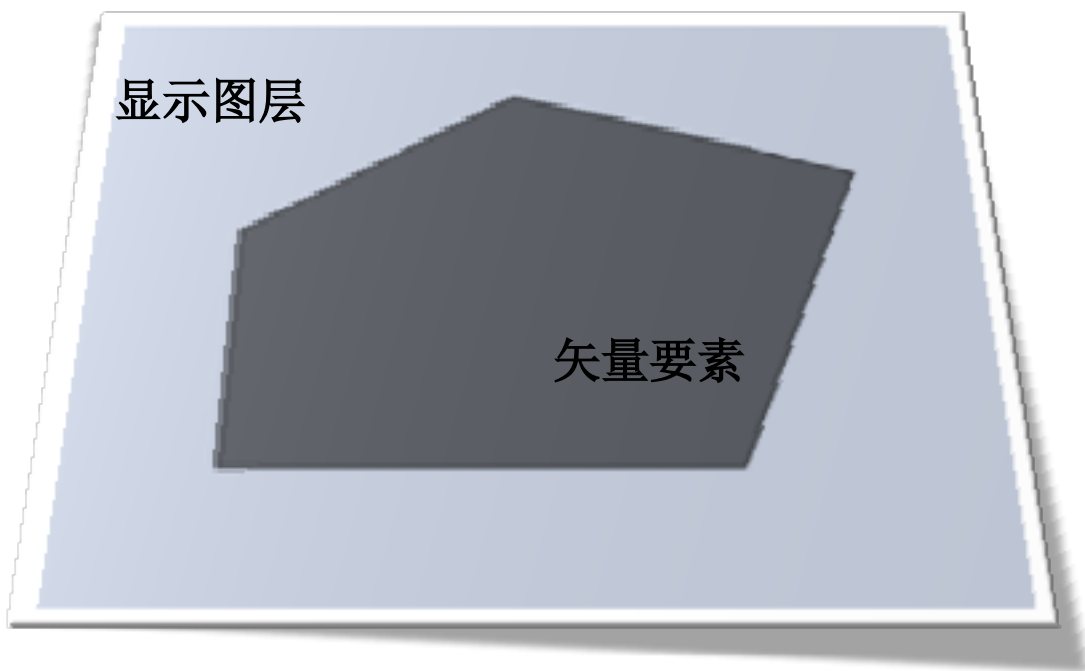




SuperMap iClient for JavaScript 绘制对象

- 矢量数据客户端显示方法
- 点对象客户端标注方法
- 交互式绘制几何对象的方法
- 气泡窗口的显示方法
- 几何对象鼠标事件的显示方法
- 标注对象的选择事件



