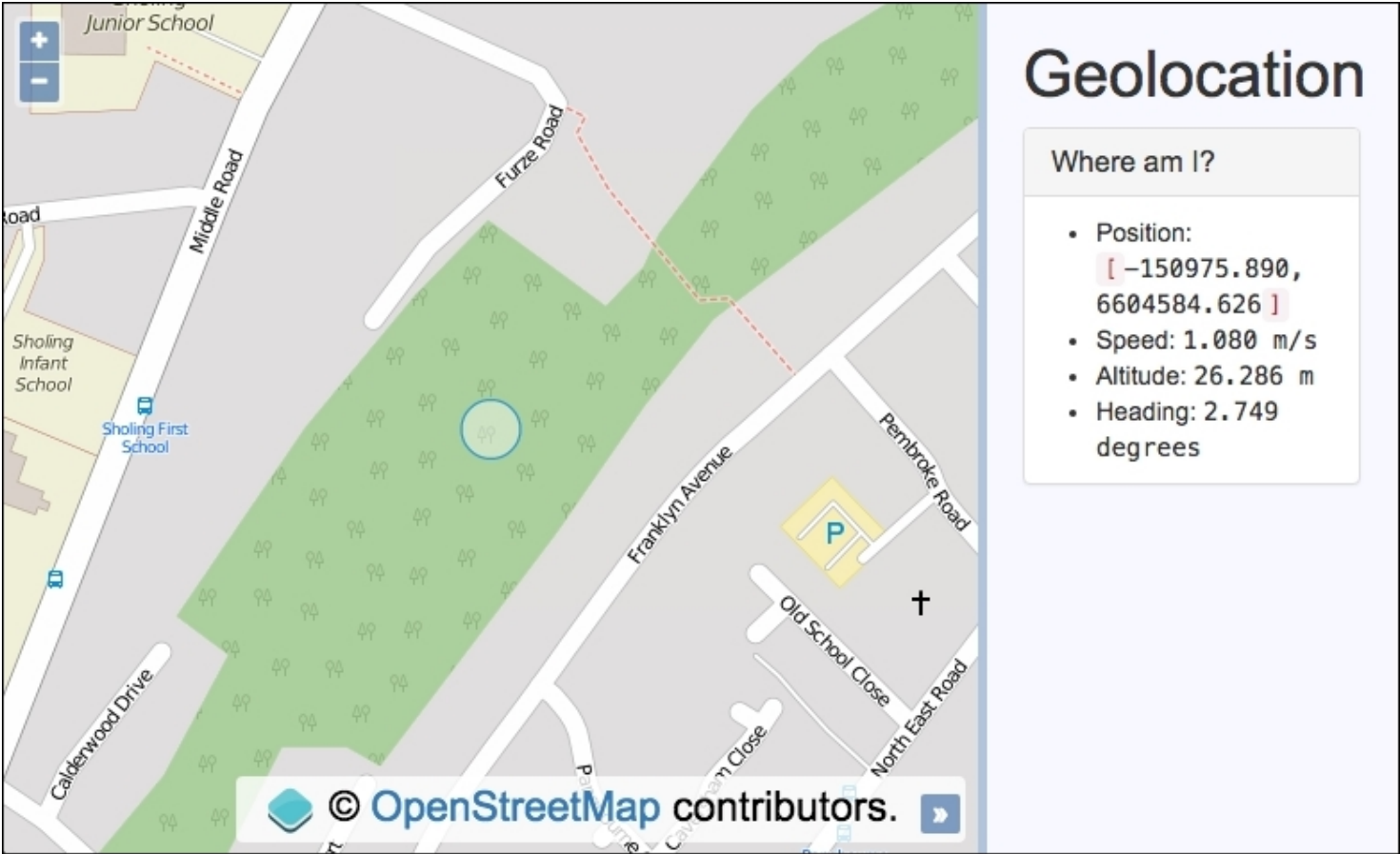


使用地理位置

使用HTML5，本规范中引入了许多新的API和概念，它可以识别正在通过以下方式加载网页的客户端的位置：地理位置API (<http://www.w3.org/TR/geolocation-API/> (<http://www.w3.org/TR/geolocation-API/>))。当然，在Web制图应用程序的世界中，这开辟了新的巨大可能性。

