

创建一个简单的全屏地图

当你在地图应用程序中工作时，首要任务是创建地图本身。该地图在您的应用程序中扮演着核心角色，这是您添加和可视化数据的地方。

本食谱将指导您完成创建我们的第一个非常简单的Web地图应用程序的过程。

做好准备

使用OpenLayers进行编程主要归结为编写HTML，CSS，当然还有JavaScript。我们只需要一个文本编辑器即可开始编写我们的食谱。文本编辑器种类繁多，因此请随便选择！

我们的HTML文件将包含一些OpenLayers库资产。尽管您会看到引用这些资产的示例，但在本书中我们不会为您显示这些大文件的文件内容。为了跟进，请先下载最新的OpenLayers源代码 (<http://openlayers.org/download/> (<http://openlayers.org/download/>)) 。

您可以在中找到此示例的源代码 `ch01/ch01-full-screen-map/` 。

怎么做...

- 1 让我们首先创建一个具有以下内容的新HTML文件：

复制

```
<!doctype html>
<html>
<head>
  <meta charset="utf-8">
  <title>Creating a simple full screen map | Chapter 1</title>
  <link rel="stylesheet" href="ol.css">
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <div id="js-map" class="map"></div>
  <script src="ol.js"></script>
  <script src="script.js"></script>
</body>
</html>
```

你会 请注意，链接到此处的OpenLayers文件是 `ol.css` 和 `ol.js` 。我们自己的自定义文件是 `style.css` 和 `script.js` 。

OpenLayers CSS (`ol.css`) 包含CSS3动画和HTML元素的样式，例如地图控件（即地图缩放按钮）等等。

使用最佳做法，OpenLayers JavaScript (`ol.js`) 和我们自己的自定义JavaScript文件已包括在结束 `</body>` 标记之前，以避免阻塞页面呈现。这样做的另一个积极结果是，可以确保在执行JavaScript之前已加载DOM。

2 接下来，创建 `style.css` 具有以下内容的样式表 ()：

复制

```
.map {
  position: absolute;
  top: 0;
  bottom: 0;
  left: 0;
  right: 0;
}
```

这组CSS规则组合导致扩展 `div`，从而完全填充了页面的可用空间。使用 `.map` 类选择器意味着这将针对我们 `<div>` 之前创建的元素：

复制

```
<div id="js-map" class="map"></div>
```



小费

下载示例代码

您可以在<http://www.packtpub.com>上 (<http://www.packtpub.com>)从帐户中下载所有Packt Publishing图书的示例代码文件。如果您在其他地方购买了此书，则可以访问<http://www.packtpub.com/support> (<http://www.packtpub.com/support>)并注册以将文件直接通过电子邮件发送给您。

您可以按照以下步骤下载代码文件：

- 使用您的电子邮件地址和密码登录或注册到我们的网站。
- 将鼠标指针悬停在顶部的“支持”选项卡上。
- 单击代码下载和勘误。
- 在搜索框中输入书籍的名称。
- 选择您要下载其代码文件的书。