获取/创建
矢量要素

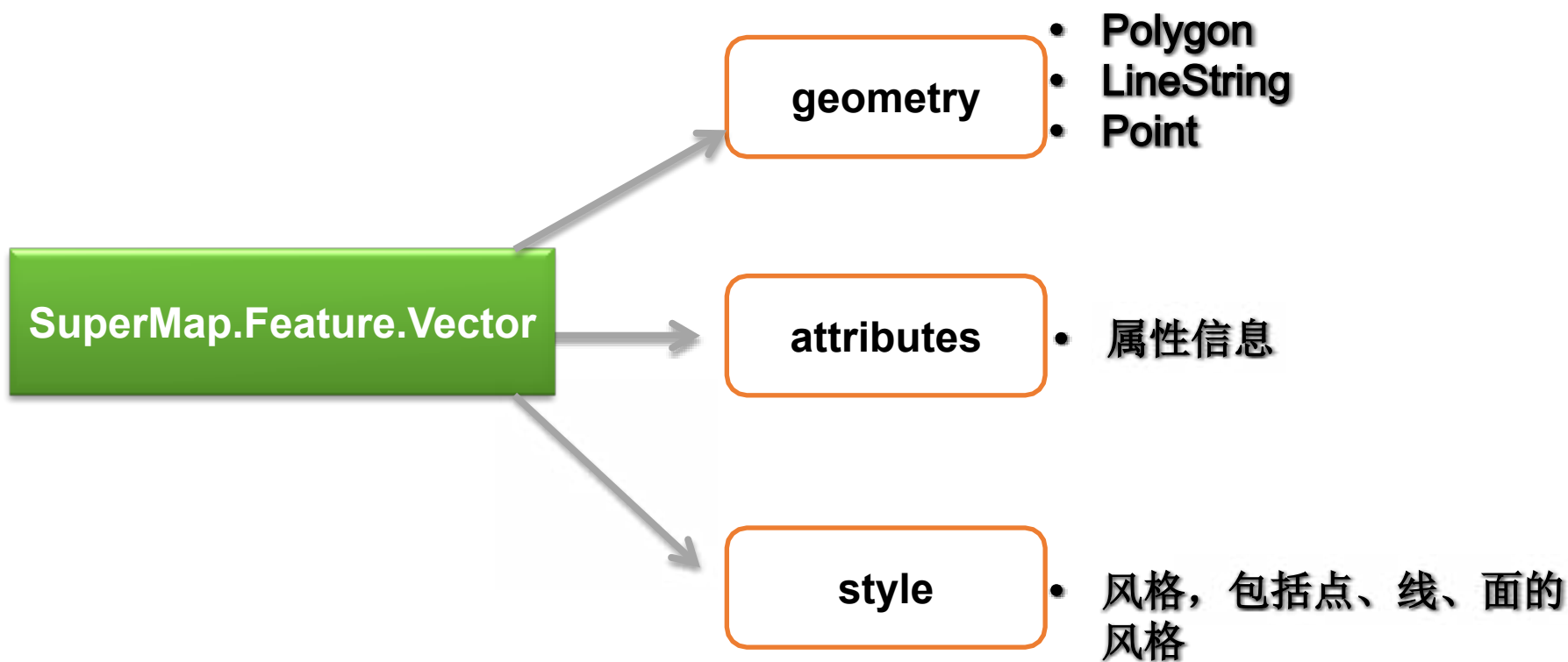


设置矢量要素
显示风格



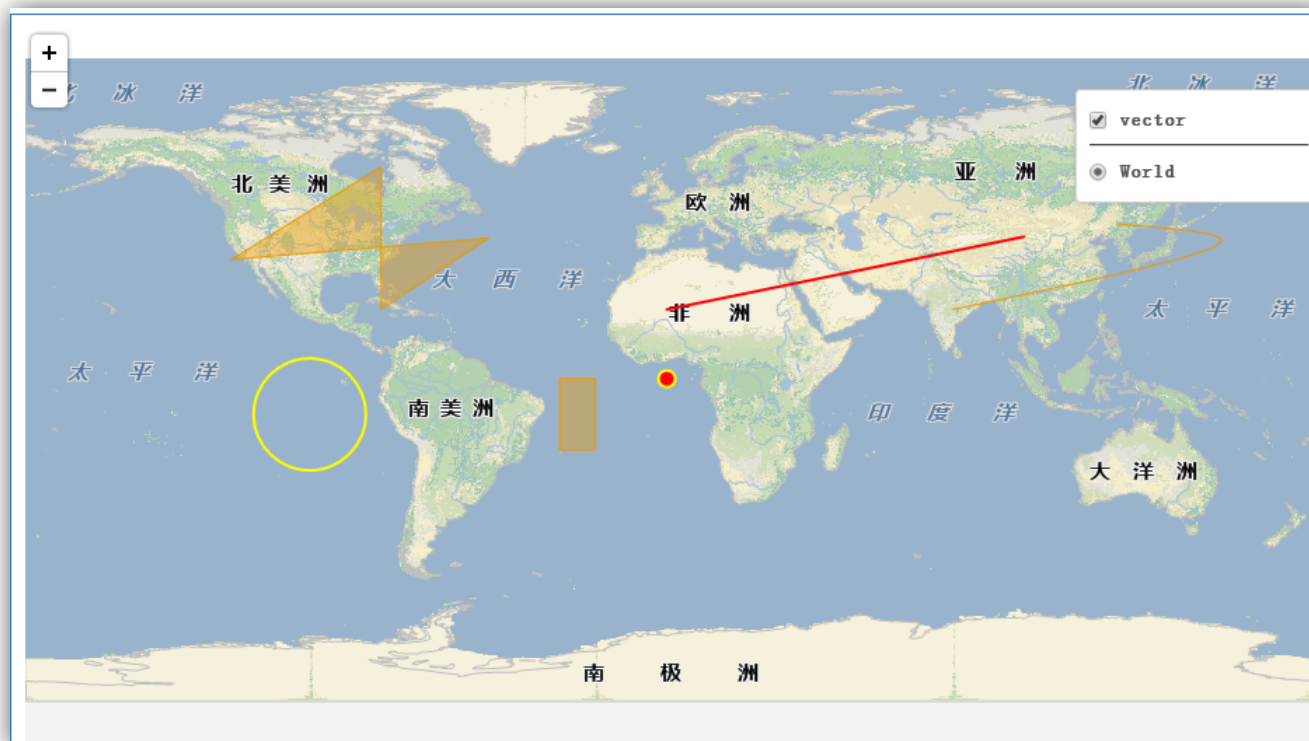
加载几何对象
到图层

- SuperMap.Feature.Vector, 点、线、面、文本等
 - 显示于SuperMap.Layer.Vector图层



■ SuperMap. Layer. Vector

- 可显示点、线、面、文本等



■ SuperMap.Layer.Vector 图层的使用

接口	说明
addFeatures	添加要素对象
removeFeatures	移除要素对象
removeAllFeatures	移除所有要素
drawFeature	在图层上绘制要素

