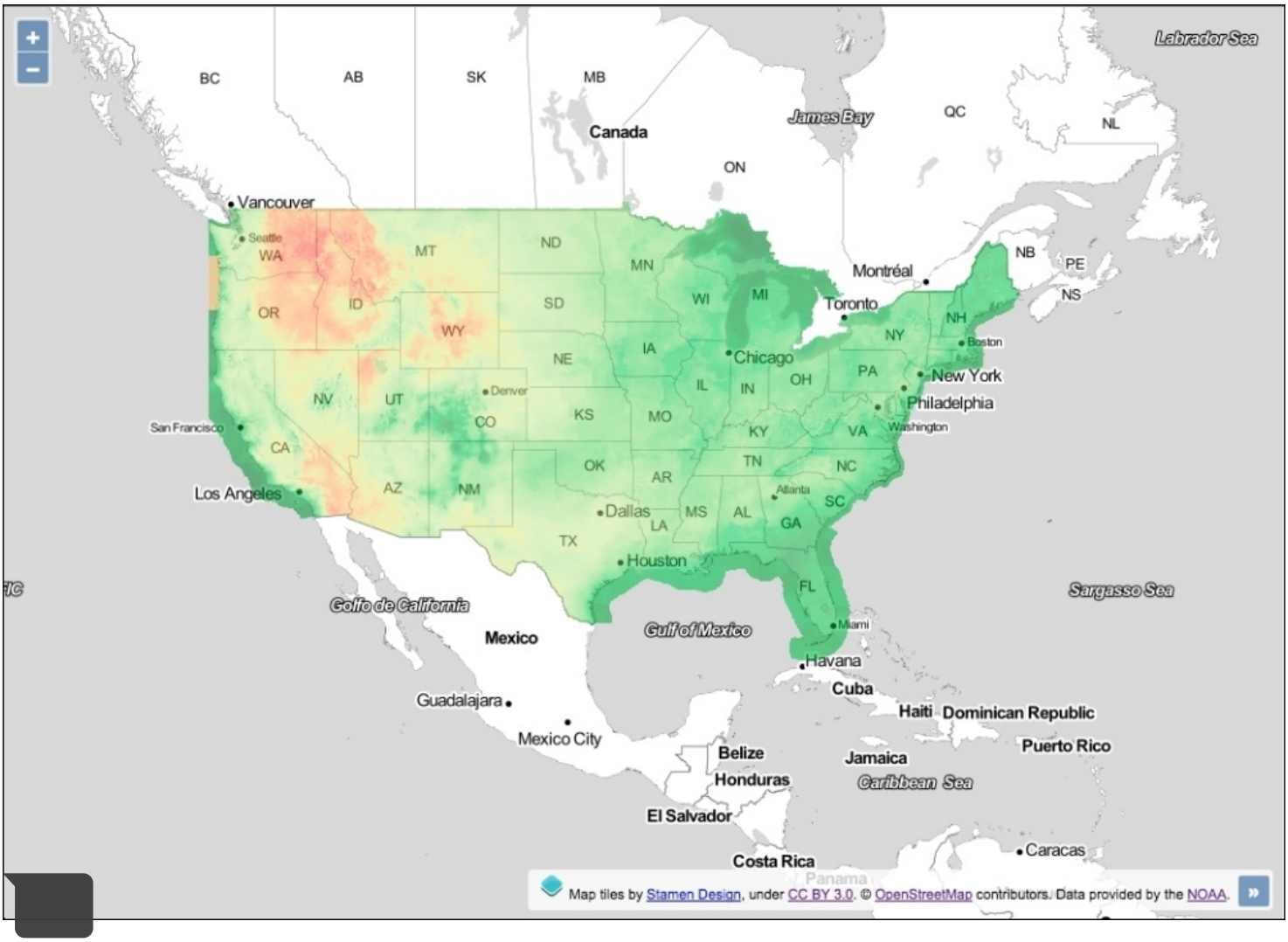
在WMS图层中设置图块大小

设置自定义图块 OpenLayers 3中WMS层的大小有点 比OpenLayers 2更加复杂。它需要使用 `ol.tilegrid.TileGrid` 该类。此类提供了对网格模式的一些灵活控制，该网格模式用于源访问平铺图像服务器。

当然，控制WMS请求的切片大小可能会影响性能。默认情况下，图块大小为**256 x 256**像素，但是我们可以将其设置为其他值。磁贴大小越大，对服务器的请求越少，但生成更大图像的计算时间就越多，每个图像的下载量也就越大。相反，较小的切片大小意味着更多的服务器请求和更少的时间来计算较小的图像。如果使用默认图块大小，则很有可能图像已经被缓存，因此可以提高性能。

无论如何，最好知道如何进行这些调整。可以在中找到源代码 `ch02/ch02-tile-size/` 。这就是我们的最终结果：



怎么做...

