近几年我国互联网技术的飞速发展，一直走在国际前沿，尤其是移动端互联的网技术发展速度更是惊人。人们越来越多的倾向于移动端互联网，几乎人手一部智能手机，人们用它浏览新闻，购物，玩游戏，看视频，聊天等等。因为它使用起来更加方便，更加快速，使用成本也很低。移动端互联网正在悄悄的改变着我们的生活。可以预见，在未来的中国，移动互联网必将取代传统的PC端互联网，成为互联网生态中重要的一环。

因此，近两年的移动端互联网技术成了关注的焦点，HTML5的出现更是在互联网里赚足了人们的眼球。以往很多公司在开发一款产品的时候，都会同时开发Android版本和ios版本，来兼顾更多的用户。而HTML5这一新的Web标准的出现，让许多企业有了新的选择。由于HTML5是基于浏览器Web开发标准，在各个平台都可以呈现开发的产品，大大的降低了产品开发和更新迭代的成本，它以其独特的优势，一跃成为很多公司和开发者的新宠。从2008年HTML5的首个版本发布，到今天HTML5的逐渐成熟，其魅力与优势已逐渐显现。HTML5的本地存储技术可以将应用常用的数据进行缓存，大大的提高了访问的速度和流畅性；它的语义化标签能够给每个结构赋予显而易见的意义，更有利于SEO优化；它的Canvas更是强大与灵活，可以结合Javascript代码在浏览器端绘制出华丽的图形，在游戏、图标制作、图形编辑器等方便都能发挥其巨大的潜力，使你眼花缭乱；它新增的多媒体标签，让flash成为过去；尤其是其新增的地理定位技术更加的便利了我们的生活。

本文旨在介绍并实际应用HTML5的地理定位技术。众所周知，互联网融入我们生活的每个部分，必不可分的就是希望获取用户的所在位置，以便更好的借助互联网来服务我们的生活。网上购物，聊天，O2O，地图导航等等，都离不开地理定位。所以人们更多的希望这种地理定位技术在移动端有更好的体现。2014年的10月，W3C正式发布了HTML5,其中的Geolocation API[1]便是其重要特性之一，它提供了获取用户位置的功能，并且可以持续定位，制定了定理位置定位相关的API[2]。基于它的这个功能，与百度地图API或者谷歌地图API相结合，便可以为用户开发出更加方便我们生活的Web应用，并且可以适用于各个平台。

**1 定理定位的多种方式**

获取地理位置信息的来源主要有：IP定位、GPS定位、WIFI定位、GSM定位。IP定位中，由于浏览器是将位置信息发送给ISP服务商来解析的，因此其IP地址与服务商所在地有关，但是可能与用户所在的位置不同，所以通过这种方式获取的地理位置误差较大；GPS定位由于是通过卫星定位，在露天的环境下获取位置比较精确，适用于移动设备；Wifi定位是通过多个Wifi接入点三角距离来获取位置信息，定位效果也同样精确，适用于室内环境的移动端；GSM定位是通过一个蜂窝基站接入GSM网络，通过GSM网络进行数据传输来获取位置信息，获取的位置信息也较为准确。

|  |  |  |
| --- | --- | --- |
| 数据源 | 优点 | 缺点 |
| IP 定位 | 任何地方都可用，在服务器端处理 | 不精确（经常出错，一般精确到城市级）  运算代价大 |
| GPS定位 | 非常精确 | 定位时间长，耗电量大，室内效果差  需要额外硬件设备支持 |
| WIFI 定位 | 精确 、可在室内使用 、简单、快捷 | 在乡村这些 Wi-Fi 接入点少的地区无法使用 |
| GSM定位 | 相当准确 、可在室内使用 、简单、快捷 | 需要能够访问手机或modem设备 |
| 用户自定义 | 可获得比程序定位服务更准确的位置数据，用户自行输入比自动检测要快 | 可能很不准确，特别是当用户位置改变后 |

