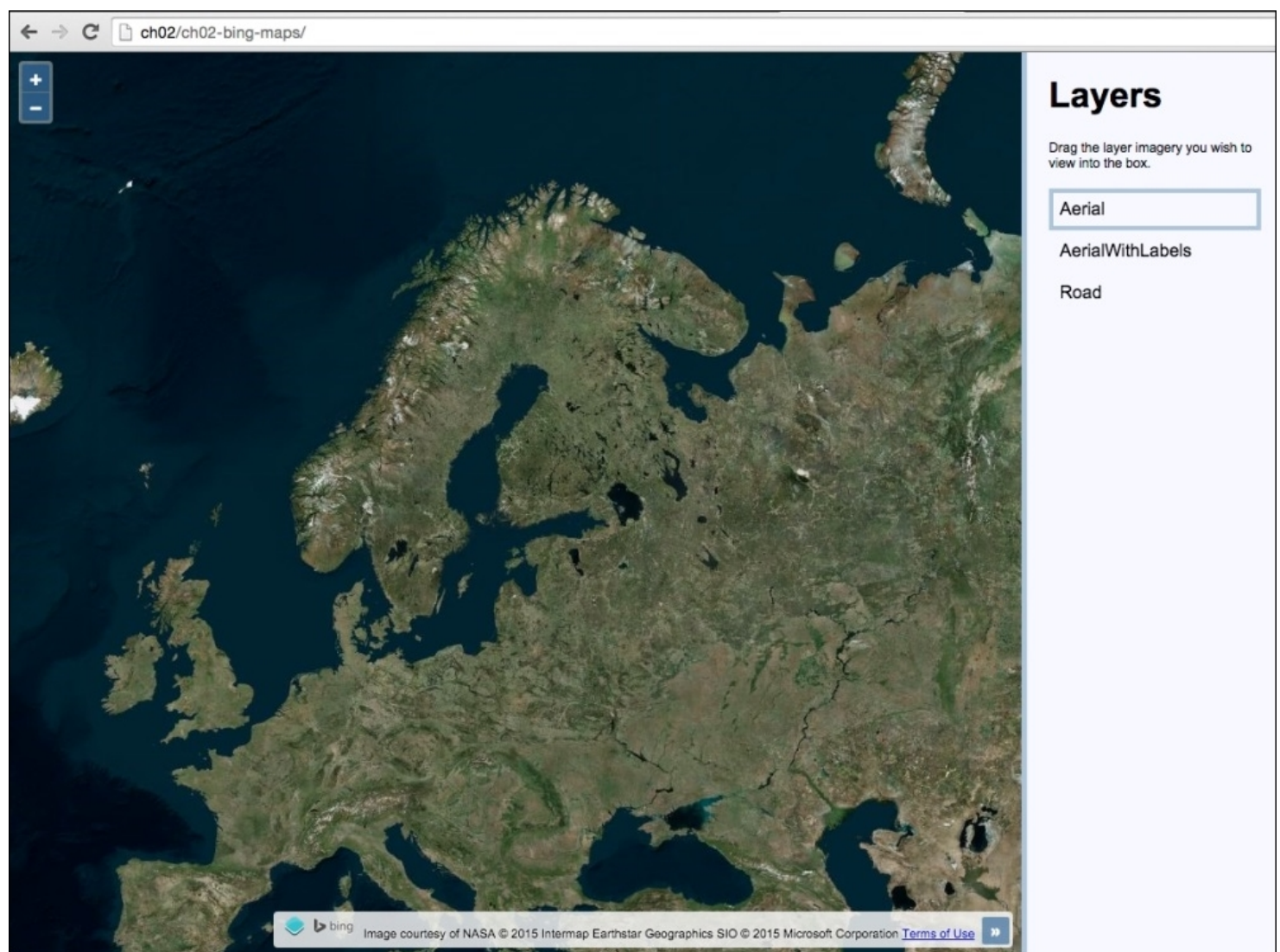


使用必应图像

必应地图是Microsoft提供的地图服务。OpenLayers使与此类平铺服务的集成非常容易
`ol.source.BingMaps`。我们将探索Bing Maps提供的各种图像。

