : "#32a8a9",
                                        fontColor : "black"
                                }
                        })
                });
                map.addLayer(clusters);

                clusters.addFeatures(features);

                var select = new OpenLayers.Control.SelectFeature(clusters, {
                        hover : true
```

```
                });
                map.addControl(select);
                select.activate();
                clusters.events.on({
                        "featureselected" : display
                });
                //console.log(clusters.getDataExtent());
                //map.zoomToExtent(clusters.getDataExtent());

        }

        function display(event) {
                var f = event.feature;
                var el = document.getElementById("output");
                if (f.cluster) {
                        el.innerHTML = "cluster of " + f.attributes.count + "
 points";
                } else {
                        el.innerHTML = "unclustered " + f.geometry;
                }
        }
</script>
</head>
<body onload="init()">
        <div id="map" style="width: 800px; height: 500px;"></div>
        <span id="output"></span>
</body>
</html>
```

See this example [examples/strategy-cluster.html]

Glossary of GEOCONCEPT terms

Coordinates x,y	On a map (plane surface), geographic position (outdistance, horizontally and vertically, by report to an origin point) corresponding to the same position on the surface of the Earth (curve surface).
GCIS	Geoconcept Internet Server : the Geoconcept map server.
HTC	High Traffic Client : Optimized client and cache server for Geoconcept Internet Server. Designed for geographic web sites with a large number of simultaneous access. It's compatible with all browsers (no ActiveX, Java..).
Layer	Group of geographic datas organized by thematic. In this API, we take into account 4 layers: the *main* layer for the dynamic map object, the *obj* layer for the objects on the map, the *sheet* layer for the infobox of the objects on the map and the *slider* layer for the scale slider shown on the map.
Logical scale	The scale is the relation between the dimensions of the entity of a map and the dimensions of real geographic objects represented. Geoconcept manages 12 zoom levels. The logical scale in Geoconcept is represented by an integer between 1 (for the maximum zoom level) and 12 (for the minimum zoom level).
Precision	Notion linked to the geometric description (localization and form) of geographic objects (entities). This value corresponds to the number of decimal significant figures used for the storage of the entities coordinates when the coordinates are expressed in the map unit.
Ratio	A ratio is an enlarging report between distance on the map and distance in pixels on the screen.
Slider	Graphic element representing the logical scale of the map.
Visibility tab	The visibility tab defines what can be visible on the map for a specific zoom level. Examples of view tabs : *Tracking*, *All*, *Administrative*, *Hydrographie*, *Roads*, …