##### 表1 位置信息获取方式对比

**2 HTML5地理定位的原理与机制**

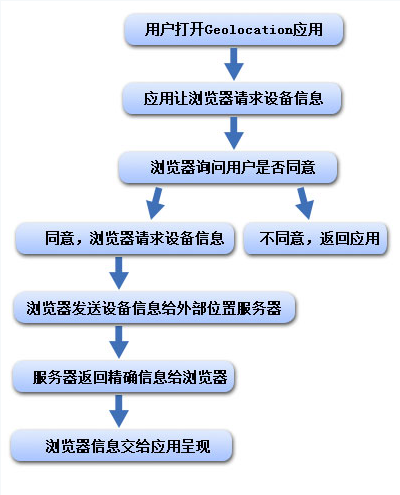
HTML5的地理定位技术的实际原理是Geolocation API调用移动终端的定位功能来获取位置信息从而实现定位的。用户在浏览器中打开Geolocation应用[3]，然后Geolocation应用向浏览器发出请求，希望获取设备的地理位置，浏览器询问用户是否允许共享地理位置。如果用户同意共享地理位置，那么浏览器通过移动端内置的功能来检测设备的地理位置信息，这些地理位置信息来源取决于设备是在什么样的联网情况下运行的，取自以上几种方式中的一种。接着浏览器就将检测到的设备的坐标信息发送到一个可信任的地理位置服务器上。假如是百度地图的位置服务器，那么百度地图的位置服务器就将具体的位置信息返回到浏览器，此时浏览器再将信息进行解析，用户便看到了自己的地理位置。如下图所示：

图1 获取位置信息机制

**3 浏览器的支持情况**

HTML5从W3C发布以来，各浏览器厂商对其支持表现不一。但幸运的是，在HTML5的所有新技术中，HTML5的Geolocation API是第一波全部被接受和实现的功能之一，并且关于Geolocation的相关规范已经非常成熟，不会再做大的改变。

|  |  |
| --- | --- |
| 浏览器 | 支持情况 |
| Firefox | 3.5及以上版本支持 |
| Chrome | 5.0版本及以上支持 |
| IE | 9.0版本及以上支持 |
| Opera | 版本10.60及以上支持 |
| Safari | 版本4及以上支持，在iPhone中可用 |

表2. 浏览器对HTML5 Geolocation 的支持情况

|  |  |
| --- | --- |
| 移动设备 |  |
| Android | 2.0版本及以上支持 |
| iPhone | 3.0版本及以上支持 |
| Symbian | S60 3rd或 5th |
| Blackberry OS | 6.0版本及以上支持 |

表3. 移动端浏览器对HTML5 Geolocation 的支持情况

**4 隐私保护**

众所周知，地理位置信息属于用户敏感的隐私信息之一，在未经过用户同意的情况下，用户是很不乐意透露自己的位置所在的。所以，用户在浏览器上使用Geolocation应用的时候，会向远端服务器发送请求，在这之前，浏览器会拦截这一请求，先询问用户是否同意共享自己的位置信息。在征得用户同意之后，才可以去发送这一请求。它的隐私保护机制尊重用户意见，很好的保护了用户隐私。



图2 浏览器询问用户是否同意共享位置信息

**5 Geolocation API 的使用方法**

**5.1 检查浏览器是否支持此方法**

在使用之前首先要检查一下该浏览器是否支持此方法，如果浏览器支持，则正常调用；如果不支持则提示浏览器不支持此方法。

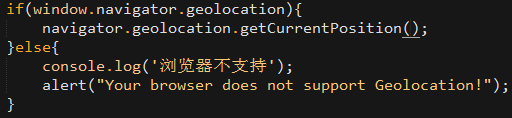


图3 检查浏览器是否支持Geolocation

**5.2 Geolocation 的 三个重要方法**

**5.2.1** getCurrentPosition()方法

getCurrentPosition()方法[4]也叫单次请求你方式，用来获取用户当前的地理位置信息，方法如下：

navigator. geolocation.getCurrentPosition(successCallback，errorCallback，options)；

getCurrentPosition()方法有三个参数:

1. successCallback: 必选参数；浏览器成功获取用户位置信息后的回调函数；
2. errorCallback: 可选参数；浏览器未获取用户位置信息后的回调函数；
3. options: 可选参数；获取用户信息的参数配置；

**5.2.1.1** successCallback()方法

successCallback()函数一样带有参数，分别是coords和timestamp。其中coords包含下表中的7个值：

|  |  |
| --- | --- |
| 属性 | 描述 |
| latitude | 设备所在的纬度 十进制度 |
| longitude | 设备所在的经度 十进制度 |
| altitude | 设备所在的海拔 WGS-84椭球 |
| accuracy | 纬度和经度的精确度 精确到米 |
| altitudeAccuracy | 海拔的精确度 精确到米 设备不支持时为null |
| heading | 设备的行进方向，相对于正北面而言 |
| speed | 当前地面设备的速度 以m/s为单位  设备不支持时为null |