■ 样式Style的使用

- 有关Style的相关属性请查阅帮助文档中 **SuperMap. Feature. Vector. style**

fill	{Boolean} 不需要填充则设置为false。
fillColor	{String} 十六进制填充颜色，默认为"#ee9900"。
fillOpacity	{Number} 填充不透明度。默认为0.4。
stroke	{Boolean} 不需要描边则设为false。
strokeColor	{String} 十六进制描边颜色。
strokeOpacity	{Number} 描边的不透明度(0-1),默认为0.4。
strokeWidth	{Number} 像素描边宽度，默认为1。

- 方法一：通过设置 SuperMap.Feature.Vector 的 style 属性获得样式

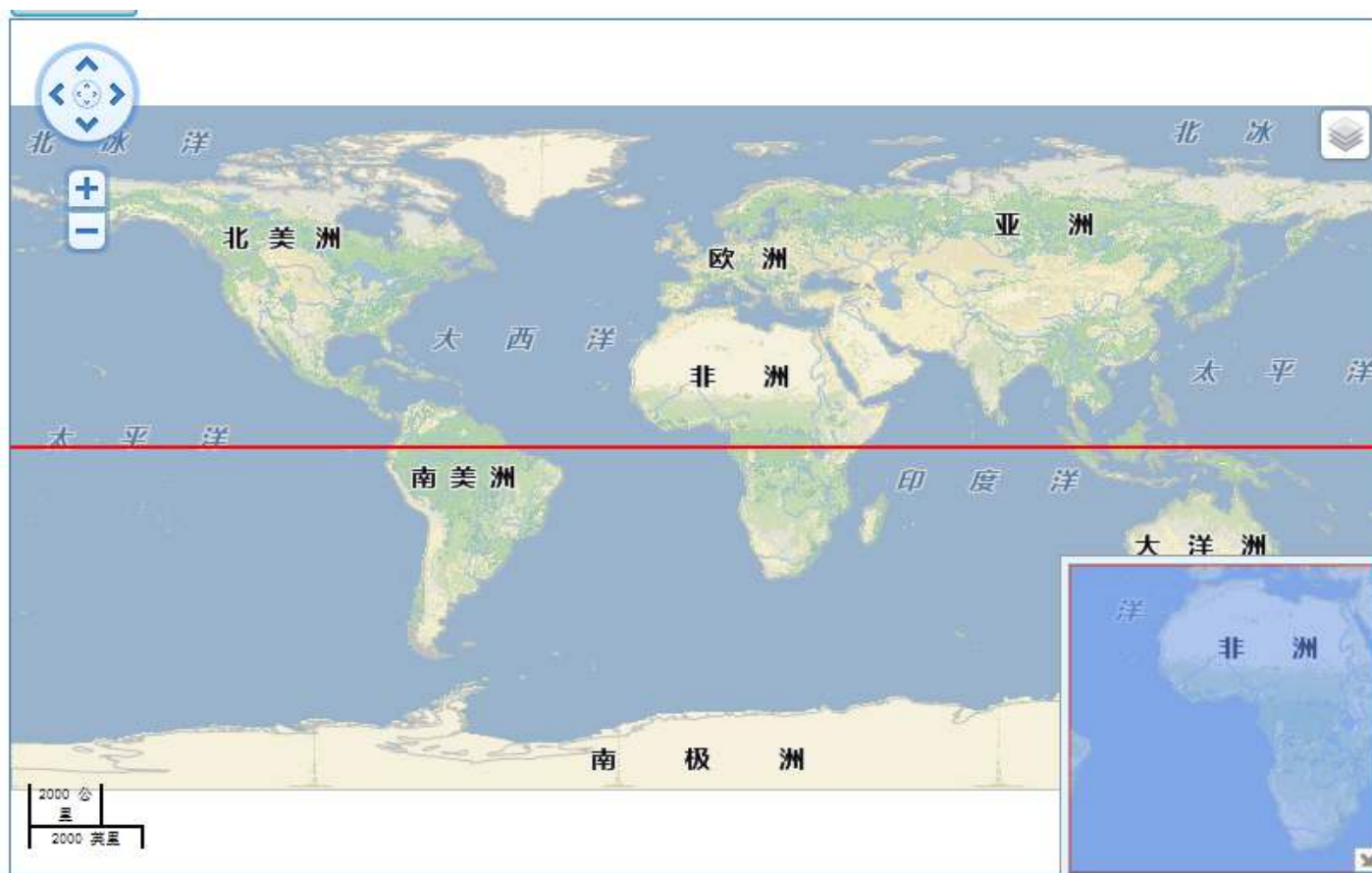
```
var vectorLayer = new SuperMap.Layer.Vector("Vector Layer");
feature = new SuperMap.Feature.Vector();
feature.style = {
    strokeColor: "#304DBE",
    strokeWidth: 1,
    pointerEvents: "visiblePainted",
    fillColor: "#304DBE",
    fillOpacity: 0.8,
    pointRadius: 2
};
vectorLayer.addFeatures(feature);
```


