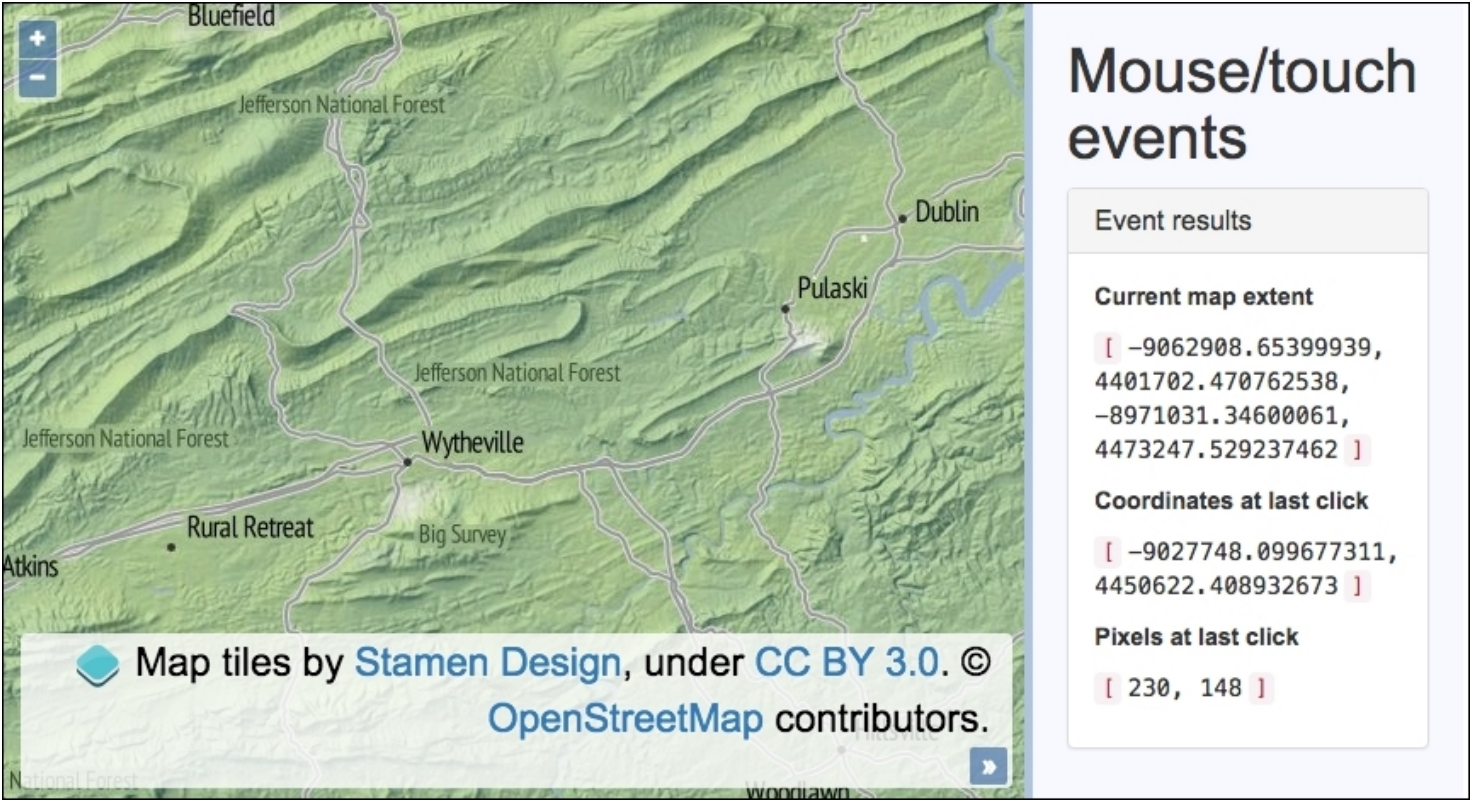


监听鼠标或触摸事件

所以 到目前为止，在本章中，我们已经研究了如何应对 OpenLayers本身发布的事件，例如切片加载事件和矢量要素事件，例如添加的要素。但是，用户驱动的手势以及与地图的交互又如何呢？本食谱介绍了其中一些类型的事件。

我们将演示单击或点击和地图平移事件。当用户单击或触摸地图时，几何形状和像素坐标将显示在侧栏中。平移地图时，新的可见范围也将显示在侧栏中。

可以在中找到源代码 `ch04/ch04-mouse-touch-events` ，这是它的屏幕截图：



怎么做...

至 设置并订阅一些用户驱动的事件，请遵循 这些步骤：

1

创建一个HTML文件，并包含OpenLayers依赖项。特别是，包括 `div` 地图元素和侧面板和内容的标记：

```
<div id="js-map" class="map"></div>
<div class="pane">
  <h1>Mouse/touch events</h1>
  <div class="panel panel-default">
    <h3 class="panel-title">Event results</h3>
    <div class="panel-body">
      <p class="event-heading">Current map extent</p>
      <p>
        <code>[</code>
        <samp id ="js-extent">n/a</samp>
        <code>]</code>
      </p>

      <p class="event-heading">Coordinates at last click</p>
      <p>
        <code>[</code>
        <samp id="js-coords">n/a</samp>
        <code>]</code>
      </p>
    </div>
  </div>
</div>
```

