

TWaver HTML5 GIS支持地图数据的展示，同时还支持将TWaver HTML5拓扑数据与Map地图进行整合，目前支持WMS服务、Google Map、Bing Map、OpenStreetMap、MapABC和百度地图。本教程要求用户熟悉JavaScript语言。想要更深入了解GIS的基本原理，需要开发者了解更多关于OGC(Open Geospatial Consortium)定义的地理参考模型WMS(Web Map Service)与WFS(Web Feature Service)，以及空间数据结构等相关知识。如果仅仅是用来在地图上展示TWaver拓扑，本教程可以给予您帮助。该章节主要介绍一个最简单的GIS所需要的程序代码，让用户对TWaver GIS有个初步的了解。

```

1 <!DOCTYPE HTML>
2 <html>
3 <head>
4     <title>HelloWorld</title>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <style type="text/css">
7         #container {
8             position: absolute;
9             margin-left: 10px;
10            margin-top: 10px;
11        }
12    </style>
13    <script src="twaver.js" type="text/javascript" ></script>
14    <script>
15        function initMap() {
16            var container = document.getElementById("container");
17            //创建一个地图，把地图放在id为container的容器中
18            var map = new twaver.gis.Map(container,"myMap");
19            //设置地图的尺寸
20            map.setSize({width:500,height:500});
21            //设置不能够双击缩放地图
22            map.setEnableDoubleZoom(false);
23            //以MapABC作为地图，注意（这里访问的是MapABC的在线地图）
24            map.addLayer("mapABC",GISConsts.EXECUTOR_TYPE_MAPABC);
25            //如果有地图服务器，并且把地图发布为tile图片，可以把图片放在tomcat之类的服务器
26            //map.addLayer("google",GISConsts.EXECUTOR_TYPE_MAPABC,"http://locc
27            //设置地图的缩放级别
28            map.setZoomLevel(10);
29            //设置地图的中心位置
30            map.setCenter(new twaver.gis.geometry.GeoCoordinate(31.41,121.48))
31        }
32    </script>
33 </head>
34 <body onload="initMap()">
35 <div id="container">
36 </div>
37 </body>
38 </html>

```

接下来我们逐步分析一下上述代码。

(1) 创建地图需要一个HTML元素作为容器，这样地图才能显示，这里我们以id为container的元素作为地图容器。

```
1 | var container = document.getElementById("container");
```

( 2 ) 创建一个地图组件，地图的id为myMap。注意：( 这里的命名空间为twaver.gis, 所有的gis类都在这个命名空间下)。

```
1 | map = new twaver.gis.Map(container, "myMap");
```

( 3 ) 设置地图的大小。

```
1 | map.setSize({width:500,height:500});
```

( 4 ) 以MapABC作为地图背景，注意：( 这里访问的是MapABC的在线地图 )。

```
1 | map.addLayer("mapABC",GISConsts.EXECUTOR_TYPE_MAPABC);
```

如果有地图服务器，并且把地图发布为tile图片，可以把图片放在tomcat之类的服务器中发布起来，然后直接访问。

```
1 | map.addLayer("google",GISConsts.EXECUTOR_TYPE_MAPABC,"http://localhost:8080/locd
```

( 5 ) 设置地图的缩放级别。

```
1 | map.setZoomLevel(5);
```

( 6 ) 设置地图的中心位置。

```
1 | map.setCenter(new twaver.gis.geometry.GeoCoordinate(31.41,121.48));
```

这样一个简单地图就创建完成：



该章节主要介绍gis和拓扑的整合，要求用户熟悉TWaver HTML5的知识。

(1) 首先创建一个network视图对象和box容器，这个属于TWaver HTML5的基本知识，这里不在赘述。

```
1 var box = new twaver.ElementBox();
2 var network = new twaver.vector.Network(box);
```

(2) 创建一个适配器对象，调用bindNetworkAndMap方法绑定network和map。

```
1 var adapter = new twaver.gis.Adapter();
2 adapter.bindNetworkAndMap(network, map);
```

(3) 创建一个经纬度对象，指定经纬度。注意（GeoCoordinate类构造方法的第一个参数是纬度，第二个参数是经度）。

```
1 var geo = new twaver.gis.geometry.GeoCoordinate(34.7, 113.6);
```

