

从数据源获取功能信息

当向量层 填充了功能，可以查询数据源以检索特征信息非常有用。OpenLayers提供了向量源类（`ol.source.Vector`）中的一些方法，这些方法使我们能够执行查询，例如找出自定义范围（`getFeaturesInExtent`）内的要素，或返回特定坐标（`getFeaturesAtCoordinate`）处的任何要素以及其他有用的类型。查询。

我们将使用自定义GeoJSON文件中的要素创建一个包含图块栅格图层和矢量图层的地图。多边形要素将代表一个区域内的露营地，每个露营地都有一些可提取以显示的属性。`getFeaturesAtCoordinate` 单击地图时，我们将调用源查询方法，并 `overlay` 在此坐标处存在要素的情况下在其中显示适用的要素信息。

可以在中找到源代码 `ch05/ch05-feature-info-from-source`，这是我们最终得到的屏幕截图：

怎么做...

为了找出如何从数据源检索特征信息，请按照下列步骤操作：

- 1 创建具有OpenLayers依赖关系的HTML文件并 `div` 包含地图。特别是，添加和的标记 `overlay`：

```
<div id="js-overlay" class="overlay">
<ul>
<li><strong id="js-ref"></strong></li>
<li><span id="js-restrictions"></span></li>
</ul>
</div>
```

复制

- 2 创建一个自定义 JavaScript文件和设置 包含来自本地GeoJSON文件的内容的矢量源：

复制

```
var vectorSource = new ol.source.Vector({
  url: 'geojson.json',
  format: new ol.format.GeoJSON({
    defaultDataProjection: 'EPSG:3857'
  })
});
```

3 初始化 map 与栅格瓦片层, 载体层, 和 view :

[复制](#)

```
var map = new ol.Map({
  view: new ol.View({
    zoom: 15, center: [872800, 6065125]
  }),
  target: 'js-map',
  layers: [
    new ol.layer.Tile({source: new ol.source.OSM()}),
    new ol.layer.Vector({source: vectorSource})
  ]
});
```

4 缓存一些可以多次访问的DOM元素:

[复制](#)

```
var overlayElem = document.getElementById('js-overlay');
var featureRefElem = document.getElementById('js-ref');
var featureRestrictionsElem = document.getElementById(
  'js-restrictions'
);
```

5 实例化的实例 overlay , 将其添加到 map , 并设置CSS显示:

[复制](#)

```
var overlay = new ol.Overlay({
  element: overlayElem
});
map.addOverlay(overlay);
overlayElem.style.display = 'block';
```

6 结束于 订阅的 singleclick 地图事件并显示内的特征信息 overlay , 如果合适:

[复制](#)

```
map.on('singleclick', function(event) {
  overlay.setPosition(undefined);
  var features =
    vectorSource.getFeaturesAtCoordinate(event.coordinate);

  if (features.length > 0) {
    overlay.setPosition(event.coordinate);
    featureRefElem.innerHTML = features[0].get('ref');
    featureRestrictionsElem.innerHTML =
      features[0].get('restrictions');
  }
});
```

怎么运行的...

为了简洁起见，我们从此食谱中排除了许多HTML和所有CSS，但是请查看完整实现的源代码。

此食谱的大部分内容将与本书中的早期示例类似，因此让我们来详细了解一下地图点击事件处理程序和所使用的源查询：

复制

```
map.on('singleclick', function(event) {
  overlay.setPosition(undefined);
  var features =
    vectorSource.getFeaturesAtCoordinate(event.coordinate);
```

当用户单击或点击地图时，我们订阅 `singleclick` 由 `map` 对象发布的事件。该 `overlay` 方法传递了一个参数，`undefined` 以便隐藏它，直到我们知道希望显示它为止。

我们利用 `getFeaturesAtCoordinate` 源查询方法来确定在点击事件（`event.coordinate`）的坐标处是否存在某个功能。结果存储在一个变量中，即 `features`。

复制

```
if (features.length > 0) {
  overlay.setPosition(event.coordinate);
  featureRefElem.innerHTML = features[0].get('ref');
  featureRestrictionsElem.innerHTML =
    features[0].get('restrictions');
}
```

尽管我们知道我们的数据在任何给定坐标处均不包含重叠要素，但对于其他数据源，可能会发生这种情况，这就是OpenLayers将查询结果返回为的数组的原因 `features`。我们检查数组的长度是否大于零。如果是这样，我们可以推断出该特定坐标上存在一个要素。

在 `overlay` 被定位在通过坐标 `setPosition` 方法，并且 `overlay` 内容被填充有该特征 `ref` 属性和 `restrictions` 两个通过属性，`get` 方法 `ol.Feature` 继承 `ol.Object`。使用JavaScript `innerHTML` 方法将文本添加到相关的DOM元素中。

我们甚至可以将 `getFeaturesAtCoordinate` 源查询方法与坐标一起使用，其中 功能存在但不存在当前从视口范围可见。它仍然会返回在此位置检测到的功能。

还有更多...

还存在其他强大的数据源查询，例如 `getClosestFeatureToCoordinate` 和

`getFeaturesInExtent`，在开发映射应用程序时可能非常有用。我建议您访问OpenLayers文档 (<http://openlayers.org>) 发现其他有用的方法，并找出这些方法可以为您做些什么

么。

也可以看看

- ▶ 第2章 (/book/web_development/9781785287756/2), **添加栅格图层中的添加WMS图层配方**
(/book/web_development/9781785287756/2)
- ▶ 在**从获取信息WMS服务器**食谱
- ▶ 第3章 (/book/web_development/9781785287756/3), **使用矢量层中的添加GML层配方**
(/book/web_development/9781785287756/3)
- ▶ 第4章 (/book/web_development/9781785287756/4), **添加控件中的侦听矢量层功能的事件配**
方 (/book/web_development/9781785287756/4)

◀ 上一节 (/book/web_development/9781785287756/5/ch05lvl1sec51/measuring-distances-and-a

下一节 ▶ (/book/web_development/9781785287756/5/ch05lvl1sec53/getting-information-from-a