表4. successCallback()的属性

其中的timestamp是获取到用户位置信息的时间戳。

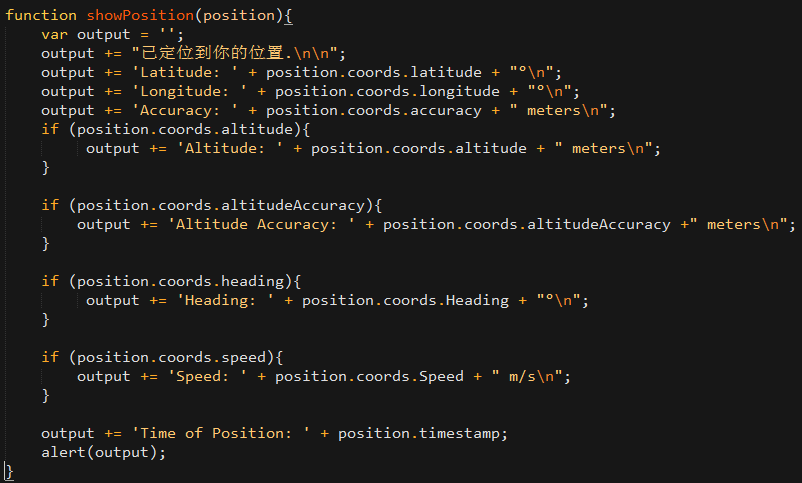


图4 showPosition函数代码

在代码中，我定义了一个showPosition函数，将整个函数作为successCallback函数，即成功获取到用户位置信息的回调函数。声明了一个output字符串变量，将coords中的七个属性和timestamp输出，来得到我的位置信息。如下图所示：

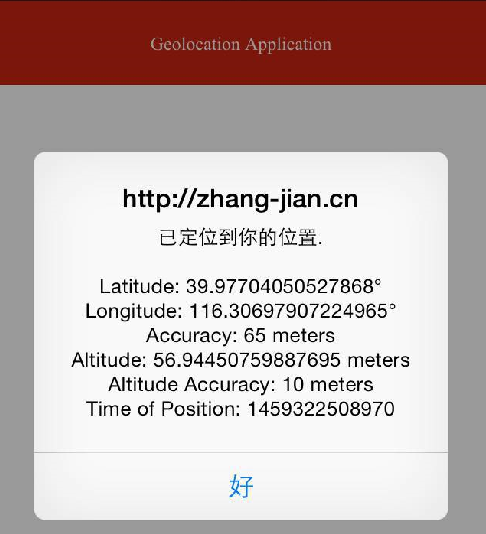


图5 获取到的位置信息

从获取到的位置信息可以看出，并没有获取到设备的heading和speed。说明本设备并不支持这两个属性。

**5.2.1.2** errorCallback ()方法

在这个API中，当然也有很多种原因导致请求位置信息失败，这时当然需要Geolocation API提供一个函数，给用户合理的解释，让用户知道是何原因没有成功获取到位置信息。此API中的errorCallback函数定义了所有需要处理的错误情况的错误编号，如下表所示：

|  |  |
| --- | --- |
| Error Code | 描述 |
| UNKNOW\_ERROR (0) | 未知错误，不包含以下几种错误编号中的错误，需要通过message查找详细错误信息 |
| PERMISSION\_DENIED (1) | 用户拒绝浏览器获取位置信息 |
| POSITION\_UNVAILABLE (2) | 尝试获取用户位置信息失败，如连不上GPS卫星或网络断线等情况 |
| TIMEOUT (3) | 获取位置信息的时间超出了option里设置的time,操作超时 |

表5. errorCallback()的error对象的code

错误的code存在于错误的对象中，错误对象作为error参数传递给错误处理函数。如下图函数：

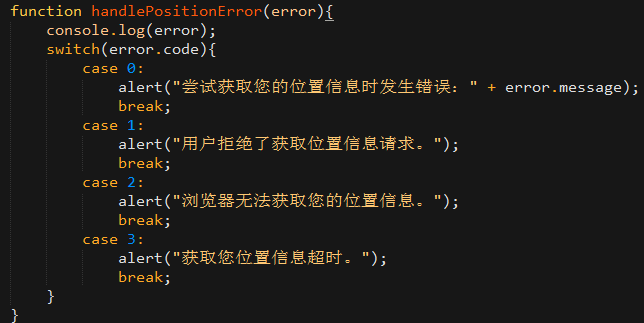


图4 handlePositionError()函数代码

**5.2.1.3** option参数

最后一个参数为option，是指配置项。此参数一般以Javascript 的JSON对象的“名称/值对”方式呈现，可选参数的配置属性有三个且都是可选项；如下表所示：

|  |  |  |
| --- | --- | --- |
| 属性 | 类型 | 描述 |
| enableHighAcuracy | Boolean | 是否启用高精确度模式 |
| timeout | Long | 对地理位置信息的获取操作设定一个超时限制(单位为ms) |
| maximumAge | Long | 对地理位置信息进行缓存的有效  时间(单位为ms) |

