**1. 添加缩放控件**



代码：

1. //实例化ZoomSlider控件
2. var zoomslider = new ol.control.ZoomSlider();
3. //加载ZoomSlider控件到地图容器中
4. map.addControl(zoomslider);
6. //实例化ZoomToExtent
7. var zoomToExtent = new ol.control.ZoomToExtent({
8. extent: [
9. //Todo 这个范围应该怎么确定
10. 13100000, 4290000,
11. 13200000, 5210000
12. ]
13. });
14. //加载ZoomToExtent到map中
15. map.addControl(zoomToExtent);

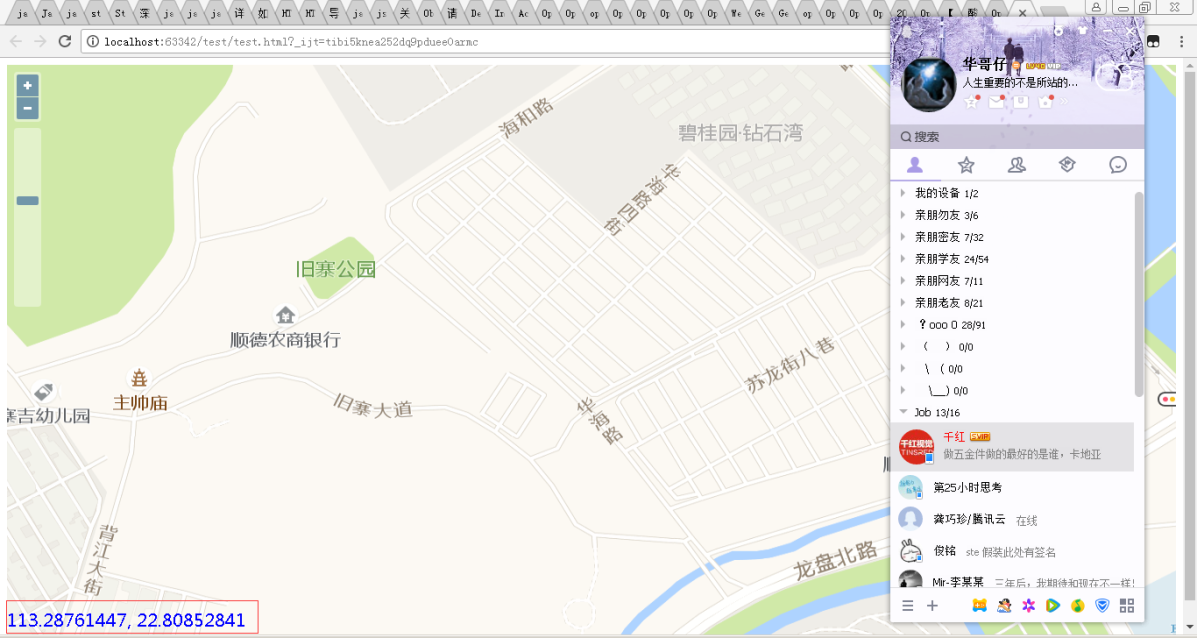
**2.呈现比例尺**



代码：

*//实例化比例尺控件***var** scaleLineControl = **new** ol.control.ScaleLine({  
 *//设置度量单位为米* **units**: **'metric'**,  
 **target**: **'scalebar'**,  
 **className**: **'ol-scale-line'**});  
map.addControl(scaleLineControl);

**3.根据鼠标移动位置显示经纬度坐标**



代码：

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>

<title></title>

<script src="../lib/ol/ol.js"></script>

<link href="../css/ol.css" rel="stylesheet" />

<style type="text/css">

#myposition

{

float:left;

position:absolute;

bottom:10px;

width:400px;

height:20px;

z-index:2000;

}

.mosuePosition

{

color:blue;

font-size:20px;

font-family:'微软雅黑';

}

</style>

<script type="text/javascript">

window.onload = function () {

//初始化鼠标位置控件

var mousePositionControl = new ol.control.MousePosition({

//样式类名称

className: 'mosuePosition',

//投影坐标格式，显示小数点后边多少位

coordinateFormat: ol.coordinate.createStringXY(8),

//指定投影

projection: 'EPSG:4326',

//目标容器

target:document.getElementById('myposition')

});

//初始化地图容器

var map = new ol.Map({

target:'map',

layers:[

new ol.layer.Tile({

source:new ol.source.OSM()

}),

],

view:new ol.View({

center:[0,0],

zoom:3

})

});

//将鼠标位置坐标控件加入到map中

map.addControl(mousePositionControl);

}

</script>

</head>

<body>

<div id="map">

<div id="myposition"></div>

</div>

</body>

</html>

**4.在地图上画图**

4.1 关系图



4.2 步骤

Step1: 创建source

**var** source = **new** ol.**source**.Vector({**wrapX**: **false**});

Step2: 创建层，用于绑定到map中，显示画好的图形，并将source绑定到其中

**var** vector = **new** ol.layer.Vector({  
 **source**: source  
});

Step3: 创建map ,并将地图层与图形显示层绑定到map中

**var** map = **new** ol.Map({  
 **layers**: [gaodeMapLayer, vector],  
 **view**: **new** ol.View({  
 **center**: [113.290007,22.810068],  
 **projection**: **'EPSG:4326'**,  
 **zoom**: 17  
 }),  
 **target**: **'map'**,  
 **logo**: **false**});