➤ 从购买本书的下拉菜单中选择。

➤ 单击代码下载。

下载文件后，请确保使用以下最新版本解压缩或解压缩文件夹：

➤ Windows的WinRAR / 7-Zip

➤ 适用于Mac的Zipeg / iZip / UnRarX

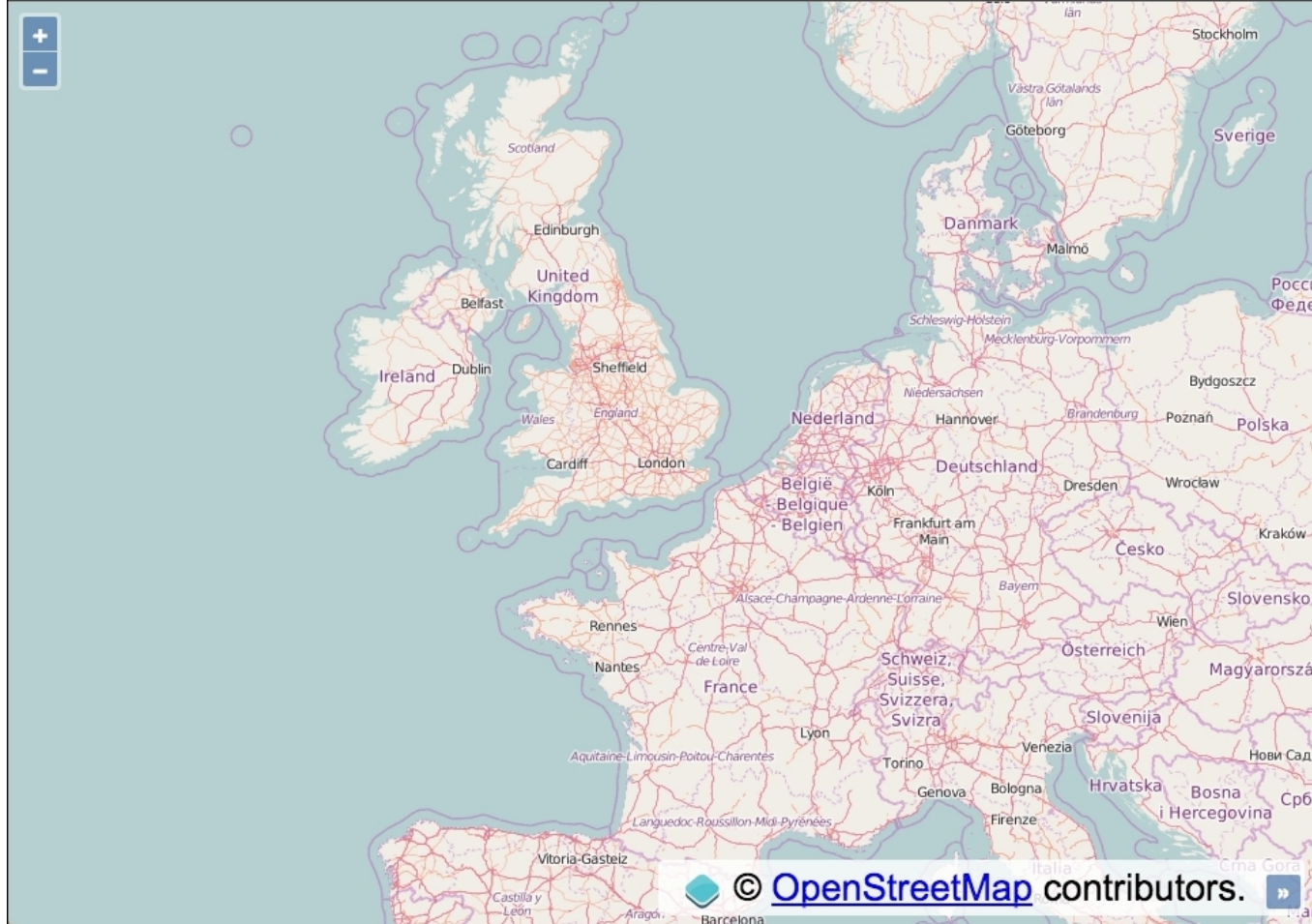
➤ 适用于Linux的7-Zip / PeaZip

3 最后，创建我们的自定义JavaScript文件（ `script.js` ），并将以下内容放入其中：

复制

```
var map = new ol.Map({
  view: new ol.View({
    center: [-15000, 6700000],
    zoom: 5
  }),
  layers: [
    new ol.layer.Tile({
      source: new ol.source.OSM()
    })
  ],
  target: 'js-map'
});
```

在浏览器中打开文件并见证结果。您将看到一张地图，该地图在页面的左上角填充了一些控件，在地图的右下角填充了地图属性，这类似于以下屏幕快照中显示的内容：



怎么运行的...

很高兴意识到使用OpenLayers创建地图可以用最少的代码快速实现。但是，我们并不是为了敬畏而阅读本书，而是希望了解JavaScript是如何做到这一点的。

最初，值得研究HTML，因为OpenLayers一直在忙于修改。您需要打开浏览器开发工具。通常，就像右键单击页面上的任意位置，然后从上下文菜单中选择“**检查元素**”一样简单。向下滚动到 `<div>` 我们最初创建的元素。它看起来应该类似于以下屏幕截图：

```
▼ <div id="js-map" class="map">
  ▼ <div class="ol-viewport" style="position: relative; overflow: hidden; width: 100%; height: 100%;">
    <canvas class="ol-unselectable" width="1197" height="861" style="width: 100%; height: 100%;">
    <div class="ol-overlaycontainer"></div>
    ► <div class="ol-overlaycontainer-stopevent">...</div>
  </div>
</div>
```