在本食谱中，我们将向您展示如何轻松识别用户的当前位置并将地图的视口居中。的源代码可以在中找到 `ch05/ch05-geolocation` 。这是我们将产生的屏幕截图：



当用户加载网页时，地图的视口将移动到当前用户的位置，并且圆心将放置在中心。如果可用，我们还将显示其他位置指标，例如速度和海拔高度。

要在自己的设备上快速查看此食谱，请访问<https://jsfiddle.net/pjlangley/9yyLovt5/> (<https://jsfiddle.net/pjlangley/9yyLovt5/>)的实时副本。

在访问地图时启用了地理位置的浏览器，您可能需要手动允许地理位置请求，以使其正常工作。大多数浏览器供应商都会在这里明显提示您采取行动。对于移动设备，您可能需要检查您的隐私设置，以确保在您的移动浏览器中可以使用位置服务。

怎么做...

要了解如何在OpenLayers中使用Geolocation API，请按照以下说明进行操作：

- 1 首先，创建具有OpenLayers依赖项的HTML文件并 `div` 保存地图。特别是，以下是构成将要填充的地理位置指标的标记：

复制

```
<ul>
<li>Position:
<code>[</code>
<samp id="js-position">n/a</samp>
<code>]</code></li>
<li>Speed: <samp id="js-speed">0</samp></li>
<li>Altitude: <samp id="js-altitude">n/a</samp></li>
<li>Heading: <samp id="js-heading">n/a</samp></li>
</ul>
```

- 2 创建一个JavaScript文件并缓存一些DOM元素以供重用，以及OpenLayers循环功能：

复制

```
var locationCircle = new ol.Feature();
var positionElem = document.getElementById('js-position');
var speedElem = document.getElementById('js-speed');
var altitudeElem = document.getElementById('js-altitude');
var headingElem = document.getElementById('js-heading');
```

- 3 `map` 使用 `view`，带有栅格特征的栅格图层和矢量图层实例化：

复制

```
var map = new ol.Map({
  view: new ol.View({
    zoom: 17,
    center: [10030840, 6731350]
  }),
  target: 'js-map',
  layers: [
    new ol.layer.Tile({
      source: new ol.source.OSM()
    }),
    new ol.layer.Vector({
      source: new ol.source.Vector({
        features: [locationCircle]
      })
    })
  ]
});
```

- 4 设置地理位置选项，订阅位置更改事件，执行必要的逻辑以重新放置地图和圆的位置，并更新边栏显示，如下所示：

[复制](#)

```
new ol.Geolocation({
  projection: map.getView().getProjection(),
  tracking: true,
  trackingOptions: {
    enableHighAccuracy: true
  }
})
.on('change', function() {
  var position = this.getPosition();
  var speed = this.getSpeed();
  var altitude = this.getAltitude();
  var heading = this.getHeading();

  map.getView().setCenter(position);

  locationCircle.setGeometry(
    new ol.geom.Circle(position, 20)
  );
});
```

怎么运行的...

在Bootstrap CSS框架的帮助下，我们对边栏进行了样式设置，以显示地理位置信息。请查看源代码以获取完整的实现，为简洁起见，我们将其省略。

我们通过缓存对将显示数据的DOM元素的引用来启动JavaScript。这是高性能的，因为在整个代码中都会经常访问它们。圆特征也存储在变量中，即 `locationCircle`，随着几何图形的频繁更新。

地图实例化对于早期的食谱来说看起来很熟悉，因此让我们深入了解地理位置功能背后的OpenLayers代码：

[复制](#)

```
new ol.Geolocation({
  projection: map.getView().getProjection(),
  tracking: true,
  trackingOptions: {
    enableHighAccuracy: true
  }
})
```