(4) 创建一个节点，并把节点放在box中。

```
1 var from = new twaver.Node();
2 box.add(from);
```

(5) 给节点设置经纬度。

```
1 from.setClient(GISSettings.GEOCOORDINATE, geo);
```

完整代码：

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <title>HelloWorld</title>
5   <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6   <style type="text/css">
7     #container {
8       position: absolute;
9       margin-left: 10px;
10      margin-top: 10px;
11    }
12  </style>
13  <script src="twaver.js" type="text/javascript" ></script>
14  <script>
15    var map, network, box;
16    function initMap() {
17      var container = document.getElementById("container");
18      //创建一个地图，把地图放在id为container的容器中
19      map = new twaver.gis.Map(container, "myMap");
20      //设置地图的尺寸
21      map.setSize({width: 500, height: 500});
```

```

22 //以MapABC作为地图，注意（这里访问的是MapABC的在线地图）
23 map.addLayer("mapABC",GISConsts.EXECUTOR_TYPE_MAPABC);
24 //设置地图的缩放级别
25 map.setZoomLevel(5);
26 //设置地图的中心位置
27 map.setCenter(new twaver.gis.geometry.GeoCoordinate(31.41,121.48))
28
29 //创建box
30 var box = new twaver.ElementBox();
31 //创建network
32 network = new twaver.vector.Network(box);
33
34 //创建一个适配器对象，用来绑定地图和拓扑
35 var adapter = new twaver.gis.Adapter();
36 adapter.bindNetworkAndMap(network,map);
37
38 //创建拓扑，注意（创建节点要放在创建adapter后边，如果放在adapter前边，由于地图
39 createTopo();
40 }
41
42 function createTopo() {
43     var box = network.getElementBox();
44     //创建一个经纬度对象
45     var geo = new twaver.gis.geometry.GeoCoordinate(34.7, 113.6);
46     var from = new twaver.Node();
47     //给节点from设置经纬度
48     from.setClient(GISSettings.GEOCOORDINATE, geo);
49     box.add(from);
50
51     var geo = new twaver.gis.geometry.GeoCoordinate(30, 120);
52     var to = new twaver.Node();
53     to.setClient(GISSettings.GEOCOORDINATE, geo);
54     box.add(to);
55
56     var link = new twaver.Link(from, to);
57     box.add(link);
58 }
59 </script>
60 </head>
61 <body onload="initMap()">
62 <div id="container">
63 </div>
64 </body>
65 </html>

```

效果图如下：



TWaver HTML5 GIS中提供了几个地图的控件供大家使用，一个是用来缩放的导航条，另一个是鹰眼。注意（**地图控件的命名空间在twaver.gis.gadget下**）

（1）**Navigator**：地图缩放控件，用来进行地图的缩放，该类的构造方法有两个参数，一个是自身的id，一个是map（**参数类型为twaver.gis.Map**）。示例中，创建一个Navigator对象，然后把Navigator放在地图的坐标点为(20,50)的位置。

```
1 var navigator = new twaver.gis.gadget.Navigator("myNav",map);
2 map.installGadget(navigator,{x:20,y:50});
```

（2）**Overview**：地图的鹰眼，该类的构造方法有两个参数，一个是自身的id，一个是map（**参数类型为twaver.gis.Map**）。示例中，创建一个Overview对象，大小为宽150，高150，设置边框的样式：宽1像素、实线、颜色为红色，然后把Overview放在地图的坐标点为(350,350)的位置。

```
1 var overview = new twaver.gis.gadget.Overview("overview",map);
2 overview.setSize({width:150,height:150});
3 overview.map.addLayer("mapABC",GISConsts.EXECUTOR_TYPE_MAPABC);
4 //设置overview的边框，类似于css的用法
5 overview.setBorder("1px solid red");
6 map.installGadget(overview,{x:350,y:350});
```

示例代码如下：