您会注意到OpenLayers修改了先前为empty的内容 `<div>`，并插入了一个 `<div class="ol-viewport">` 子元素，该子元素扩展为父元素的总尺寸，我们将其设置为填充屏幕。您可以通过CSS完全控制地图的大小。



在这个生成 `<div>` 的 `<canvas>` 元素构成了您之前看到的地图。HTML5 canvas技术比组装的图像DOM元素性能更高，后者是OpenLayers 2中的默认结构。

出于好奇，请进一步研究其他 `<div>` 元素，您将很快迷失于地图控件的HTML中。与将图像用于地图控件的OpenLayers 2不同，OpenLayers 3仅使用CSS。这意味着自定义地图控件比以前容易得多。

让我们暂时离开HTML，然后将注意力转移到使所有这些都起作用的JavaScript上。我们将逐步检查代码：

复制

```
var map = new ol.Map({  
  // ...  
});
```

该 `ol.Map` 构造是我们的切入点，以创建地图。在实例化上，发生的部分事情涉及创建我们之前查看过的HTML元素。构造函数至少需要一个视图，一个或多个层以及一个作为参数的目标：

复制

```
view: new ol.View({  
  center: [-15000, 6700000],  
  zoom: 5  
}),
```

为了帮助我们理解创建地图所需的单独步骤，让我们想象以下类比。让我们假设地图是一个广阔而风景秀丽的世界，您只能通过双筒望远镜观看，并且 `ol.View` 是双筒望远镜。您可以倾斜头部并旋转（视图旋转），将视线指向其他位置（更改视图中心），并针对一定距离（缩放/分辨率）的不同对象调整焦点。

考虑到这一类比，我们使用双筒望远镜（视图）设置起始位置。中心xy坐标通过数组传递（随着本书的进行，我们将更详细地探讨坐标和投影）。我们还提供了缩放级别。我们有选择地创建了世界的子集视图。

复制

```
layers: [  
  new ol.layer.Tile({  
    source: new ol.source.OSM()  
  })  
],
```

的 `layers` 属性的属性为 `ol.Map` 数组，因为每个地图可以包含多个图层。

该 `ol.layer.Tile` 构造函数的一个子类 `ol.layer.Layer`，但它是专为那些在网格结构，并通过对具体决议缩放级别组织分段的平铺图像设计。

平铺图层的源是从 `ol.source.OSM` 构造函数派生的，这使我们能够轻松使用OpenStreetMap平铺服务。该构造函数是的子类 `ol.source.XYZ` ，它是OSM使用的格式。

复制

```
target: 'js-map'
```

最后，的 `target` 属性 `ol.Map` 可以是字符串（必须表示HTML元素的ID），也可以传入DOM元素。我们的字符串， `'js-map'` 与我们的HTML元素匹配：

复制

```
<div id="js-map" class="map"></div>
```

或者，我们可以传递DOM元素：

复制

```
target: document.getElementById('js-map')
```

既然我们已经涵盖了此难题的所有部分，我们希望您能够对实际发生的事情有更好的了解。这些基本知识将帮助您在我们不断前进的过程中打下坚实的基础。

还有更多...

在第一个示例中，我们用尽了尽可能多的网页，但是我们都 know 这并不是全屏的定义！为了使全屏真正正常运行，OpenLayers可以利用HTML5全屏API。

您可以在中找到此示例的源代码 `ch01/ch01-html5-full-screen-map/` 。

保持HTML和CSS与先前版本完全相同，但是修改JavaScript，使其与以下内容匹配：

复制

```
var map = new ol.Map({
  view: new ol.View({
    center: [-15000, 6700000],
    zoom: 5
  }),
  layers: [
    new ol.layer.Tile({
      source: new ol.source.OSM()
    })
  ],
  controls: ol.control.defaults().extend([
    new ol.control.FullScreen()
  ]),
  target: 'js-map'
});
```