我们要去使用包含可在其中切换的图层列表的面板创建地图。可以在中找到源代码 `ch02/ch02-bing-maps/`。我们最终将获得类似于以下屏幕截图的内容：



做好准备



必应地图 要求您注册为使用者用户才能访问其REST Tile服务。注册后，您将能够查看初始化 OpenLayers Bing Maps源层所需的个人API密钥。您的API密钥用于根据Bing Maps服务对您进行身份验证。



注意

您可以了解如何在<https://www.bingmapsportal.com>上 (<https://www.bingmapsportal.com>)注册API密钥。

除此之外，您还可以要了解Bing Maps提供的图像，请访问<https://msdn.microsoft.com/en-us/library/ff701716.aspx> (<https://msdn.microsoft.com/en-us/library/ff701716.aspx>)。

从这一点开始，假定您有一个API密钥要在以后的代码中使用。

怎么做...

在本部分中，我们将看到如何使用Bing Maps图像。以下是要遵循的步骤：

- 1 创建一个HTML文件并添加OpenLayers依赖项以及jQuery和jQuery UI（负责图层的可排序列表）。
- 2 添加DOM元素以保存地图和图层面板，如下所示：

复制

```
<div id="js-map" class="map"></div>
<div class="pane">
  <h1>Layers</h1>
  <p>Drag the layer imagery you wish to view into the box.</p>
  <ul id="js-layers"></ul>
</div>
```

- 3 在您的自定义JavaScript文件中，创建以下地图实例：

复制

```
var map = new ol.Map({
  view: new ol.View({
    zoom: 4,
    center: [2520000, 8570000]
  }),
  target: 'js-map'
});
```

- 4 存放你的 API密钥转换为变量，如下所示：

复制

```
var apiKey = 'your_api_key';
```

5 创建包含一些Bing图层的图层组并将其添加到地图中：

[复制](#)

```
var layerGroup = new ol.layer.Group({
  layers: [
    new ol.layer.Tile({
      source: new ol.source.BingMaps({
        key: apiKey,
        imagerySet: 'Aerial'
      }),
      title: 'Aerial'
    }),
    new ol.layer.Tile({
      source: new ol.source.BingMaps({
        key: apiKey,
        imagerySet: 'AerialWithLabels'
      }),
      title: 'AerialWithLabels',
      visible: false
    }),
    new ol.layer.Tile({
      source: new ol.source.BingMaps({
```

6 动态地 填充用户界面中的图层列表：

[复制](#)

```
var $layersList = $('#js-layers');

layerGroup.getLayers().forEach(function(element) {
  var $li = $('<li />');
  $li.text(element.get('title'));
  $layersList.append($li);
});
```

7 在用户界面中对图层重新排序时，更新要在地图上显示的图层：

[复制](#)

```
$layersList.sortable({
  update: function() {
    var topLayer = $layersList.find('li:first- child').text();

    layerGroup.getLayers().forEach(function(element) {
      element.setVisible(element.get('title') === topLayer);
    });
  }
});
```

怎么运行的...

HTML和CSS将页面分为两部分：左侧的地图和右侧的薄层切换面板。我们将在这里不再赘述，因为我们希望专注于OpenLayers JavaScript代码：

```
var map = new ol.Map({
  view: new ol.View({
    zoom: 4,
    center: [2520000, 8570000]
  }),
  target: 'js-map'
});
```

我们使用属性 `view` 和创建地图实例 `target` 。我们暂时设置了`layers`属性。

```
var apiKey = 'your_api_key';
```