```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4     <title>HelloWorld</title>
5     <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
6     <style type="text/css">
7         #container {
8             position: absolute;
9             margin-left: 10px;
10            margin-top: 10px;
11        }
12    </style>
13    <script src="twaver.js" type="text/javascript" ></script>
14    <script>
15        var map,network,box;
16        function initMap() {
17            var container = document.getElementById("container");
18            //创建一个地图，把地图放在id为container的容器中
19            map = new twaver.gis.Map(container,"myMap");
20            //设置地图的尺寸
21            map.setSize({width:500,height:500});
22            //以MapABC作为地图，注意（这里访问的是MapABC的在线地图）
23            map.addLayer("mapABC",GISConsts.EXECUTOR_TYPE_MAPABC);
24            //设置地图的缩放级别
25            map.setZoomLevel(5);
26            //设置地图的中心位置
27            map.setCenter(new twaver.gis.geometry.GeoCoordinate(31.41,121.48))
28        }
```



```

29 //创建box
30 var box = new twaver.ElementBox();
31 //创建network
32 network = new twaver.vector.Network(box);
33
34 //创建缩放条
35 var navigator = new twaver.gis.gadget.Navigator("myNav",map);
36 map.installGadget(navigator,{x:20,y:50});
37
38 //增加鹰眼
39 var overview = new twaver.gis.gadget.Overview("overview",map);
40 overview.setSize({width:150,height:150});
41 overview.map.addLayer("mapABC",GISConsts.EXECUTOR_TYPE_MAPABC);
42 //设置overview的边框,类似于css的用法
43 overview.setBorder("1px solid red");
44 map.installGadget(overview,{x:350,y:350});
45
46
47 //创建一个适配器对象,用来绑定地图和拓扑
48 var adapter = new twaver.gis.Adapter();
49 adapter.bindNetworkAndMap(network,map);
50
51 //创建拓扑,注意(创建节点要放在创建adapter后边,如果放在adapter前边,由于地图
52 createTopo();
53
54 map.addMapEventListener(function(event) {
55     console.log(event);
56 });
57 }
58
59 function createTopo() {
60     var box = network.getElementBox();
61     //创建一个经纬度对象
62     var geo = new twaver.gis.geometry.GeoCoordinate(34.7, 113.6);
63     var from = new twaver.Node();
64     //给节点from设置经纬度
65     from.setClient(GISSettings.GEOCOORDINATE, geo);
66     box.add(from);
67
68     var geo = new twaver.gis.geometry.GeoCoordinate(30, 120);
69     var to = new twaver.Node();
70     to.setClient(GISSettings.GEOCOORDINATE, geo);
71     box.add(to);
72
73     var link = new twaver.Link(from, to);
74     box.add(link);
75 }
76 </script>
77 </head>
78 <body onload="initMap()">
79 <div id="container">
80 </div>
81 </body>
82 </html>

```

效果图如下：





TWaver HTML5 GIS中地图的监听器分为两大类，一个是Map作为HTML DOM本身的事件监听，例如单击、双击、鼠标移动等；另一是用来监听地图自身属性变化的监听器，例如用来监听地图的图层变化、缩放级别变化、中心位置变化等。

示例用法：