■ 方法二：直接设置图层对象的style对象

```
var vectorLayer = new SuperMap.Layer.Vector("Vector Layer",  
{style: myStyle});
```

使用优先级比较 **feature.style > layer.style**

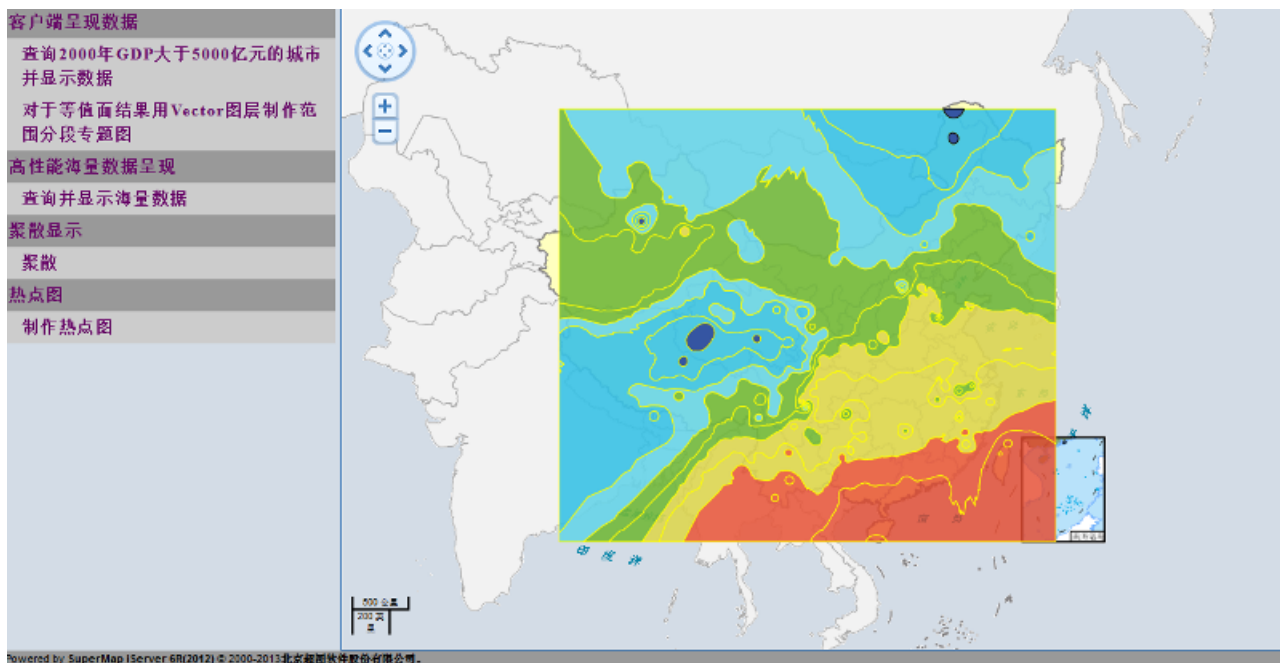
■ 在世界地图中，绘制赤道线



■ 主要代码

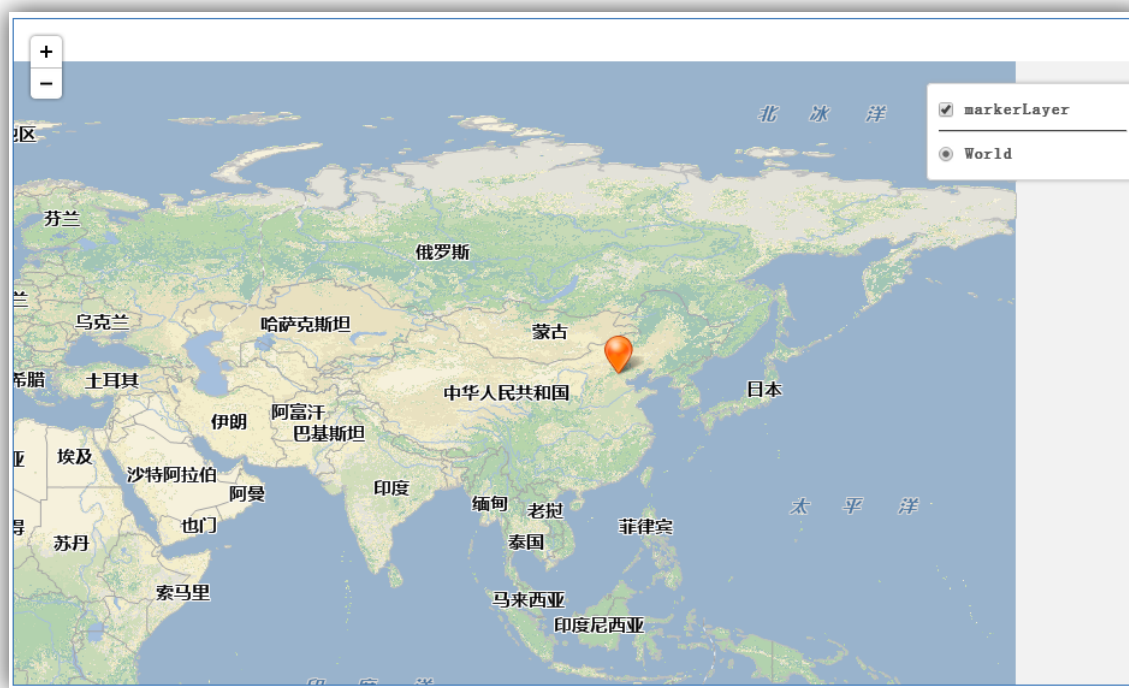
```
// 显示赤道
function show(){
    // 获取赤道的线段
    var points = new Array();
    points.push(new SuperMap.Geometry.Point(180,0));
    points.push(new SuperMap.Geometry.Point(-180,0));
    var geometry = new SuperMap.Geometry.LineString(points);
    var style = {
        strokeColor: "red", //"#304DBE",
        strokeWidth: 2,
        pointerEvents: "visiblePainted",
        fillColor: "#304DBE",
        fillOpacity: 0.8
    }
    // 将赤道显示在vector图层上
    var feature = new SuperMap.Feature.Vector(geometry,null,style);
    lineLayer.drawFeature(feature,style);
}
```

