处理投影

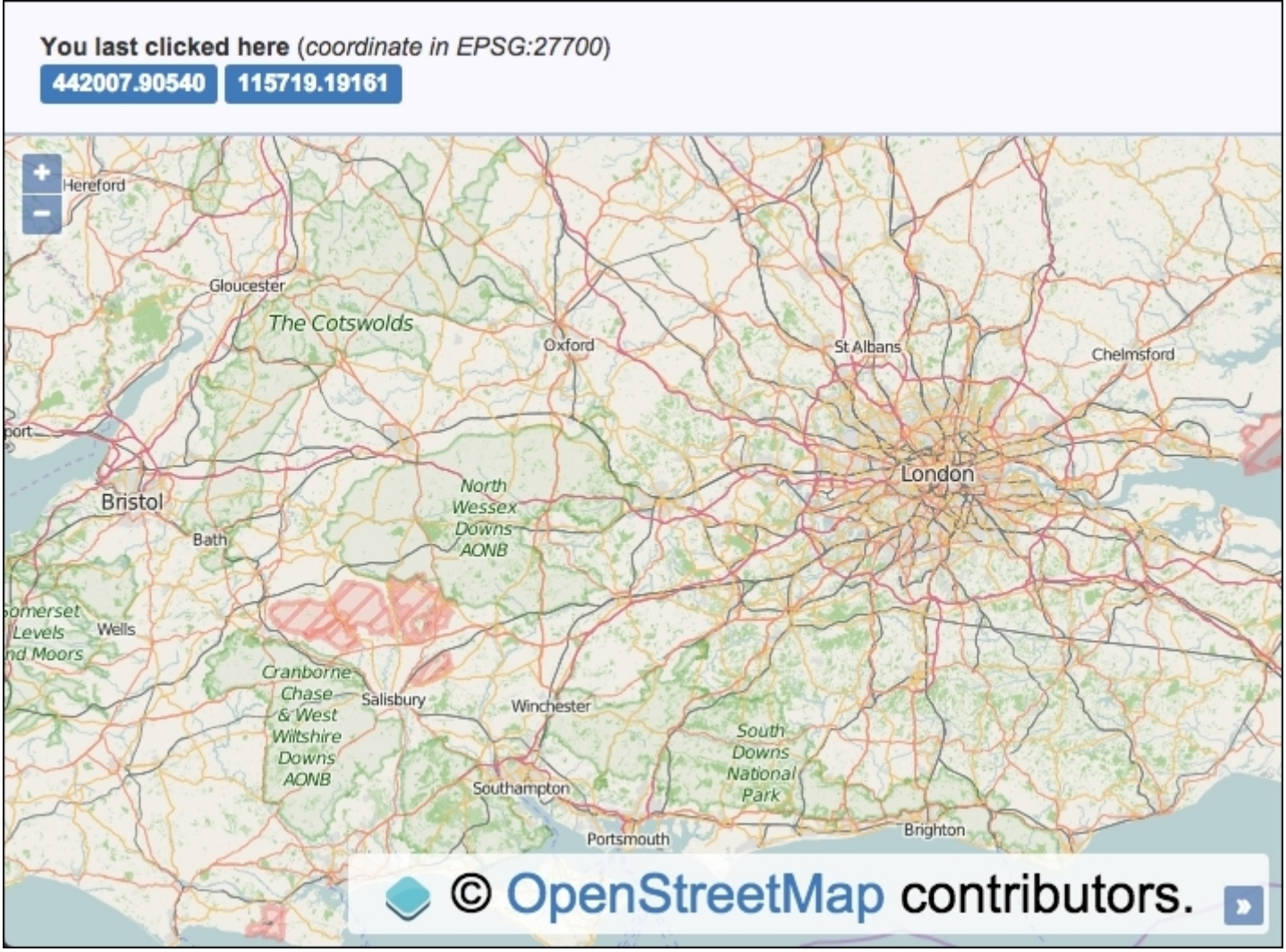
就像我们已经 在本书的先前食谱中了解到，OpenLayers内置了对 `EPSG:4326` 和 `EPSG:3857` 投影的支持。当然，这非常有用，但是当您需要与世界范围内的其他投影一起使用时，它也相当有限。提供更多的内置投影会增加库的膨胀，并且在投影之间进行转换不是一件容易的事。