```
1 //监听鼠标的单击事件
2 map.getView().addEventListener("click", function(event) {
3     //console.log(event);
4 });
5 //监听鼠标的双击事件
6 map.getView().addEventListener("dblclick", function(event) {
7     //console.log(event);
8 });
9 //监听地图的变化
10 map.addMapEventListener(function(event) {
11     //console.log(event);
12 });
```

注意：如果拓扑和地图结合使用，想要监听拓扑的变化，可以直接用拓扑组件自身的事件监听器去监听。

---

Map类 (命名空间：twaver.gis)

构造方法

构造函数	描述
Map(container,id)	在指定的容器内创建地图实例。

实例方法

方法	返回值	描述
setEnableDoubleClick(enable:Boolean)	none	设置是否可以双击来缩放地图。
isEnableDoubleClick()	Boolean	判断是否可以双击来缩放地图，返回值为true或false。如果为true，可以双击缩放地图，如果为false，不能双击缩放地图。
setEnableMouseWheel(enable:Boolean)	none	设置是否可以滚动滚轮来缩放地图。
isEnableMouseWheel()	Boolean	判断否可以滚动滚轮来缩放地图，返回值为true或false。如果为true，可以滚动滚轮来缩放地图，如果为false，不能滚动滚轮来缩放地图。
getView()	HTMLELEMENT	获取地图自身的容器，返回值是一个html元素，比如div。
addLayer(name:String,type:String,path:String)	none	添加地图图层，第一个参数为图层名称；第二个参数为地图类型，取值为GISConsts.EXECUTOR_TYPE_*，可以参考GISConsts类中的所有该变量的定义；第三个参数可选，表示本地缓存图片的url。
removeLayer(name:String)	none	移除地图图层，参数为图层名称。
setZoomLevel(zoom:int)	none	设置地图的缩放级别。
getZoomLevel()	int	得到当前地图的缩放级别。
setCenter(geo:GeoCoordinate)	none	设置地图的中心位置。
getCenter()	GeoCoordinate	得到当前地图的中心位置。
setBackgroundColor(color:String)	none	设置地图的背景，参数为rgb或rgba字符串。
getGeoPointFromPointOnMap(x:int,y:int)	GeoCoordinate	把屏幕坐标转换为经纬度坐标，返回值为经纬度。
getScreenPointFromGeoPoint(geo:GeoCoordinate)	Object	把经纬度坐标转换为屏幕坐标。
installGadget(gadget:Navigator Overview,position:Object)	none	把控件放在地图的指定位置，第一个参数是被放置的控件，第二个参数是位置，例如{x:10,y:10}。
setZoomLowLimit(level:int)	none	设置地图的最小缩放级别。
setZoomUpLimit(level:int)	none	设置地图的最大缩放级别。
addMapEventListener(listener,scope)	none	监听地图的变化，监听的事件类型可以参考MapEvent类。

Navigator类 (命名空间：twaver.gis.gadget)

构造方法

构造函数	描述
Navigator(id:String,map:Map)	创建一个导航条，第一个参数是自身的id，第二个参数是导航条所依附的地图。

Overview类 (命名空间：twaver.gis.gadget)

构造方法

构造函数	描述
Overview(id:String,map:Map)	创建一个鹰眼，第一个参数是自身的id，第二个参数是鹰眼所依附的地图。

实例方法

方法	返回值	描述
setSize(size:Object)	none	设置Overview的尺寸，例如：setSize({width:150,height:150})。
setBorder(border:String)	none	设置鹰眼的边框，参数为String类型，类似于css的形式，例如：setBorder("1px solid red")。

**Adapter类 (命名空间: twaver.gis)**

构造方法

构造函数	描述
Adapter()	适配器类，可以将TWaver的Network和地图绑定起来，让拓扑在地图上显示。

实例方法

方法	返回值	描述
bindNetworkAndMap(network:twaver.network.Network twaver.canvas.Network twaver.vector.Network,map:Map)	none	把network和map绑定。

**GISConsts类**

常量列表

常量	描述
EXECUTOR_TYPE_BINGMAP	BingMap地图执行器。
EXECUTOR_TYPE_YAHOO	YAHOO地图执行器。
EXECUTOR_TYPE_MAPABC	MAPABC地图执行器。
EXECUTOR_TYPE_GOOGLEMAP	Google地图执行器。
EXECUTOR_TYPE_OSM	OpenStreetMap地图执行器。
EXECUTOR_TYPE_GEOSERVER_WMS	GeoServer WMS服务执行器。
EXECUTOR_TYPE_ARCGIS_WMS	Arcgis WMS服务执行器。

**MapEvent类**

常量列表

常量	描述
EXECUTOR_TYPE_BINGMAP	地图尺寸发生变化。
CENTER_CHANGED	地图中心位置发生变化。
ZOOM_CHANGED	地图缩放级别发生变化。
UPLIMIT_CHANGED	地图的最大缩放上限发生变化。
LOWLIMIT_CHANGED	地图的最小缩放下限发生变化。
LAYER_ADDED	增加图层。
LAYER_EXECUTOR_CHANGED	地图执行器发生变化。
LAYER_ORDER_CHANGED	地图图层顺序发生变化。
LAYER_INDEX_CHANGED	地图图层的索引发生变化。
ALL_LAYERS_REMOVED	删除所有的图层。

-----