我们设置了一个变量来存储密钥，即 `apiKey` 。 `'your_api_key'` 用您自己的API密钥替换字符串，这将是一个长随机字符串。

该代码将继续构建一个图层组，其中包含来自Bing Maps的三个不同的图像图层。让我们分别检查这些层：

```
new ol.layer.Tile({
  source: new ol.source.BingMaps({
    key: apiKey,
    imagerySet: 'Aerial'
  }),
  title: 'Aerial'
})
```

每个必应地图 layer是 `ol.layer.Tile` 该类的实例，其源是的实例 `ol.source.BingMaps` 。必填属性为 `key` 和 `imagerySet` 。该图层的图层类型为 `Aerial` ，可提供世界各地令人印象深刻的卫星图像。

我们 `'Aerial'` 为此层设置了自定义标题属性。此名称将显示在UI的图层列表中，以后将用于某些JavaScript逻辑。您会看到，我们为每个Bing Maps图层都赋予了自定义标题以进行识别。

```
new ol.layer.Tile({
  source: new ol.source.BingMaps({
    key: apiKey,
    imagerySet: 'AerialWithLabels'
  }),
  title: 'AerialWithLabels',
  visible: false
})
```

与第一个Bing Maps图层相似，此图层类型为 `AerialWithLabels` 。该图像 `Aerial` 用一些有用的标签覆盖了图像。我们还为该层提供了自定义标题，并将其可见性设置为`false`。这是因为我们只想一次显示一个图层。这样可以确保在看不见图层时，OpenLayers不会发出任何不必要的切片请求。

```
new ol.layer.Tile({
  source: new ol.source.BingMaps({
    key: apiKey,
    imagerySet: 'Road',
    culture: 'en-GB'
  }),
  title: 'Road',
  visible: false
})
```

最终的Bing Maps图层为类型 `Road` 。毫无疑问，该层提供了道路细节，非常适合导航指导。除了（`title` 和 `visible` ）熟悉的属性，我们还用 `'en-GB'` 值设置了新的属性文化。Bing Maps尝试将街道名称本地化为当地文化（如果适用）。因此，如果您要在Great英国（`en-GB`），它将在该层可用的任何地方加载本地化数据。对于其他支持的区域性代码，请访问 <https://msdn.microsoft.com/en-us/library/hh441729.aspx> (<https://msdn.microsoft.com/en-us/library/hh441729.aspx>)。

```
map.addLayer(layerGroup);
```

Bing图层组已添加到地图：

```
var $layersList = $('#js-layers');

layerGroup.getLayers().forEach(function(element) {
  var $li = $('<li />');
  $li.text(element.get('title'));
  $layersList.append($li);
});
```

我们将图层列表缓存（`<ul id="js-layers" class="layers">`） 到一个变量中，即 `$layersList` 。然后，我们遍历图层组的每个图层，并将图层名称动态添加到列表中以进行显示。便捷的 `get` 方法用于获取初始化期间设置的图层的标题。

```
$layersList.sortable({
  update: function() {
    var topLayer = $layersList.find('li:first-child').text();

    layerGroup.getLayers().forEach(function(element) {
      element.setVisible(element.get('title') === topLayer);
    });
  }
});
```

jQuery UI可以对图层列表进行排序。将某项拖到列表中的新位置时，更新事件触发。在事件处理程序中，我们将列表（ `topLayer` ）中顶层的名称缓存。之后，我们遍历地图上的所有图层并显示相应的图层。所有其他层都被隐藏（通过将其可见性设置为false）。我们能够通过它们的title属性链接两组图层。

也可以看看

 所述添加WMS层配方

[◀ 上一节 \(/book/web_development/9781785287756/2/ch02lvl1sec16/introduction\)](/book/web_development/9781785287756/2/ch02lvl1sec16/introduction)

[下一节 ▶ \(/book/web_development/9781785287756/2/ch02lvl1sec18/using-openstreetmap-image\)](/book/web_development/9781785287756/2/ch02lvl1sec18/using-openstreetmap-image)