相反，OpenLayers将此任务外包给其他专用于将投影从一个转换为另一个的库。一个这样的图书馆称为 **Proj4js**（http://proj4js.org (http://proj4js.org)），OpenLayers与之无缝集成。

所以，当我们想比其他预测工作 `EPSG:4326` 和 `EPSG:3857` ，我们期待Proj4js的帮助。

> **注意**
>
> 关于预测的教you超出了本书的范围。EPSG代码只是用于分类和识别大量可用投影的标准化方法。 `EPSG:4326` 对应于 **世界大地测量系统**（**WGS84**）， `EPSG:3857` 是球形墨卡托投影。

对于本食谱，我们将向您展示如何将Proj4js与OpenLayers集成以及如何轻松地使用它。这个想法是创建一个应用程序来显示被点击或点击的位置的坐标。可以在中找到源代码 `ch07/ch07-projections` ，最后得到的内容类似于以下屏幕截图：



做好准备

我们必须 当然，包括Proj4js库，可以是在http://proj4js.org上 (http://proj4js.org)找到。通过网站下载该库的副本，或链接到CDN副本（例如，http://www.cdnjs.com/libraries/proj4js (http://www.cdnjs.com/libraries/proj4js)），或通过其他方式，并将其包含在HTML文件中。

怎么做...

1 创建一个HTML文件，并添加OpenLayers依赖项，Proj4js库（如本食谱的“**准备就绪**”部分中所述）和 `div` 用于保存地图的元素。特别是，添加以下标记以便在单击或轻击后显示坐标：

复制

```
<span id="js-coordX">n/a</span>
<span id="js-coordY">n/a</span>
```

2 创建一个自定义JavaScript文件，并为我们选择的投影定义Proj4js字符串定义，该定义为 `EPSG:27700` ：

复制

```
proj4.defs(
  'EPSG:27700',
  '+proj=tmerc +lat_0=49 +lon_0=-2 ' +
  '+k=0.9996012717 +x_0=400000 +y_0=-100000 ' +
  '+ellps=airy ' +
  '+towgs84=446.448,-125.157,542.06,0.15,0.247,0.842,-20.489 ' +
  '+units=m +no_defs'
);
```

3 创建一个 `extent` 实例，该实例涵盖 `EPSG:27700` 投影范围：

复制

```
var extent = ol.proj.transformExtent(
  [-8.74, 49.81, 1.84, 60.9], 'EPSG:4326', 'EPSG:3857'
);
```

4 使用实例和栅格图层初始化 `map` 实例， `view` 并将视图的 `extent` 范围限制在 `EPSG:27700` 投影有效性的范围内：

复制

```
var map = new ol.Map({
  view: new ol.View({
    zoom: 8, center: [-177333, 6626173], extent: extent
  }),
  target: 'js-map',
  layers: [
    new ol.layer.Tile({source: new ol.source.OSM()})
  ]
});
```

5 缓存 用于显示转换后的坐标的DOM元素：

复制

```
var coordXElem = document.getElementById('js-coordX');
var coordYElem = document.getElementById('js-coordY');
```

6 最后，订阅地图的 `click` 事件， `transform` 坐标，并将其显示在HTML中：

复制

```
map.on('click', function(event) {
  var coordinate = ol.proj.transform(
    event.coordinate, 'EPSG:3857', 'EPSG:27700'
  );
  coordXElem.innerHTML = coordinate[0].toFixed(5);
  coordYElem.innerHTML = coordinate[1].toFixed(5);
});
```

怎么运行的...

正如我们通常所做的那样，为简洁起见，省略了完整的CSS和HTML实施，但是请查看本书的源代码以完整地了解它。

让我们跳入JavaScript，看看如何使用我们选择的投影：

复制

```
proj4.defs(
  'EPSG:27700',
  '+proj=tmerc +lat_0=49 +lon_0=-2 ' +
  ...
);
```

当我们包括我们的网页外Proj4js库，该库暴露了 `proj4` 全球要使用的名称空间。Proj4js尚未意识到我们选择的投影，因此我们必须先对其进行定义。这个特定的投影定义字符串可以在有用的资源http://epsg.io/27700上找到 (http://epsg.io/27700)。该网站还包含许多其他预测；这是书签的重要资源。