表6. option的属性

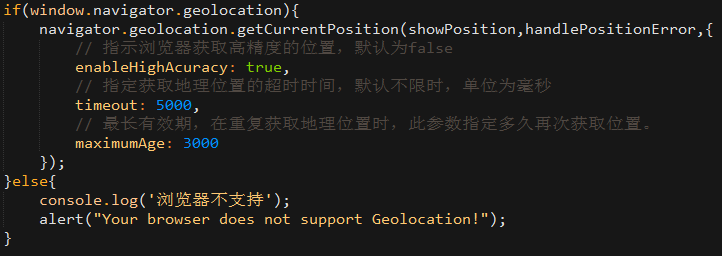


图5 option参数配置

**5.2.2** watchPosition ()方法

watchPosition()方法[5]也叫重复性位置请求，它可以连续获取用户的当前地理位置信息，并且会定期地自动获取用户的当前地理位置信息，用于监听用户的位置信息变化。方法定义如下：

navigator. geolocation.watchCurrentPosition(onSuccess，onError，option)；

watchPosition时刻监视用户的当前位置，当用户位置发生变化时，会触发onSuccess函数处理新的数据，否则会触发onError。另外，watchPosition的返回值类似于Javascript脚本中的延迟函数，可以停止对用户当前地理位置信息的监视。除此之外，它的其他参数配置和getCurrentPosition类似。

**5.2.3** clearWatch方法

clearWatch()方法用于停止对用户当前地理位置的监视。用法如下：

navigator. geolocation.clearWatch(watcher);

该方法的参数为调用watchCurrentPosition方法监视地理位置信息时的返回参数，功能是清除对用户位置的循环监视。

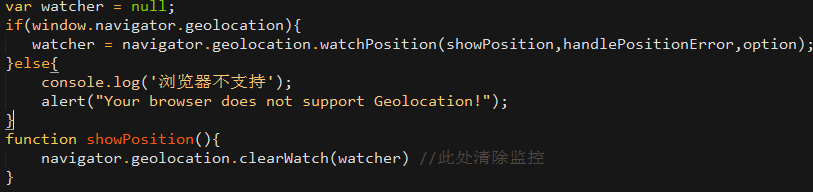


图6 clearWatch 使用方法

**6 结合百度地图API 构建实际应用**

本节主要介绍用上述的Geolocation API获取用户的位置，再结合百度地图，构建一个简单使用的Web应用。此Web应用具有定位，关键字检索和单个点沿线运动简单的功能。

**6.1** html代码：

<header id="city">

Geolocation Application

</header>

<section id="mymap"></section>

<footer>

<span id="bj"></span>

<span id="xx"></span>

<span id="meal"></span>

<span id="move"></span>

</footer>

**6.1.1**  HTML页面内容解析

页面中用到了HTML5新增的语义化标签，可以清楚的展示页面结构。header标签表示头部；section标签表示中间的地图展示，并给了id=mymap；footer标签表示页面的底部；其中header和footer运用了弹性盒布局，可以使其子元素的宽度均匀分配，都是相等的。页面内的section就是地图的容器;footer内的四个span标签是页面底部的四个按钮，分别代表不同的功能。

**6.2**  CSS部分代码：

body,section,header,footer,article,{

margin: 0;

padding: 0;

}

@media screen and (max-width:320px){

html{font-size:12px;}

}

@media screen and (min-width:321px) and (max-width:360px){

html{font-size:16px;}

}

@media screen and (min-width:361px) and (max-width:414px){

html{font-size:20px;}

}

@media screen and (min-width:415px) and (max-width:460px){

html{font-size:24px;}

}

@media screen and (min-width:461px){

html{font-size:26px;}

}

html{font-size:26px;}

}

**6.2.1**  CSS代码解析

代码的开始，初始化页面，去除页面内标签的默认margin和padding，防止在页面中出现不可控的外边距和内边距，影响页面呈现效果。接下来的@media screen使用了CSS3的新添特性——媒体查询，用于适配不同的屏幕尺寸。未的CSS代码是具体的页面布局，给了header和footer具体的高，中间部分的section自适应页面的长度；footer内部的四个span同样是用了CSS3新添的特性——弹性盒布局，达到四个按钮同比例分部的效果。本篇旨在介绍Geolocation API的功能及其应用，关于CSS3的特性在这里不再赘述，以免喧宾夺主。

