# OpenLayers 教程

# 1 开始使用 openlayers

## 1.1 设置

先到它的官方网站 http://www.openlayers.org 下载他的压缩包，解压。

拷贝目录下的 OpenLayer.js、根目录下的 lib 目录、根目录下的 img 目录到你网站的 Scripts 目录下（当然，这个只是例子，您网站的目录结构您自己说得算，只要保证 OpenLayers.js，/lib，/img 在同一目录中即可）。 然后，创建一个****.html 作为查看地图的页面。

# 2 试验 openlayers

环境：geoserver1.7

Openlayers2.4

Dreamviever8

## 2.1 第一个地图窗口

目标：用 openlayers 加载 geoserver wms。

步骤：

（1） 空白 html 文件

（2） 插入 div-map

（3） 为 div 付风格

以上为未加载地图的静态页面

代码为：

效果为：

（4） 插入 openlayers 代码引用

```
<link rel="stylesheet" type="text/css" href="../theme/default/style.css"/>
<script src="OpenLayers.js" type="text/javascript"></script>
```

（5） 写 js 代码，主要是 init（）

第一个地图窗口就完成了

**注1. js 中 defer 的作用是页面加载完成后，执行脚本。**

注2.　　222

## 2.2　控制地图与 div 的占据区域

目标：让地图默认占满展现区
方法：
设置 map 的 options，由其中两个因素决定：maxExtent－最大地图边界；maxResolution－最大解析度。
当 maxExtent 设置为地图的最大边界后，maxResolution 设置为 auto，那地图就占满 DIV。

```
var options = {
                controls: [],
                maxExtent: bounds,
                maxResolution: "auto",
                projection: "EPSG:4326",
                numZoomLevels: 7,
                units: 'degrees'
            };
            map = new OpenLayers.Map('map',options);
```

## 2.3　地图控制－尺度缩放

目标：填加尺度缩放控件

步骤：
（1）　map 初始化赋参数

```
var options = {
                controls: [],
                //scales: [50000000, 30000000, 10000000, 5000000],
                maxExtent: bounds,
                maxResolution: "auto",
                projection: "EPSG:4326",
                numZoomLevels: 7,
（表示有几个缩放级别）
                units: 'degrees'
            };
        map = new OpenLayers.Map('map',options);
```

（2）　填加控件，代码

```
map.addControl(new OpenLayers.Control.PanZoomBar({
                    position: new OpenLayers.Pixel(2, 15)（右边距，上边距）
                }));
```

**思考：**级别的计算，个人推测由(maxResolution- minResolution)/ numZoomLevels，但是默认值是书面日后再细究。

### 2.4 地图控制－鼠标坐标拾取

目标：地图上鼠标移动拾取
步骤：
*map.addControl(new OpenLayers.Control.MousePosition());*

**注1. Control 的构造函数可以带参数，var control = new OpenLayers.Control({div: myDiv});例如：map.addControl(new OpenLayers.Control.MousePosition({element: $('location')}));**
**就是制定在页面的 location 元素位置显示坐标。**
**注2. 2**
**注3. 2**

### 2.5 地图控制－其他几个常用控件初次、初级使用

**初次使用，就只写下代码与作用，至于参数以后用到进行研究。**
（1） 鼠标拖动、滚轴放大缩小，自带一个拉框放大。
*map.addControl(new OpenLayers.Control.MouseToolbar());*
（2） 图层控制
*map.addControl(new OpenLayers.Control.LayerSwitcher({'ascending':false}));*
（3） 填加永久链接
*map.addControl(new OpenLayers.Control.Permalink());*
（4） 鹰眼窗口
*map.addControl(new OpenLayers.Control.OverviewMap());*
（5） 默认的键盘操作支持，比如 pageup、→等
*map.addControl(new OpenLayers.Control.KeyboardDefaults());*

# 1 Openlayers 关于数据加载

### 1.1 GML

目标：加载 GML 图层

步骤：

*gmlLayer=new OpenLayers.Layer.GML("GML",*
*"gml/polygon.xml",{isBaseLayer: true} );*
*//map.addLayers([untiled,gmlLayer]);*