或者，您可以从 `script` 以下URL 的标签直接链接到定义设置：http (http://epsg.io/27700.js) ://epsg.io/27700.js (http://epsg.io/27700.js)。现在已经定义了定义（使用库的 `defs` 方法），我们已经成功启用了Proj4js的使用，以便它可以将此投影与其他投影进行相互转换。

复制

```
var extent = ol.proj.transformExtent(
  [-8.74, 49.81, 1.84, 60.9], 'EPSG:4326', 'EPSG:3857'
);
```

注意

请注意，Proj4js确实内置了一些定义，例如 `EPSG:4236` 和 `EPSG:3857` 。但是，如果您需要将投影与其他类型进行相互转换，则也需要定义其他投影。

由于该 `EPSG:27700` 投影不是世界范围的投影，而是仅对英国准确的投影，因此我们设置了覆盖该区域的范围，稍后将其应用于该视图以适当地限制平移。

`extent` 投影有效性的变量在中提供 `EPSG:4326` 。但是，我们的视图将使用 `EPSG:3857` 投影作为默认值，我们希望保留该默认值。为了兼容，我们使用该 `ol.proj.transformExtent` 方法。此方法需要一个 `ol.Extent` 类型为数组的数组，该数组遵循模式[minX，minY，maxX，maxY]。第二和第三个参数分别是源和目标投影。

这成功地为我们提供了 `extent` 在 `EPSG:3857` 投影。

复制

```
map.on('click', function(event) {
  var coordinate = ol.proj.transform(
    event.coordinate, 'EPSG:3857', 'EPSG:27700'
  );
  coordXElem.innerHTML = coordinate[0].toFixed(5);
  coordYElem.innerHTML = coordinate[1].toFixed(5);
});
```

我们订阅了地图 `click` 事件，并将投影中 `event.coordinate` 当前的视图坐标（）转换为 `EPSG:3857` 所需的 `EPSG:27700` 。我们看到我们是 `EPSG:27700` 用Proj4js 定义的，但是这如何与OpenLayers神奇配合?

好吧，OpenLayers 如果需要，可以在内部使用Proj4js代码。当 `ol.proj.transform` 方法被称为首次（通过点击或点）的OpenLayers调用 `ol.proj.get` 方法。 `EPSG:27700` OpenLayers尚不知道我们的投影，因此OpenLayers通过引用代码检查Proj4js库中的定义。如果已经使用Proj4js进行了定义，则会通过 `ol.proj.Projection` 构造函数创建一个新的投影实例。这会将投影内部存储在 `ol.proj.projections` 对象中。这意味着在下次需要使用投影时无需重新实例化该投影。

该 `ol.proj.transform` 方法的工作方式就像我们在前面的食谱中所看到的一样，采用源坐标，源投影以及最终要转换为目标投影的参数。结果存储在 `coordinate` 变量中。

最后，我们将 `coordinate` 变量添加到HTML `5` ，并使用JavaScript `toFixed` 方法将小数位数限制为。

还有更多...

如我们在本食谱中所见，OpenLayers根据需要隐式 `EPSG:27700` 为我们创建了投影对象。但是，有时可能需要使用 `extent` 数组来显式设置投影。当您为 `extent` 投影提供阵列时，OpenLayers可以确定缩放级别0的视图分辨率，这在某些情况下可能非常有用。

这是我们设置 `EPSG:27700` 投影的方法自己（首先使用Proj4js设置定义之后）：

复制

```
var proj27700 = ol.proj.get('EPSG:27700');
proj27700.setExtent([0, 0, 700000, 1300000]);
```

或出于任何原因，如果您想自己调用投影构造函数，则可以执行以下操作：

复制

```
var proj27700 = new ol.proj.Projection({
  code: 'EPSG:27700',
  extent: [0, 0, 700000, 1300000]
});
```

也可以看看

- 在**与地图的选项，播放**配方在第1章 (/book/web_development/9781785287756/1)，**Web制图基础知识**
- 在**创建功能编程**配方在第3章 (/book/web_development/9781785287756/3)，**工作与矢量图层**

上一节 (/book/web_development/9781785287756/7/ch07lvl1sec60/introduction)

下一节 (/book/web_development/9781785287756/7/ch07lvl1sec62/creating-a-custom-control)