Step4: 创建draw,并将source绑定到draw中

**draw** = **new** ol.interaction.Draw({  
 **source**: source,  
 **type**: */\*\** ***@type*** *{ol.geom.GeometryType} \*/* **"Circle"**});

Step5: 将draw关联到map

map.addInteraction(**draw**);

**5.各类地图的source**

5.1.1 天地图(路图)

**var** tian\_di\_tu\_road\_layer = **new** ol.layer.Tile({  
 **title**: **"天地图路网"**,  
 **source**: **new** ol.**source**.XYZ({  
 **url**: **"http://t4.tianditu.com/DataServer?T=vec\_w&x={x}&y={y}&l={z}"** })  
});

5.1.2 天地图(卫星图)

**var** tian\_di\_tu\_satellite\_layer = **new** ol.layer.Tile({  
 **title**: **"天地图卫星影像"**,  
 **source**: **new** ol.**source**.XYZ({  
 **url**: **'http://t{0-4}.tianditu.com/DataServer?T=img\_w&x={x}&y={y}&l={z}'** })  
});

5.1.3 天地图(注释图)

**var** tian\_di\_tu\_annotation = **new** ol.layer.Tile({  
 **title**: **"天地图文字标注"**,  
 **source**: **new** ol.**source**.XYZ({  
 **url**: **'http://t3.tianditu.com/DataServer?T=cva\_w&x={x}&y={y}&l={z}'** })  
});

5.2.1 高德地图（路图）

**var** gaodeMapLayer = **new** ol.layer.Tile({  
 **source**: **new** ol.**source**.XYZ({  
 **url**: **'http://wprd0{1-4}.is.autonavi.com/appmaptile?lang=zh\_cn&size=1&style=7&x={x}&y={y}&z={z}'** })  
});

5.2.1 高德地图（卫星图）

**var** gaodeMapLayerSat = **new** ol.layer.Tile({  
 **source**: **new** ol.**source**.XYZ({  
 **url**: **'http://webst0{1-4}.is.autonavi.com/appmaptile?style=6&x={x}&y={y}&z={z}'** })  
});

# 6. [Openlayers获取鼠标点击时的经纬度](http://www.cnblogs.com/lianche/p/3623332.html)

map.events.register('click', map, function (e) {

var pixel = new OpenLayers.Pixel(e.xy.x,e.xy.y);

var lonlat = map.getLonLatFromPixel(pixel);

lonlat.transform(new OpenLayers.Projection("EPSG:900913"), new OpenLayers.Projection("EPSG:4326")); //由900913坐标系转为4326  
　　　　 alert(lonlat.lon+", "+lonlat.lat);  
})

7.创建点与线（自动创建，不是通过人工画）

**7.1 创建点**

**Step1: 创建Featrue,设置geometry属性**

var saoguan = new ol.Feature({geometry:new ol.geom.Point(ol.proj.fromLonLat([113.5991,24.8166]))

});

**Step2: 设置此点的样式**

saoguan.setStyle(new ol.style.Style({

image : new ol.style.Icon({

color : '#4271AE',

src : 'data/dot.png'

})

})

);

**Step3: 将feature传入source**

var source = new ol.source.Vector({

features:[saoguan]

});

**Step4: 将source传入到layer**

var layer = new ol.layer.Vector({

source: source

});

**Step5: 设置底图layer**

var rasterLayer = new ol.layer.Tile({

source: new ol.source.OSM()

})；

Step6: 最后将地图layer和点layer一并传入map中

var map = new ol.Map({

layers: [rasterLayer, layer],

target: document.getElementById('map'),

view: new ol.View({

center: ol.proj.fromLonLat([113.5991,24.8166]),

zoom: 3

})

**7.2 创建线**

**Step1: 创建Featrue,设置geometry属性**

var saoguan = new ol.Feature({**geometry**:**new** ol.geom.LineString(  
 [[113.290007,22.810068], [113.310007, 22.810068] , [113.300007, 22.830068]])});

**Step2: 设置此点的样式**

feature.setStyle(new ol.style.Style({

stroke: new ol.style.Stroke({

width: 3,

color: [255, 0, 0, 1]

})

}));

**Step3: 将feature传入source**

var source = new ol.source.Vector({

features:[saoguan]

});

**Step4: 将source传入到layer**

var layer = new ol.layer.Vector({

source: source

});

**Step5: 设置底图layer**

var rasterLayer = new ol.layer.Tile({

source: new ol.source.OSM()

})；

Step6: 最后将地图layer和点layer一并传入map中

var map = new ol.Map({

layers: [rasterLayer, layer],

target: document.getElementById('map'),

view: new ol.View({

center: ol.proj.fromLonLat([113.5991,24.8166]),

zoom: 3

})

**8.更换地图**

**var** s = **new** ol.**source**.XYZ({  
 **url**: **'http://wprd0{1-4}.is.autonavi.com/appmaptile?lang=zh\_cn&size=1&style=7&x={x}&y={y}&z={z}'**});  
**var** l = map.getLayers().getArray()[0];  
l.setSource(s);