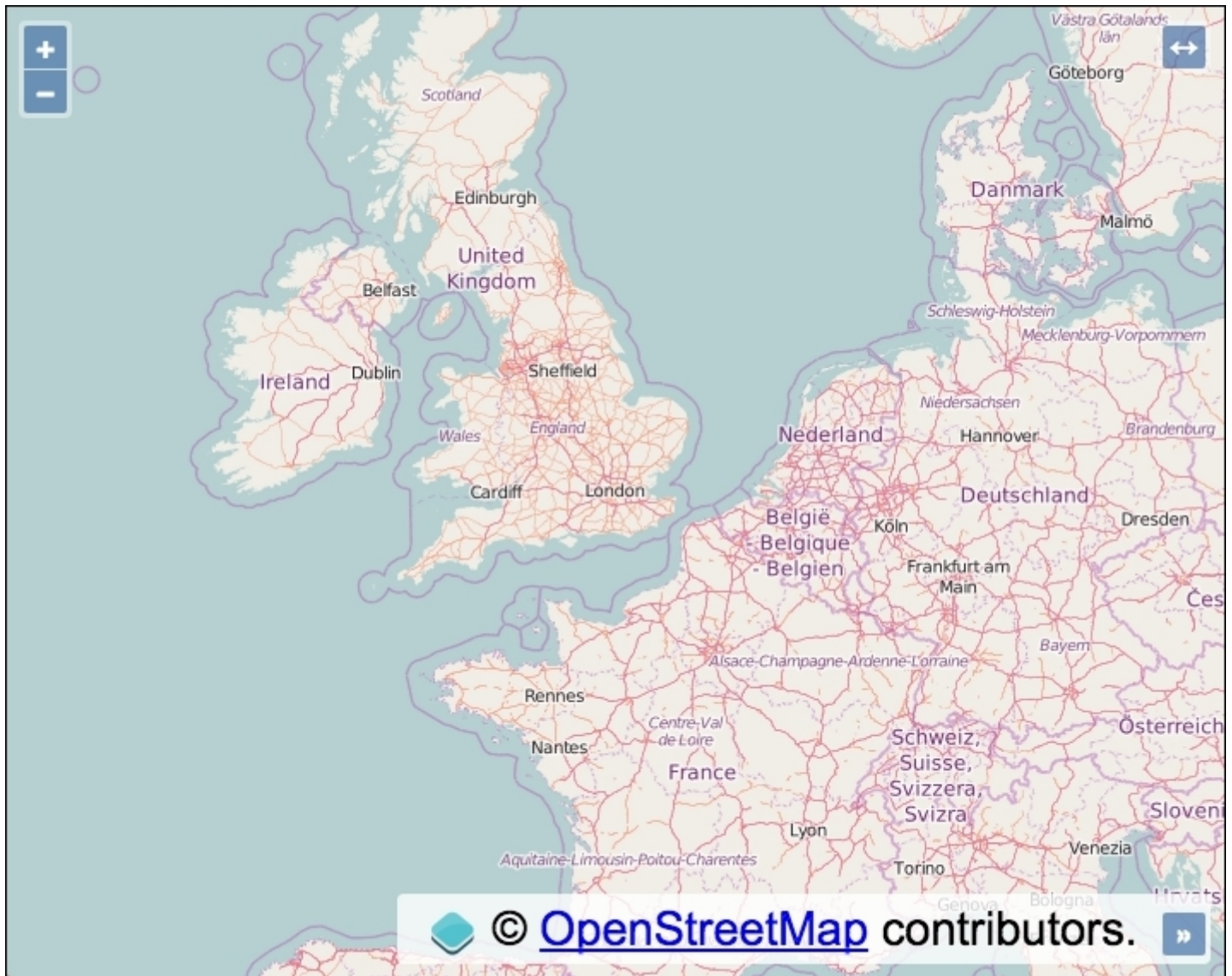

你们中间的警惕 已经注意到，无论我们没有将任何控件传递到地图的先前版本中，还是包含缩放和归因控件。这是因为OpenLayers如果未指定默认控件，则会添加一些默认控件。

复制

```
controls: ol.control.defaults().extend([
  new ol.control.FullScreen()
]),
```

我们决定扩展OpenLayers通常提供的默认控件，并附加全屏控件。扩展实用程序方法来自Google Closure库，该库扩展了一个对象，并放置了另一个对象。

在浏览器中打开文件，您将在地图的右上角看到新的全屏控件。点击按钮进入全屏模式！



如果我们只想启用 全屏控制，我们可以使用以下代码：

复制

```
controls: [
  new ol.control.FullScreen()
]
```

尽管我们只传递一个控件，但OpenLayers希望有一个集合，因此它包装在数组中。

在学习了如何使用一些自定义控件从头开始创建新地图之后，我们完成了本主题。现在该继续下一个主题！

◀ 上一节 (/book/web_development/9781785287756/1/ch01lvl1sec09/introduction)

下一节 ▶ (/book/web_development/9781785287756/1/ch01lvl1sec11/playing-with-the-map-s-opt)

