

## 从WKT读取和创建功能

因为我们已经发现，OpenLayers附带了很多格式帮助器，用于读取/写入不同的文件数据格式。GeoJSON，GML和KML是我们已经探讨过的许多可用格式中的一些。

如果阅读了本章中的**添加GML图层**配方，您将知道矢量层可以读取存储在文件中的要素，指定数据源的格式并将所包含的要素放置在地图中。

对于此配方，我们将以编程方式创建WKT格式的多边形，然后从矢量图层将特征导出为WKT。您可以在下面阅读有关WKT GIS格式的更多信息维基百科 ([http://en.wikipedia.org/wiki/Well-known\\_text](http://en.wikipedia.org/wiki/Well-known_text)) ([http://en.wikipedia.org/wiki/Well-known\\_text](http://en.wikipedia.org/wiki/Well-known_text))。可以在中找到源代码 `ch03/ch03-wkt-format/`。这就是我们的最终结果：

### WKT format

Export layer

```
POLYGON((-82.22044 49.3780339
49.65922 63.5117188,-82.17687.58
31.68084
49.67566.031225319,-82.17572.92
76.25656
49.67527.812711176,-82.16999.64
99.13518
49.67718.905281889,-82.16082.40
55.74095
49.65616.887004048,-82.18260.86
08.80223
49.64890.735235338,-82.20324.66
06.43922 49.65349.357405049))
```

Map tiles by **Stamen Design**, under **CC BY 3.0**. © **OpenStreetMap** contributors.

怎么做...



- 1 创建一个新的具有OpenLayers的HTML文件 依赖性。特别是，添加HTML来创建地图和侧面板，如下所示：

[复制](#)

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

- 2 创建一个自定义JavaScript文件并使用WKT构建多边形几何体：

[复制](#)

```
var wkt = [
  'POLYGON ((' ,
  '-8222044.493780339 4965922.635117188, ' ,
  '-8217687.583168084 4967566.031225319, ' ,
  '-8217572.927625656 4967527.812711176, ' ,
  '-8216999.649913518 4967718.905281889, ' ,
  '-8216082.405574095 4965616.887004048, ' ,
  '-8218260.860880223 4964890.735235338, ' ,
  '-8220324.660643922 4965349.357405049, ' ,
  '))'
];
```

- 3 转换 数组 转换为字符串，如下所示：

[复制](#)

```
var feature = new ol.format.WKT().readFeature(wkt.join(''));
```

- 4 map 使用背景栅格图层和矢量图层创建实例，将其添加 feature 到其中，如下所示：

[复制](#)

```
var map = new ol.Map({
  view: new ol.View({
    zoom: 12,
    center: [-8224433, 4965464]
  }),
  target: 'js-map',
  layers: [
    new ol.layer.Tile({
      source: new ol.source.Stamen({layer: 'terrain'})
    }),
    new ol.layer.Vector({
      source: new ol.source.Vector({
        features: [feature]
      })
    })
  ]
});
```

## 5 听取表单提交并将矢量图层要素导出为WKT:

[复制](#)

```
document.forms[0].addEventListener('submit', function(event) {
    event.preventDefault();
    var wktFormat = new ol.format.WKT();
    var features = map.getLayers().item(1)
        .getSource().getFeatures();
    var wkt = wktFormat.writeFeatures(features);

    document.getElementById('js-textarea').value = wkt;
});
```

## 怎么运行的...

地图创建过程对以前的食谱。对于背景栅格图层，我们使用了Stamen的地形图层，在撰写本文时，该图层仅覆盖美国范围。让我们继续关注WKT OpenLayers代码和功能。

我们创建了一个数组，`wkt` 它构建了一个多边形几何。您可以从WKT创建许多其他几何类型，例如点，线串等。WKT的各个部分已使用数组进行填充，因为它比多行连接的字符串更易于阅读。

[复制](#)

```
var feature = new ol.format.WKT().readFeature(wkt.join(''));
```

该 `readFeature` 方法期望WKT为字符串。当我们将数组（`wkt`）传递给WKT格式方法（`readFeature` JavaScript数组方法）时，您会看到这种情况，方法是 `join` 将它们的项目数组与空白区域连接在一起，从而将我们的项目数组转换为单个字符串。

如预期的那样，我们必须实例化WKT format（`new ol.format.WKT()`）的新实例，该实例返回format对象。我们链接了前面讨论的阅读特征方法。

[复制](#)

```
document.forms[0].addEventListener('submit', function(event) {
    event.preventDefault();
    var wktFormat = new ol.format.WKT();
    var features = map.getLayers().item(1)
        .getSource().getFeatures();
    var wkt = wktFormat.writeFeatures(features);

    document.getElementById('js-textarea').value = wkt;
});
```

作为WKT格式导出的一部分，我们为表单 `submit` 事件附加处理程序，并阻止默认操作。



我们创建一个新的WKT格式对象并将其存储在 `wktFormat` 变量中。然后，我们从矢量层检索多边形要素。为此，我们 `ol.Collection` 通过从地图返回一个对象 `map.getLayers()` 。该集合具有一个称为的有用方法 `item` ，该方法用于选择图层的数组索引。我们知道我们的地图只有两层，其中第二层是矢量层，位于索引1。

我们继续链接另一个名为的函数 `getSource` ，该函数检索矢量层源。最后，我们 `getFeatures` 使用向量源对象中可用的方法获取所有功能。

使用WKT格式化对象（ `wktFormat` ），我们将序列化通过 `writeFeatures` 方法以WKT格式显示OpenLayers功能。文本框的内容已更新为包括导出的WKT。

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/ch03lvl1sec29/exporting-features-as-geojs

下一节 ▶ (/book/web\_development/9781785287756/3/ch03lvl1sec31/using-point-features-as-ma

---