*map.addLayer(gmlLayer);*

**注1.　　isBaseLayer 属性确定该图层是否是基础图层。**

### 1.2　　其他类型数据加载

目前还不需要深究，暂不研究。

# 2　　要素编辑

关于 openlayers 里的要素编辑，就是客户端的操作，因此分为操作和保存两个环节研究。

### 2.1　　关于矢量要素的填加

分为点、线、多边形

以一段例子来说明要素的填加。

```
<html xmlns="http://www.w3.org/1999/xhtml">
   <head>
      ……
      <script src="OpenLayers.js"></script>
      <script type="text/javascript">
            var map, drawControls;
            OpenLayers.Feature.Vector.style['default']['strokeWidth'] = '2';
//*****************************************************************
**********
```

① 初始化函数，填加一个基础图和一个矢量图层

```
//*****************************************************************
**********
function init(){
                  map = new Ope —nLayers.Map('map');
                  var wmsLayer = new OpenLayers.Layer.WMS(
                        "OpenLayers WMS",
                        "http://labs.metacarta.com/wms/vmap0",
                        {layers: 'basic'}
```

```
                );
                var vectors = new OpenLayers.Layer.Vector("Vector Layer");
                map.addLayers([wmsLayer, vectors]);
//**********************************************************************
```

② 定义一个 Control 参数对{point:画要素点,line:画线,polygon:画多边形,select:要素选择,selecthover:要素选择}一

```
//**********************************************************************
                drawControls = {
                    point: new OpenLayers.Control.DrawFeature(
                        vectors, OpenLayers.Handler.Point
                    ),
                    line: new OpenLayers.Control.DrawFeature(
                        vectors, OpenLayers.Handler.Path
                    ),
                    polygon: new OpenLayers.Control.DrawFeature(
                        vectors, OpenLayers.Handler.Polygon
                    ),
                    select: new Open ———Layers.Control.SelectFeature(
                        vectors,
                        {
                            clickout: false, toggle: false,
                            multiple: false, hover: false,
                            toggleKey: "ctrlKey", // ctrl key removes from selection
                            multipleKey: "shiftKey", // shift key adds to selection
                            box: true
                        }
                    ),
                    selecthover: new OpenLayers.Control.SelectFeature(
                        vectors,
                        {
                            multiple: false, hover: true,
                            toggleKey: "ctrlKey", // ctrl key removes from selection
                            multipleKey: "shiftKey" // shift key adds to selection
                        }
                    )
                };
//**********************************************************************
```

```
**********
```

③ 控制动作选择

```
//**********************************************************************
**********
                for(var key in drawControls) {
                    map.addControl(drawControls[key]);
                }
                map.setCenter(new OpenLayers.LonLat(0, 0), 3);
            }
            function toggleControl(element) {
                for(key in drawControls) {
                    var control = drawControls[key];
                    if(element.value == key && element.checked) {
                        control.activate();
                    } else {
                        control.deactivate();
                    }
                }
            }
//**********************************************************************
**********
```

④ 控制撤销选择动作

```
//**********************************************************************
**********
            function update() {
                var clickout = document.getElementById("clickout").checked;
                drawControls.select.clickout = clickout;
                var hover = document.getElementById("hover").checked;
                drawControls.select.hover = hover;
                drawControls.select.box                                    =
document.getElementById("box").checked;
                if(drawControls.select.active) {
                    drawControls.select.deactivate();
                    drawControls.select.activate();
                }
            }
        </script>
    </head>
    <body onload="init()">
        <h1 id="title">OpenLayers Select Feature Example</h1>
        <p id="shortdesc">
            Select a feature on hover or click with the Control.SelectFeature on a
```

```
        vector layer.
    </p>
    <div id="map" class="smallmap"></div>
    <ul id="controlToggle">
        <li>
            <input type="radio" name="type" value="none" id="noneToggle"
                onclick="toggleControl(this);" checked="checked" />
            <label for="noneToggle">navigate</label>
        </li>
        <li>
            <input type="radio" name="type" value="point" id="pointToggle"
                onclick="toggleControl(this);" />
            <label for="pointToggle">draw point</label>
        </li>

        。。。。。。
  </body>
</html>
```