2 缓存一些用于显示的DOM元素，如下所示：

```
var coords = document.getElementById('js-coords');
var pixels = document.getElementById('js-pixels');
var extent = document.getElementById('js-extent');
```

3 使用 view 和栅格图层设置地图：

```
new ol.Map({
  view: new ol.View({
    zoom: 10,
    center: [-9016970, 4437475]
  }),
  target: 'js-map',
  layers: [
    new ol.layer.Tile({source: new ol.source.Stamen(
      {layer: 'terrain'}
    )))
  ]
})
```

4 订阅一些地图事件并更新 侧边栏元素：

```
.on(['click', 'moveend'], function(event) {
  if (event.type === 'click') {
    coords.innerHTML = event.coordinate.join(',<br>');

    var pixelsAtCoords =
      event.map.getPixelFromCoordinate(event.coordinate);
    pixels.innerHTML = [
      pixelsAtCoords[0].toFixed(0),
      pixelsAtCoords[1].toFixed(0)
    ].join(', ');
  } else {
    extent.innerHTML = event.map.getView().calculateExtent(
      event.map.getSize()
    ).join(',<br>');
  }
});
```

怎么运行的...

我们已经使用CSS框架Bootstrap来设置边栏内容的样式。为了简洁起见，一些HTML已被省略，因此请查看本书的源代码以获取完整的实现。

参考下面的代码部分，请注意，在实例化地图之后，我们立即链接该 `on` 方法。映射实例化返回 `map`（的 `ol.Map`）实例，这使我们能够继续进行操作而无需 `map` 在另一个变量中存储对的引用。

复制

```
.on(['click', 'moveend'], function(event) {
  if (event.type === 'click') {
    coords.innerHTML = event.coordinate.join(',<br>');

    var pixelsAtCoords =
      event.map.getPixelFromCoordinate(event.coordinate);
    pixels.innerHTML = [
      pixelsAtCoords[0].toFixed(0),
      pixelsAtCoords[1].toFixed(0)
    ].join(', ');
  }
});
```

我们提供了一个事件数组作为第一个参数该 `on` 方法，即 `click` 和 `moveend` 事件。由于这种组合的事件处理程序，因此在调用处理程序时必须区分事件类型，这就是为什么我们 `event.type` 在执行相关逻辑之前有条件地检查属性的原因。

事件对象包含在中单击或触摸地图的坐标 `event.coordinate`。这是x和y坐标的数组，我们将其转换为字符串，并使用有用的JavaScript `join` 方法将其用逗号和HTML break标记分隔。这将根据需要设置坐标格式，然后通过JavaScript `innerHTML` 方法将其添加到DOM中。

除了几何坐标外，我们还需要事件的像素坐标。事件对象存储对map（ `event.map` ）的引用，为此map对象具有一种将从几何坐标返回像素坐标的方法，称为 `getPixelFromCoordinate` 。此方法需要一个类型为array的数组 `ol.Coordinates` ，因此我们传递事件坐标（ `event.coordinate` ）并将 `ol.Pixel` 类型数组存储到一个变量中，即 `pixelsAtCoords` 。



注意

值得一提的是，事件对象已经在中提供了像素坐标 `event.pixel` 。方便地返回像素值，不带任何小数位。无论如何，这是学习另一个OpenLayers API方法的很好的学习练习。

我们通过更新显示像素坐标的DOM元素完成。为了节省手动的字符串连接，我们建立了一个临时数组，其中包含像素坐标，没有任何小数位（使用 `toFixed` JavaScript方法）。我们还 `join` 再次使用JavaScript方法将数组项与逗号和空格分隔符融合在一起。

复制

```
} else {
  extent.innerHTML = event.map.getView().calculateExtent(
    event.map.getSize()
  ).join(',<br>');
}
```

在同一个处理程序中，如果不是 `click` 事件，那么它一定是 `moveend` 事件。用户完成对地图的平移后，将发布此事件。为了检索可见地图的新范围，我们获取了view（ `event.map.getView` ）的引用并利用 `calculateExtent` view方法。此方法期望感兴趣区域的大小（以像素为单位）。对我们来说，这是可见的整个地图，它是保留在map方法中的 `getSize` 。

的返回的范围数组转换为字符串，定界由逗号和HTML break元素组成，并添加到适用的DOM元素中进行显示。

还有更多...

还有许多地图事件可供抢占，例如不言自明的事件 `pointerdrag` 和 `pointermove` 。要捕获双击或点击，您可以使用 `dblclick` ，并且还有一些事件，用于更改地图的视图（ `change:view` ）和更改地图的大小（ `change:size` ）。这对于响应式设计技术很有用。

我建议您在自己的时间阅读OpenLayers文档，以熟悉所提供的内容。

也可以看看

- ⌕ 所述**变焦到层的程度**的配方在第3章 (/book/web_development/9781785287756/3), **与矢量层工作**
- ⌕ 所述**创建并排侧地图比较**配方
- ⌕ 所述**矢量层听力设有事件**食谱

◀ 上一节 (/book/web_development/9781785287756/4/ch04lvl1sec42/listening-for-the-vector-layer)

下一节 ▶ (/book/web_development/9781785287756/4/ch04lvl1sec44/using-the-keyboard-to-pan-the-map)

