

将要素导出为GeoJSON

在较早的食谱中，我们看到了如何读取不同格式的数据，例如 作为GML和KML，但是从地图导出数据呢？制图应用程序的用户可能想要获取矢量层数据的副本，以便在其他地方使用它。或者，也许您正在保存地图的状态，并且需要将图层内容发送到服务器以进行持久存储，以便以后可以检索相同的几何图形。

在本食谱中，我们将矢量图层的内容导出为GeoJSON格式。可以在中找到源代码 `ch03/ch03-export-geojson` 。这就是我们的最终结果：

怎么做...

我们准备去 启用 `export` 按钮的使用，该按钮将输出矢量图层功能在文本框中显示为GeoJSON。请按照以下步骤实现此目标：

- 1 创建具有OpenLayers依赖项的HTML文件。特别是，创建以下内容：

[复制](#)

```
<div id="js-map"></div>
<div>
  <h1>Exporting GeoJSON</h1>
  <form>
    <button type="submit">Export layer</button>
    <textarea id="js-textarea"></textarea>
  </form>
</div>
```

- 2 创建一个JavaScript文件，并初始化 `map` , `view` 和背景栅格图层：

[复制](#)

```
var map = new ol.Map({
  view: new ol.View({
    zoom: 5,
    center: [2103547, 6538117]
  }),
  target: 'js-map',
  layers: [
    new ol.layer.Tile({
      source: new ol.source.MapQuest({layer: 'osm'})
    })
  ]
});
```

3 创建一种从整数范围中随机选择一个整数的方法，如下所示：

复制

```
var getRandomInt = function(min, max) {
  return Math.floor(Math.random() * (max - min + 1)) + min;
};
```

4 构建随机生成的圆形特征的数组，如下所示：

复制

```
var features = [], numberOfFeatures = 0;

while(numberOfFeatures < 10) {
  var circle = new ol.geom.Circle(
    ol.proj.fromLonLat(
      [getRandomInt(14, 23), getRandomInt(48, 54)]
    ), getRandomInt(4, 15) * 10000
  );
  var polygonCircle = ol.geom.Polygon.fromCircle(circle);
  features.push(new ol.Feature(polygonCircle));
  numberOfFeatures++;
}
```

5 创建 向量层，添加 features 到向量源，并将其添加到 map：

复制

```
var vectorLayer = new ol.layer.Vector({
  source: new ol.source.Vector({
    features: features
  })
});
map.addLayer(vectorLayer);
```

6 将一个侦听器添加到该 export 按钮，并 features 作为GeoJSON 导出到文本框：

复制

```
document.forms[0].addEventListener('submit', function(event) {
    event.preventDefault();
    var format = new ol.format.GeoJSON();
    var features = vectorLayer.getSource().getFeatures();
    var geoJson = format.writeFeatures(features);
    document.getElementById('js-textarea').value = geoJson;
});
```

怎么运行的...

JavaScript包含方法（ `getRandomInt` ），该方法随机选择一个范围内的整数。有关此方法的内部工作原理的更多信息，请访问<http://stackoverflow.com/questions/1527803> (<http://stackoverflow.com/questions/1527803>)。

让我们集中讨论此食谱中使用的新组件：

复制

```
var circle = new ol.geom.Circle(
    ol.proj.fromLonLat(
        [getRandomInt(14, 23), getRandomInt(48, 54)]
    ),
    getRandomInt(4, 15) * 10000
);
```

在 `while` 循环内（执行10次迭代），我们 `circle` 每次都创建一个几何。为了以编程方式创建一组大小不同的圆，我们提供了一系列数字，这些数字表示经度和纬度值，这些值由随机选择 `getRandomInt` 。这些范围在波兰，欧洲周围形成了不准确的边界框。

视图使用的 `EPSG:3857` 投影，因此我们使用 `ol.proj.fromLonLat` 到的经度和纬度值转换为凸台（这是默认，如果它没有被指定为第二个参数）。

我们还随机创建生成的半径，然后将其乘以 `10,000` 使其在起始分辨率下清晰可见。

复制

```
var polygonCircle = ol.geom.Polygon.fromCircle(circle);
```

我们将圆的几何形状转换为规则的多边形特征，从而生成一个近似的圆。这部分可能不明显，但是 GeoJSON（在撰写本文时）尚不支持圆形几何体，因此我们必须将其转换为常规多边形，以便稍后进行序列化和导出。

复制

```
features.push(new ol.Feature(polygonCircle));
```



注意

您可以 `ol.geom.Polygon.fromCircle` 通过传递更多的边作为第二个参数来提高的圆形精度。默认值为32。

然后，将多边形圆转换为 `ol.Feature` 实例，以便OpenLayers可以在图层源上使用它，如下所示。

[复制](#)

```
document.forms[0].addEventListener('submit', function(event) {
    event.preventDefault();
    var format = new ol.format.GeoJSON();
    var features = vectorLayer.getSource().getFeatures();
    var geoJson = format.writeFeatures(features);
    document.getElementById('js-textarea').value = geoJson;
});
```

我们将 `submit` 事件侦听器添加到页面上的第一个也是唯一的表单中，以防止默认表单提交。

我们创建了GeoJSON格式的新实例，并将其存储在 `format` 变量中。可通过矢量层中的有用 `getSource` 方法以及矢量源中的方法从图层源中检索矢量特征 `getFeatures` 。

最后，使用 `writeFeatures` GeoJSON格式提供程序中的方法对GeoJSON进行了序列化，该方法需要一系列OpenLayers功能。然后，GeoJSON字符串显示在 `textarea` 元素内部。

在文本框中显示它是一个想法。您可能决定对此数据执行其他操作，例如通过AJAX将其发布到服务器以进行保存，等等。

也可以看看...

- 所述**添加GML层**配方
- 在**创建要素编程**食谱
- 在**阅读功能，直接使用AJAX**食谱

◀ [上一节 \(/book/web_development/9781785287756/3/ch03lvl1sec28/creating-features-program\)](/book/web_development/9781785287756/3/ch03lvl1sec28/creating-features-program)

[下一节 ▶ \(/book/web_development/9781785287756/3/ch03lvl1sec30/reading-and-creating-featu\)](/book/web_development/9781785287756/3/ch03lvl1sec30/reading-and-creating-featu)