**总结：**对于要素填加基本可以理解为以下几步。

① 定义 *control:OpenLayers.Control. SelectFeature*

② Map.addControl

③ Control.active()

解释几个方法。

- OpenLayers.Control.DrawFeature （ layer{OpenLayers.Layer.Vector} ， handler {OpenLayers.Handler}，options {Object}）
- 为图层填加风格

```
var myStyles = new OpenLayers.StyleMap({
                    "default": new OpenLayers.Style({
                        pointRadius: "${type}", // sized according to type attribute
                        fillColor: "#000000",
                        strokeColor: "#ff9933",
                        strokeWidth: 2
                    }),
                    "select": new OpenLayers.Style({
                        fillColor: "#66ccff",
```

```
                    strokeColor: "#3399ff"
                })
        });
        // Create a vector layer and give it your style map.
        var points = new OpenLayers.Layer.Vector(
                'Points', {styleMap: myStyles}
        );
```

## 1.1 矢量要素选择

在4.1节的源码中已经涉及

对于要素选择基本可以理解为以下几步。

① 定义 *control:OpenLayers.Control.DrawFeature*

*分为 hover 和非 hover*

```
select: new OpenLayers.Control.SelectFeature(
                vectors,
                {
                    clickout: false, toggle: false,
                    multiple: false, hover: false,
                    toggleKey: "ctrlKey", // ctrl key removes from selection
                    multipleKey: "shiftKey", // shift key adds to selection
                    box: true
                }
        ),
        selecthover: new OpenLayers.Control.SelectFeature(
                vectors,
                {
                    multiple: false, hover: true,
                    toggleKey: "ctrlKey", // ctrl key removes from selection
                    multipleKey: "shiftKey" // shift key adds to selection
                }
        )
```

② Map.addControl

③ Control.active()

## 1.2 矢量要素拖动

对于要素拖动基本可以理解为以下几步。

① 定义 control: OpenLayers.Control.DragFeature(图层名称)

② Map.addControl

③ Control.active()