为我们的瓷砖设置尺寸 WMS层请求，执行以下操作 脚步：

1. 创建一个具有OpenLayers依赖关系的HTML文件，并 `div` 为Map容器创建一个HTML文件。
2. 创建一个自定义JavaScript文件，并使用Stamen集合中的背景图层初始化地图：

复制

```
var map = new ol.Map({
  view: new ol.View({
    zoom: 0,
    maxZoom: 8,
    center: [-10439500, 4256000]
  }),
  target: 'js-map',
  layers: [
    new ol.layer.Tile({
      source: new ol.source.Stamen({
        layer: 'toner-lite'
      })
    })
  ]
});
```

3. 缓存 `view` 到变量中以供重用，并为我们的自定义WMS层建立分辨率：

复制

```
var view = map.getView();

var resolutions = [view.getResolution()];
for (var i = 1; i < 8; i++) {
  resolutions.push(resolutions[0] / Math.pow(2, i));
}
```

4. 调整的缩放级别 `view` 并创建自定义图块网格图层：

复制

```
view.setZoom(4);

var tileGrid = new ol.tilegrid.TileGrid({
  extent: view.getProjection().getExtent(),
  resolutions: resolutions,
  tileSize: [512, 512]
});
```

5. 最后，创建图块图层，将自定义图块网格应用于图层源，然后将图层添加到地图：

复制

```
map.addLayer(new ol.layer.Tile({
  source: new ol.source.TileWMS({
    url: 'http://gis.srh.noaa.gov/arcgis/services/' +
         'NDFDTemps/MapServer/WMSServer',
    params: {
      LAYERS: 16
    },
    attributions: [
      new ol.Attribution({
        html: 'Data provided by the ' +
              '<a href="http://noaa.gov">NOAA</a>.'
      })
    ],
    tileGrid: tileGrid
  }),
  opacity: 0.50
}));
```

怎么运行的...

我们使用了来自 NOAA提供整个温度的数据 美国。为了给这些数据提供一些背景，背景映射来自雄蕊源提供者，使用称为墨粉-薄层的层。在本节的其余部分，让我们集中讨论主要影响自定义图块大小的代码：

复制

```
var resolutions = [view.getResolution()];
for (var i = 1; i < 8; i++) {
  resolutions.push(resolutions[0] / Math.pow(2, i));
}
```

我们为WMS层建立了自定义分辨率列表。数组存储在名为的变量内部 `resolutions` 。我们已经通过 `getResolution` 视图实例上的方法将数组的初始分辨率预加载到索引0处。当我们以0缩放级别实例化视图时，该初始视图分辨率等于可用的最小分辨率。

我们使用 `for` 循环来填充自定义分辨率列表。新的分辨率从索引1（ `var i = 1` ）添加到数组中，因为索引0已包含最小分辨率。我们已任意决定仅添加其他七个分辨率（ `i < 8` ）。从WMS服务返回的图像没有超出此分辨率的任何附加值，因此将其限制为8的缩放级别是合适的。

当我们遍历 `for` 循环时，我们将计算下一个分辨率。为了打破这种逻辑，让我们举一个人为的例子：假设最小分辨率从20开始。要获得下一个分辨率，我们将其除以2的缩放系数（这是的默认缩放系数 `ol.View` ）。结果为10。要获得下一个分辨率级别，我们再次将10除以2，得到5，依此类推（再进行5次）。

实现此结果的一种简洁方法是通过取幂。起始（最小）分辨率在计算过程中变为恒定值，然后将其除以2（缩放因子）的结果再乘以迭代值。为了证明这一点，我们将继续前面的示例：**20 / (2 x 1) = 10**，然后**20 / (2 x 2) = 5**，依此类推。

最小分辨率始终位于 `resolutions[0]` 数组的索引0（），因此在循环中进行迭代时，我们使用JavaScript `Math.pow` 方法（进行幂运算）来计算下一个分辨率（如前所述），然后将其推入resolutions数组。

复制

```
var tileGrid = new ol.tilegrid.TileGrid({
  extent: view.getProjection().getExtent(),
  resolutions: resolutions,
  tileSize: [512, 512]
});
```

我们将的实例存储 `ol.tilegrid.TileGrid` 到变量中，即 `tileGrid` 。我们设置的第一个属性是 `extent` 。

为了得到适当的 范围，我们首先获取投影对象 与该 `getProjection` 方法一起使用的视图。此方法返回type的投影对象 `ol.proj.Projection` 。视图初始化的默认投影为EPSG：3857。默认情况下，OpenLayers包含EPSG：3857和EPSG：4326的投影对象。

`ol.proj.Projection` 使用投影时，该对象包含一些有用的实用程序方法，其中之一是 `getExtent` 。此方法返回用于自定义网格图层的投影的完整范围。

注意

我们本可以将这一层的范围限制在美国范围内，因为无论如何该数据仅可用于该地区。这将是一个轻微的优化，因为它将避免对超出范围的区域进行不必要的切片请求。

该 `resolutions` 属性随我们自定义的较早创建的分辨率数组一起提供。

该 `tileSize` 属性需要一个 `ol.Size` 类型数组，该数组只是一对表示大小的数字：宽度，然后是高度。我们选择将默认图块大小从**256 x 256倍**增加到**512 x 512**像素。

这完成了我们的定制网格配置，我们稍后将其添加到 `ol.source.TileWMS` 设置中通过 `tileGrid` 财产。

还有更多...

该 `ol.tilegrid.TileGrid` 班允许您设置一组自定义的分辨率（如我们所见），还可以将不同的图块大小与特定的分辨率进行匹配。代替使用该 `tileSize` 属性，可以使用复数 `tileSizes` 属性，并传入一个瓦片大小数组。 `tileSizes` 数组中的项目总数应与resolutions数组的长度匹配。例如，`resolutions` 阵列中的第一个分辨率将使用阵列中的第一个图块大小尺寸 `tileSizes` 。

也可以看看

- 在**创建图像层**配方
- 所述**缓冲层的数据，以改善地图导航**食谱

上一节 (/book/web_development/9781785287756/2/ch02lvl1sec23/creating-an-image-layer)

下一节 (/book/web_development/9781785287756/3)