OpenLayers有一个HTML5 Geolocation API帮助程序类，即 `ol.Geolocation`。该帮助程序类使用我们的 `window.navigator.geolocation.watchPosition` 方法来设置和订阅地理位置更改，从而代表我们与JavaScript API进行交互。

使用HTML5技术时，最好先确认可用的浏览器支持范围。的网站<http://caniuse.com> (<http://caniuse.com>) 是此类信息的有用资源。

实例化时 `ol.Geolocation`，我们指定一些配置属性。我们通过传入视图的当前投影来告知 OpenLayers 投影。通过设置 `tracking` 为 `true`，我们会在页面加载后自动开始跟踪用户的位置（只要用户允许）。除此之外，我们还提供了一个对象，其 `enableHighAccuracy` 设置为 `true`。此设置要求从应用程序获得最佳结果。请注意，在移动设备上，这可能会消耗更多的功率和带宽。

参考其他清单 地理位置API的可用位置选项位于http://www.w3.org/TR/geolocation-API/#position_options_interface (http://www.w3.org/TR/geolocation-API/#position_options_interface)。

复制

```
.on('change', function() {
  var position = this.getPosition();
  var speed = this.getSpeed();
  var altitude = this.getAltitude();
  var heading = this.getHeading();
```

我们将 `on` 方法链接到地理位置实例化的末尾，并订阅通用更改事件。这是一个包罗万象的事件监听器，将最新的响应 `position`，`speed` 和 `altitude` 更新，很多其他的变化之中。

地理位置帮助器类提供了许多 `get` 方法来检索最新值。我们将这些值存储在变量中，例如 `speed` 包含的结果 `this.getSpeed()`。在这种情况下，`this` JavaScript关键字引用了我们的地理位置实例。

复制

```
map.getView().setCenter(position);
locationCircle.setGeometry(
  new ol.geom.Circle(position, 20)
);
```

有了最新的坐标，我们相应地使地图视图居中，并使用这些相同的坐标更新矢量图层上的圆形要素的几何形状。该 `ol.geom.Circle` 构造函数将 `ol.Coordinate` 阵列和半径作为参数。特征方法会使用生成的几何体 `setGeometry`，以更新圆的位置。

复制

```
positionElem.innerHTML = position.join(',<br>');
speedElem.innerHTML = speed ? speed.toFixed(3) + ' m/s' : 'n/a';
altitudeElem.innerHTML =
  altitude ? altitude.toFixed(3) + ' m' : 'n/a';
headingElem.innerHTML =
  heading ? heading.toFixed(3) + ' degrees' : 'n/a';
```

最后，我们用所有最新读数更新侧边栏。类型的`position`数组 `ol.Coordinate` 是通过JavaScript `join` 方法传递的，因此，数组条目以字符串形式显示，中间有逗号和HTML标记 `
`。

的 `speed` , `altitude` 和 `heading` 值是固定在 3 小数位 (`toFixed(3)`) 和测量的单元被附加到字符串。如果地理位置类型没有可用的值, 则使用 `'n/a'` 字符串代替。的 `innerHTML` JavaScript方法填充的DOM元素。

根据Geolocation API规范 (http://www.w3.org/TR/geolocation-API/#coordinates_interface (http://www.w3.org/TR/geolocation-API/#coordinates_interface)) , 单位 `speed` 为米/秒, 单位 `altitude` 为 `heading` 度, 单位为度 (0-360) 。

为了见证最佳结果, 我建议在移动时在移动设备上查看此示例。希望您能够看到所有值在移动时更新。

也可以看看

- 在添加和删除控制配方
- 该改性特征食谱

◀ 上一节 (/book/web_development/9781785287756/5/ch05lvl1sec46/adding-and-removing-controls)

下一节 ▶ (/book/web_development/9781785287756/5/ch05lvl1sec48/placing-controls-outside-the-form)