## 1.3 获取矢量要素信息

对于获取要素信息基本可以理解为以下几步。

① 选中要素，就获取了 Feature 对象。对于填加要素的同时，也是选中的一种情况。

② 应用 Feature 的有关属性获取几何或属性信息。

一个比较综合的例子，编辑要素、获取 GML 要素属性信息

***注：feature.attributes.name:获取 gml 中要素的属性值。***

```
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>Open Popup on Layer.Vector</title>
    <link rel="stylesheet" href="theme/default/style.css" type="text/css" />
    <link rel="stylesheet" href="style.css" type="text/css" />
    <style type="text/css">
        #controlToggle li {
            list-style: none;
        }
    </style>
    <script src="lib/OpenLayers.js"></script>
    <script type="text/javascript">
            var map, drawControls, selectControl, selectedFeature;
                    function onPopupClose(evt) {
                            selectControl.unselect(selectedFeature);
                    }
```

```
/*****************************************************************
****

                    ***选择要素动作

                    ***弹出要素信息

*****************************************************************
****/
                    function onFeatureSelect(feature) {
                                        selectedFeature = feature;
                                        popup = new
OpenLayers.Popup.FramedCloud("chicken",

feature.geometry.getBounds().getCenterLonLat(),

null,

"<div style='font-size:.8em'>Feature: " +feature.attributes.name+"<br />Area: " +
feature.geometry.getArea()+"</div>",

null, true, onPopupClose);

                                        feature.popup = popup;
                                        map.addPopup(popup);
                                }
                                function onFeatureUnselect(feature) {
                                        map.removePopup(feature.popup);
                                        feature.popup.destroy();
                                        feature.popup = null;
                                }
        function init(){
                map = new OpenLayers.Map('map');
                var wmsLayer = new OpenLayers.Layer.WMS( "OpenLayers WMS",
                        "http://labs.metacarta.com/wms/vmap0?", {layers: 'basic'});
                var polygonLayer = new OpenLayers.Layer.Vector("Polygon Layer");
                                //var polygonLayer=new
OpenLayers.Layer.GML("GML", "gml/polygon.xml"));
                                var gmlLayer=new OpenLayers.Layer.GML("GML",
"gml/polygon.xml");
                                map.addLayer(gmlLayer);
                map.addLayers([wmsLayer, polygonLayer]);
                map.addControl(new OpenLayers.Control.LayerSwitcher());
                map.addControl(new OpenLayers.Control.MousePosition());
```

```
            selectControl = new OpenLayers.Control.SelectFeature(gmlLayer,
                {onSelect: onFeatureSelect, onUnselect: onFeatureUnselect});
            drawControls = {
                polygon: new OpenLayers.Control.DrawFeature(polygonLayer,
                                OpenLayers.Handler.Polygon),
                select: selectControl
            };

            for(var key in drawControls) {
                map.addControl(drawControls[key]);
            }


            map.setCenter(new OpenLayers.LonLat(0, 0), 3);

        }

/**************************************************************************
****
                    ***选择触发函数

                    drawControls 在 init()中已经定义（polygon,select）

**************************************************************************
****/
                    function toggleControl(element) {
            for(key in drawControls) {
                var control = drawControls[key];
                if(element.value == key && element.checked) {
                    control.activate();
                } else {
                    control.deactivate();
                }
            }
        }
    </script>
  </head>
  <body onLoad="init()">
    <h1 id="title">Open Popup on Layer.Vector</h1>
    <p id="shortdesc">
        Using a Control.SelectFeature, open a popup on click.
    </p>
```

```
    <div id="map" class="smallmap"></div>
    <ul id="controlToggle">
        <li>
            <input type="radio" name="type" value="none" id="noneToggle"
                    onclick="toggleControl(this);" checked="checked" />
            <label for="noneToggle">navigate</label>
        </li>
        <li>
            <input type="radio" name="type" value="polygon"
id="polygonToggle"
                    onclick="toggleControl(this);" />
            <label for="polygonToggle">draw polygon</label>
        </li>
        <li>
            <input type="radio" name="type" value="select" id="selectToggle"
                    onclick="toggleControl(this);" />
            <label for="selectToggle">select polygon on click</label>
        </li>
    </ul>
    <p>It is possible to use the onSelect/onUnselect hooks on the SelectFeature
        to do fun things -- like open a popup.
    </p>
  </body>
</html>
```

# 1 研究 GML 命名空间

目前个人认为，gml 命名空间中的有关内容是空间数据的定义标准，地位重要。

经过网上的学习与代码测试，有关 GML 的 Schema 应用，其实是通常 XML 中 Schema 的应用 schema 规则都是需要专门的代码检验的。

而在目前 openlayers 中的 GML 属性定义可以自己定义，故暂且不考虑

schema 的问题。

# 1 项目实例

开始用 GeoServer+Openlayers 做一个实例项目。需要说明的是由于本人是规划部门的信息化从业者，所以实例是规划管理方面的。

## 1.1 需求描述

（1） 项目名称：用地许可地图发布工具

（2） 简单描述：将用地许可基本信息与示意位置在定位图上发布出来，并提供查询功能。以下开始明确有关细节内容。

（3） 许可基本信息包括名称、建设单位、证号、发证日期、JPG 证书。其他信息没必要，因为 JPG 证书其实已经比较够用。

（4） 查询首先提供属性的查询。

（5） 示意位置采用点位表示。

（6） 定位图首先采用区县图，在工具开发到位的情况下，填加道路、镇、村、街道等数据。

（7） 编辑功能、地图查询是以后的事情。

## 1.2 软件体系结构

（1） B/S

（2） 地图 Server 采用 Geoserver WMS，定位图采用 shape 文件，示意点与属性用 GML 文件，B 端采用 openlayers。

（3） 体系结构示意图

## 1.3 开发分解

（1） 定位图的实现

（2） 示意点与属性 GML 文件标准的制定

Openlayers 展现的实现