

Web前端开发

第21章：地理定位

阮晓龙

13938213680 / rxl@hactcm.edu.cn
<http://web.book.51xueweb.cn>
<http://www.51xueweb.cn>

河南中医药大学管理科学与工程学科

2018.5

本章主要内容

- 常见地理定位方式
- 使用HTML5实现地理定位
- 案例：使用百度地图展示当前位置
- 案例：使用百度地图展示运动轨迹



1.常见地理定位方式

□ IP定位

- IP定位是过去很长一段时间获取用户位置信息的常用方式，其实现原理是：自动查找用户的IP地址，然后检索其注册的物理地址。
- 优点：任何地方都可用、在服务器端处理
- 缺点：不准确（经常出错，一般只能精确到城市级别）、运算代价高（在服务器端处理）
- *如果用户的IP地址是ISP（互联网服务提供商）提供的，则其位置往往就由服务提供商的物理地址决定，因此其定位信息可能距用户位置还有很长一段距离。

1. 常见地理定位方式

□ GPS定位

- GPS (Global Positioning System) 全球定位系统，它通过收集运行在地球周围的多个GPS卫星信号实现地理定位。
- 优点：准确性高
- 缺点：响应时间可能较长、耗电

1.常见地理定位方式

□ Wi-Fi定位

- Wi-Fi定位通常是指使用一个或多个Wi-Fi接入点完成三角定位，其通过计算用户当前位置与已知的多个Wi-Fi接入点的距离，从而实现地理定位。
- 优点：准确、可在室内使用、快速
- 缺点：仅适合静态定位（用户位置保持不变）、定位效果不是很好（Wi-Fi接入点少的地区）

1.常见地理定位方式

□ 手机基站定位

- 基于手机的地理定位是通过用户到电信运营商基站的三角距离确定的，基站越多定位就越准确。
- 优点：准确（依据电信运行商基站数量而定）、可在室内使用、快速
- 缺点：如果你的位置很偏僻，附近只有一个基站，则定位的精度就会出现很大的偏差

2.使用HTML5实现地理定位

2.1浏览器支持性检查

□ 浏览器支持性检查

- Geolocation是HTML5规范的一部分，目前可在几乎所有的主流浏览器中使用Geolocation API实现地理定位。
- W3C组织为Geolocation制定的接口定义：

```
[NoInterfaceObject];  
interface Geolocation {  
    void getCurrentPosition(PositionCallback successCallback, ,  
                            optional PositionErrorCallback errorCallback, ,  
                            optional PositionOptions options);  
    long watchPosition(PositionCallback successCallback, ,  
                       optional PositionErrorCallback errorCallback, ,  
                       optional PositionOptions options);  
    void clearWatch(long watchId);  
};  
callback PositionCallback = void (Position position);  
callback PositionErrorCallback = void (PositionError positionError);
```

2.使用HTML5实现地理定位

2.1浏览器支持性检查

- 浏览器支持性检查
 - 通过下面的代码检查浏览器支持性：

```
function checkLocation() {  
    if (navigator.geolocation) {  
        //浏览器支持 geolocation。  
        alert("浏览器支持地理定位!");  
    } else {  
        //浏览器不支持 geolocation。  
        alert("浏览器不支持地理定位，请升级浏览器!");  
    }  
}
```


2.使用HTML5实现地理定位

2.2位置请求

□ 位置请求

- Geolocation目前只有两种类型的位置请求：单次定位请求以及重复位置更新请求。

2.使用HTML5实现地理定位

2.2位置请求

□ 单次定位请求

- 单次定位请求只请求用户地理位置信息一次，使用 `getCurrentPosition` 方法可以得到用户地理位置信息，其调用方法为：

```
navigator.geolocation.getCurrentPosition(onSuccess, onError, options);
```

- 该方法的三个参数为 `onSuccess`、`onError`、`options`。

2.使用HTML5实现地理定位

2.2位置请求

- onSuccess：浏览器成功获得地理定位信息时调用的函数，因为获取定位信息时可能需要较长的时间完成，为防止在检索位置时浏览器被锁定，或者被暂停（需等待用户同意后才能获得地理位置信息），所以这个函数参数十分重要，它是得到定位信息并进行处理的地方。

- 浏览器获取地理位置成功时调用函数代码如下所示：

```
//成功时。  
function onSuccess(position) {  
    //获取成功时的处理。  
}
```

- 在获取地理位置信息成功时，执行的回调函数中，用到了参数 position，它代表一个 position 对象，通过访问 position 对象的属性即可得到地理位置信息。

2.使用HTML5实现地理定位

2.2位置请求

- position属性
 - latitude: 当前地理位置的纬度。
 - longitude: 当前地理位置的经度。
 - altitude: 当前地理位置的海拔高度, 未获取到时为null。
 - accuracy: 获取到的纬度/经度的精度, 以米为单位。
 - altitudeAccuracy: 获取到的海拔高度精度, 以米为单位, 未获取到时为null。
 - heading: 设备的前进方向, 用面朝正北方向的顺时针旋转角度表示, 未获取到时为null。
 - speed: 设备的前进速度, 以m/s为单位, 未获取到时为null。
 - timestamp: 获取地理位置时的时间。

