

使用点要素作为标记

记号笔 可以理解为 **兴趣点**（POI） 我们放置一个图标来识别它，并且 该图标具有与之相关的一些信息，例如纪念碑，停车场，桥梁等。

在本食谱中，我们将学习如何将自定义标记图标与点几何类型相关联。可以在中找到源代码 `ch03/ch03-markers/` 。这就是我们的最终结果：



怎么做...

使用以下 学习如何将图标（在这种情况下为标志）附加到步骤 地图上的几何点：

- 1 创建具有OpenLayers依赖关系的HTML文件，并 `div` 包含来包含地图。
- 2 创建一个自定义JavaScript文件；在此文件中，创建一个函数来生成并返回自定义要素样式：

```
var createIconStyle = function(country) {  
  return new ol.style.Style({  
    image: new ol.style.Icon({  
      src: country + '.png'  
    })  
  })  
};
```

3 创建三个几何点要素：

[复制](#)

```
var algeria = new ol.Feature(new ol.geom.Point([146759,3297187]));  
var libya = new ol.Feature(new ol.geom.Point([1927436,3160212]));  
var niger = new ol.Feature(new ol.geom.Point([968610,1986139]));
```

4 将自定义样式分配给每个功能，以便它们包含适用的图标图像：

[复制](#)

```
algeria.setStyle(createIconStyle('algeria'));  
libya.setStyle(createIconStyle('libya'));  
niger.setStyle(createIconStyle('niger'));
```

5 创建一个矢量层并将其添加 features 到矢量中 source：

[复制](#)

```
var vectorLayer = new ol.layer.Vector({  
  source: new ol.source.Vector({  
    features: [algeria, libya, niger]  
  })  
});
```

6 最后，实例化 map，view 和 layers，如下所示：

[复制](#)

```
var map = new ol.Map({  
  view: new ol.View({  
    zoom: 3,  
    center: [1995923, -4167958]  
  }),  
  target: 'js-map',  
  layers: [  
    new ol.layer.Tile({  
      source: new ol.source.MapQuest({layer: 'osm'})  
    }),  
    vectorLayer  
  ]  
});
```

我们选了 居住在三个不同国家/地区的三个几何点 整个非洲。每个POI都有一个代表其各自国家的国旗的图标。

要获得此结果，我们必须使用 `ol.style` OpenLayers对象中的类。让我们仔细看一下这段代码：

复制

```
return new ol.style.Style({
  image: new ol.style.Icon({
    src: country + '.png'
  })
})
```

在我们的 `createIconStyle` 函数中，我们实例化了一个新实例 `ol.style.Style` 。这将返回我们需要的包装对象，以便为矢量要素创建渲染样式。它提供的方法，例如 `getImage` ， `getFill` ， 和 `setZIndex` 。

该 `image` 属性期望使用的类型 `ol.style.Image` ，这是 `ol.style.Icon` 我们在此处使用的基类。该 `ol.style.Style` 班有可用的其他属性，如 `fill` ， `stroke` 和 `text` 。该 `ol.style.Image` 基类提供的方法，例如 `getOpacity` ， `getScale` ， 和 `setRotation` 。

我们将实例 `src` 属性的值动态设置 `ol.style.Icon` 为URL，该URL是根据传入的国家/地区名称构建的。所有图像均为 `.PNG` 格式，因此 `.png` 始终将其串联在一起。

我们已经隐式接受了许多OpenLayers属性默认值 `ol.style.Icon` ，例如 `opacity 1`， `anchor` 例如`[0.5, 0.5]`，它将图标位置居中标记图标的中央（可以相应地自定义以适合图标类型），以及 `anchorXUnits` “**小数**”而不是“**像素**”的类型。我鼓励您检查更多可定制的属性。

复制

```
algeria.setStyle(createIconStyle('algeria'));
```

之后我们 `ol.Feature` 为每个国家/地区创建了一个实例，必须为每个单独的特征设置矢量特征样式，因为它们都使用唯一的图像。在 `ol.Feature class` 提供了 `setStyle` 接受一个实例（或实例的数组）的方法 `ol.style.Style` ，这是从我们的自定义返回真实 `createIconStyle` 的功能。

本食谱的其他部分将设置矢量层和地图，我们都很满意并且不需要解释。

还有更多...

标记当然，可以在将它们最初添加到矢量层后对其进行突变。例如，我们可能希望以编程方式调整属性，例如标志图标之一的不透明度。

让我们通过将第一个标志的不透明度设置为50%来扩展此配方：

复制

```
var vectorLayer = map.getLayers().item(1);
var feature = vectorLayer.getSource().getFeatures()[0];
feature.getStyle().getImage().setOpacity(0.5);
vectorLayer.changed();
```

我们将对矢量层（地图上的第二层）的引用存储在名为的变量中 `vectorLayer` 。然后，我们通过访问矢量层源，从该源中获取所有特征，然后从数组中仅提取第一个特征，来搜索我们要操纵的特征。结果存储在 `feature` 变量中。

我们从功能（ `getStyle` ）中检索样式对象，然后将关注范围缩小到包含标志图标（ `getImage` ）的图像样式对象。我们将不透明度设置为50%。

但是这个 由于矢量层需要重新呈现自身才能使更改生效，因此不会显示任何可见的标志更改。它非常高兴地没有意识到我们的修改，因此我们通过 `changed` 在向量层上手动调用该方法来对其进行友好的调整，该方法随后会递增内部修订计数器并调度一个change事件。OpenLayers通过重新渲染矢量层进行响应，我们最终可以看到第一个标志（阿尔及利亚）现在是半透明的。

也可以看看

- 🔗 在**创建要素编程**食谱
- 🔗 在**添加文本标签几何点**食谱
- 🔗 在**使用集群战略**食谱

◀ 上一节 (/book/web_development/9781785287756/3/ch03lvl1sec30/reading-and-creating-featur

下一节 ▶ (/book/web_development/9781785287756/3/ch03lvl1sec32/removing-or-cloning-featur