- 原理：根据SuperMap.Feature.Vector的attribute属性进行范围分段并设置SuperMap.Feature.Vector.style值
- 实例：根据等温面的温度值制作范围分段专题图



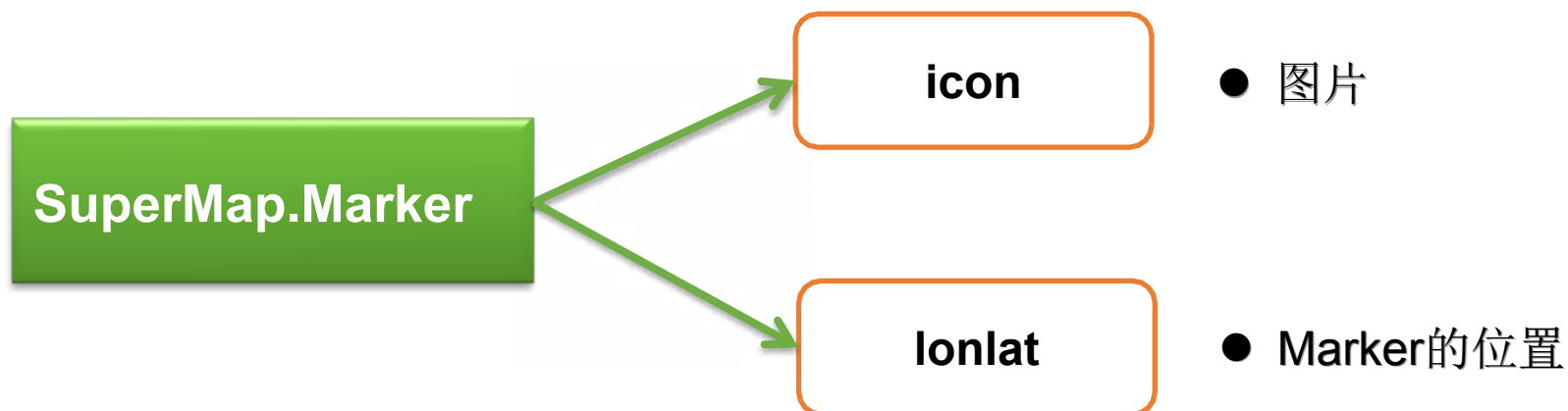
■ 标注图层：SuperMap.Layer.Markers

■ 显示点对象的标注信息



■ 标注对象：SuperMap.Marker

- 对点进行标注，可以自定义选择标注的图标
- 需添加到SuperMap.Layer.Markers图层上



■ 标注对象的事件，实现对标注的鼠标操作响应

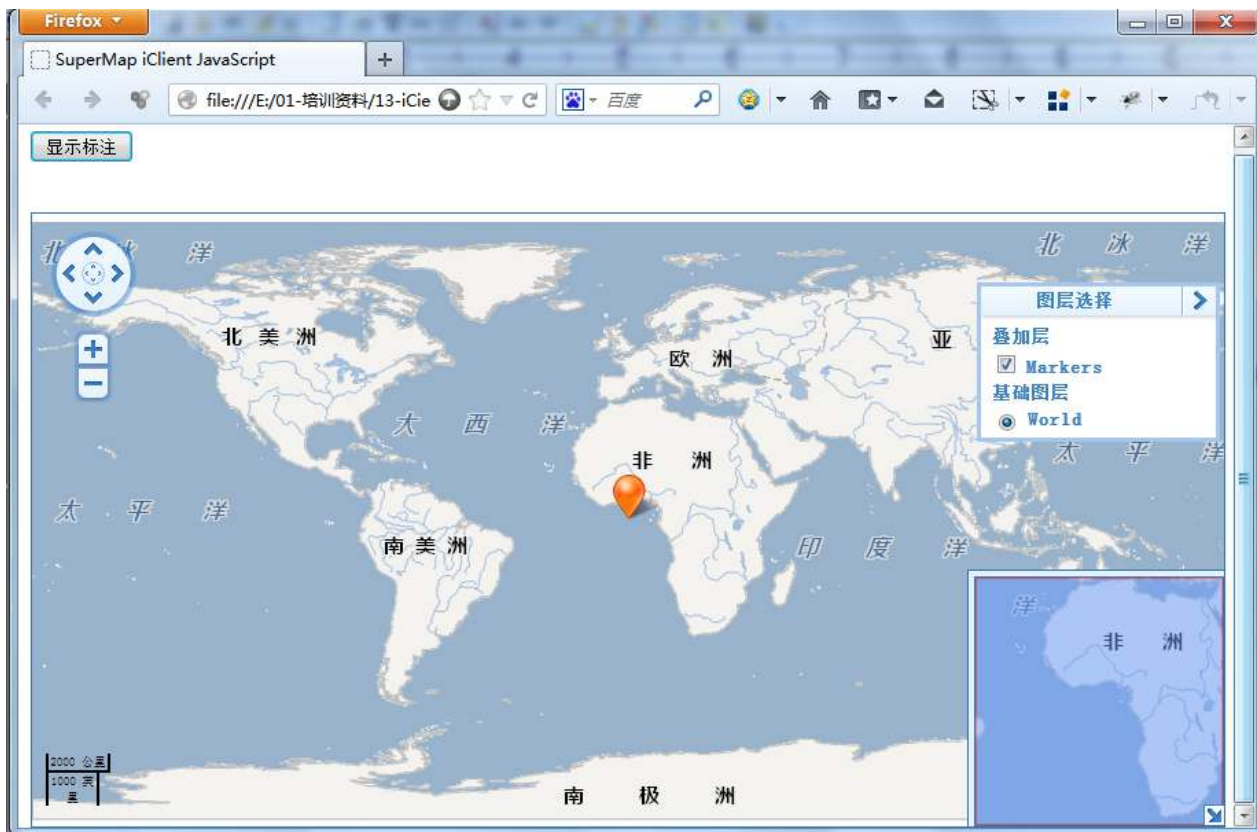
事件名称	说明
click	当鼠标单击maker时触发此事件。
dblclick	当鼠标双击maker时触发此事件。
mousedown	当鼠标在maker上按下时触发此事件。
mouseup	当鼠标在maker上按下并放开时触发此事件。
mousemove	当鼠标移过maker时触发此事件。
mouseout	当鼠标移出maker时触发此事件。
mouseover	当鼠标移进maker时触发此事件。
rightclick	当鼠标右键单击maker时触发此事件。

添加鼠标响应事件的同时，需要添加“**scope**”—作用域设置

■ SuperMap.Layer.Markers图层的使用

接口	说明
Markers.addMarker (marker)	添加Marker地标
Markers.removeMarker	清除Marker地标

- 创建标签图层，并在 (0, 0) 的坐标显示图标，并实现鼠标点击图标时更换图标的功能。



■ 标注功能主要代码

```
// 添加标签
function ShowMarker(){
    // 创建标签图层
    markers = new SuperMap.Layer.Markers( "Markers" );
    map.addLayer(markers);
    // 添加标签对象
    var size = new SuperMap.Size(44,33);
    var offset = new SuperMap.Pixel(-(size.w/2), -size.h);
    var icon = new
SuperMap.Icon('./resource/controlImages/marker.png', size, offset);
    markers.addMarker(new SuperMap.Marker(new
SuperMap.LonLat(0,0),icon));
}
```

■ 点击标注更换图片的主要代码

```
mar = new SuperMap.Marker(new SuperMap.LonLat(0,0),icon);
// 添加marker对象的click事件监听
mar.events.on({"click":onclick, "scope": mar});}
function onclick(event){
    var marker = this;
    var size = new SuperMap.Size(21,25);
    var offset = new SuperMap.Pixel(-(size.w/2), -size.h);
    var icon = new
SuperMap.Icon('../theme/images/cluster4.png',size,offset);
    var mar = new SuperMap.Marker(marker.getLonLat(),icon);
    markers.addMarker(mar);
    markers.removeMarker(marker);
}
```

交互式操作绘制对象

```
graph TD; A[交互式操作绘制对象] --> B[获取几何对象]; B --> C[设置几何对象显示风格]; C --> D[加载几何对象到图层];
```

获取几何对象

设置几何对象显示风格

加载几何对象到图层

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/105124111214011311>