2.使用HTML5实现地理定位

2.2位置请求

- position主要包含坐标（coords特性）和一个获取位置信息的时间戳，在实际应用中并不一定需要时间戳，重要的位置数据都包含在了coords属性中。

2.使用HTML5实现地理定位

2.2位置请求

- `onError`: 因为位置计算服务很可能出错, 所以对于HTML5 Geolocation应用程序来说错误处理非常重要。HTML5 Geolocation API定义了所有需要处理的错误情况的错误编号。错误编号设置在错误对象中, 错误对象作为`code`参数传递给错误处理程序。

2.使用HTML5实现地理定位

2.2位置请求

- 错误编号
 - PERMISSION_DENIED（错误编号1）：用户选择拒绝浏览器获取其位置。
 - POSITION_UNAVAILABLE（错误编号2）：尝试获取用户位置数据失败。
 - TIMEOUT（错误编号3）：尝试确定用户位置超时。
- 处理地理定位错误信息代码如下所示：

```
function onError(error) {  
    switch (error.code) {  
        case error.PERMISSION_DENIED:  
            alert("用户拒绝获取地理位置的请求");  
            break;  
        case error.POSITION_UNAVAILABLE:  
            alert("尝试获取用户位置数据失败");  
            break;  
        case error.TIMEOUT:  
            alert("请求用户地理位置超时");  
            break;  
    }  
}
```

2.使用HTML5实现地理定位

2.2位置请求

- options: 请求选项options是可选参数, 用来调整HTML5 Geolocation服务的数据收集方式。其声明方式如下所示:

```
var options = {  
  enableHighAccuracy: true,  
  timeout: 10000,  
};
```

- options主要由enableHighAccuracy、timeout和maximumAge三个可选参数组成, 将这三个参数传递给HTML5 Geolocation服务以调整数据收集方式。三个参数可以使用JSON对象传递, 这样便于添加到HTML5 Geolocation请求应用中。

2.使用HTML5实现地理定位

2.2位置请求

■ options参数

- enableHighAccuracy: 布尔值, 如果选择启用, 则将通知浏览器启用HTML5 Geolocation服务的高精确度模式, 默认值为false。
- timeout: 单位为ms, 告诉浏览器计算当前位置所允许的最长时间, 如果在限制时间内未完成定位, 就会调用错误处理程序, 默认值为Infinity, 即无限制一直执行。
- maximumAge: 数值, 单位为ms, 表示浏览器重新计算位置的时间间隔, 默认值为0, 这意味着浏览器每次请求时必须立即重新计算位置。



现场演示:

- 案例21-01: 展示position属性信息

2.使用HTML5实现地理定位

2.2位置请求

□ 重复位置更新请求

- 使用watchPosition方法可以持续获取用户的当前地理位置信息，它会定期的自动获取，其调用方法为：

```
navigator.geolocation.watchPosition(onSuccess, onError, options);
```

- 该方法的三个参数均与getCurrentPosition（单次定位请求）方法的参数说明和使用方法相同。只要用户位置发生变化，Geolocation服务就会调用onSuccess函数，它的效果就像是程序在监视用户的位置，并在其发生变化时及时通知用户一样。

2.使用HTML5实现地理定位

2.2位置请求

□ 停止获取位置信息

- 使用clearWatch方法可以停止当前用户的地理位置信息监视，其调用方法为：

```
navigator.geolocation.clearWatch(watchID);
```

- 该方法的参数为调用watchPosition方法监视地理位置信息时返回的参数，比如使用下面的方法监视地理位置时，就可以使用上面的方法停止获取位置信息。

```
var watchID = navigator.geolocation.watchPosition(onSuccess, onError, options);
```

2.使用HTML5实现地理定位

2.3隐私保护

□ 隐私保护

- 访问使用HTML5 Geolocation API的页面时，会触发隐私保护机制，浏览器会弹出框提示是否允许网站获取你的位置。执行HTML5 Geolocation代码时会触发这一机制，如果仅仅是添加了HTML5 Geolocation代码并未执行，则不会触发隐私保护机制。
- 需要特别注意的是如果想要使用HTML5 Geolocation的地理定位服务，必须将开发的页面通过Web服务器发布出来，并且发布之后必须选择https协议，才能获取到用户的地理位置信息。

3.案例：使用百度地图展示当前位置

■ 简介

- 本例通过HTML5 Geolocation API接口获取用户地理位置信息，并通过百度地图将其当前位置展示出来。

■ 描述

- 浏览器访问该页面，询问用户是否授权获取地理位置，用户同意后，页面将加载百度地图，展示用户当前地理位置。



现场演示：

- 案例21-02：使用百度地图展示当前位置

4.案例：使用百度地图展示运动轨迹

■ 简介

- 本例使用HTML5 Geolocation API监视用户地理位置信息，通过获取用户地理位置信息，从而得到其运动轨迹。

■ 描述

- 页面使用前面讲解过的jQuery Mobile构建，初次加载时展示其当前位置信息，点击“开始”，浏览器开始监视用户地理位置信息，并实时更新至百度地图中，绘制成运动轨迹，点击“结束”即停止监视，不再绘制运动轨迹。



现场演示:

- 案例21-03: 使用百度地图展示运动轨迹

Thanks.