**6.3**  Javascript代码：

6.3.1 getLocation()函数:

getLocation();

**var** meal = document.getElementById("meal");

**var** move = document.getElementById("move");

**var** beijing = document.getElementById("bj");

**var** xinxiang = document.getElementById("xx");

**var** city = document.getElementById('city');

**function** getLocation(){

**if**(window.navigator.geolocation){

navigator.geolocation.getCurrentPosition(

showPosition,handlePositionError,{

enableHighAcuracy: **true**,

timeout: 5000,

maximumAge: 3000

});

}**else**{

console.log('浏览器不支持');

alert("Your browser does not support Geolocation!");

}

}

**function** handlePositionError(error){

console.log(error);

**switch**(error.code){

**case** 0:

alert("尝试获取您的位置信息时发生错误：" + error.message);

**break**;

**case** 1:

alert("用户拒绝了获取位置信息请求。");

**break**;

**case** 2:

alert("浏览器无法获取您的位置信息。");

**break**;

**case** 3:

alert("获取您位置信息超时。");

**break**;

}

}

6.3.1.1 getLocation()函数代码解析

Javascript中的代码,其中函数getLocation()作为最终函数的调用，询问当前浏览器是否支持地理定位技术，在浏览器支持的情况下，才执行getCurrrentPosition()方法；如上文所讲，getCurrentPosition方法有三个参数：onSuccess, onError,option；前两个函数onSuccess和onError分别由函数showPositon()和handlerPosition()来执行，option即enableHighAcuracy(高精度),

timeout(超时时间)和maximumAge(最长有效期)。

6.3.2 handlePositionError ()函数:

**function** handlePositionError(error){

console.log(error);

**switch**(error.code){

**case** 0:

alert("尝试获取您的位置信息时发生错误：" + error.message);

**break**;

**case** 1:

alert("用户拒绝了获取位置信息请求。");

**break**;

**case** 2:

alert("浏览器无法获取您的位置信息。");

**break**;

**case** 3:

alert("获取您位置信息超时。");

**break**;

}

}

6.3.2.1 handlerPosition()函数代码解析

函数handlerPosition()即为onError()函数，也就是用户未能成功的获取用户位置信息，在这个函数里，会返回错误的代码编号，以便识别未能获取用户位置的具体原因。

6.3.3 showPosition ()函数:

**var** map = **new** BMap.Map("mymap");//创建地图实例

**var** point = **new** BMap.Point(position.coords.longitude, position.coords.latitude);//创建点坐标

map.centerAndZoom(point,15); //地图初始化

**var** marker = **new** BMap.Marker(point); // 创建标注

map.addOverlay(marker);

map.addControl(**new** BMap.GeolocationControl());//定位控件

map.addControl(**new** BMap.NavigationControl()); //地图平移缩放控件

map.addControl(**new** BMap.ScaleControl()); //比例尺控件

6.3.3.1 showPosition ()函数代码解析

函数showPosition()即为onSuccess()函数，也就是成功获取用户位置后执行的函数。上面代码中的position.coords.longitude是用户位置信息的经度，position- -.coords.latitude表示用户位置信息的纬度。经过百度地图的BMap.Point()函数，new出来百度地图中的坐标，即我们在百度地图中看到的坐标。

在这里要简单介绍一下百度地图API。百度地图API是供开发者开发实用性较强的地图应用时用到的接口，支持多种平台。它提供了很多实用的地图功能，如：基本地图展示功能、地图控件展示功能、定位功能、本地搜索功能、驾车检索、地图导航功能等。在百度地图Javascript API中，使用BMap作为命名空间，所有的类名都在此命名空间之下，如：BMap.Control、BMap.Overlay、BMap.Map等。

在页面加载上面的这个script前，需要先加载百度地图提供的api地址，这样才可以调用百度地图的接口。使用百度地图接口，需要申请相应的密钥，密钥申请后才可以拿着申请的密钥生成属于自己的调用百度地图接口的链接，这样做的原因也是为了更好的保护用户隐私。在页面的头部，我已经加入了百度地图提供的api地址，如下所示：



创建地图实例 var map = new BMap.Map(“mymap”),这里的mymap就是页面中的地图容器。Map是BMap下的一个类，表示地图。通过new字符创建一个地图的实例对象。

接着就是创建点坐标，我将当前的经纬度传递给BMap.Point，然后创建出当前的点坐标。使用map.centerAndZoom(point,15)将地图初始化，使用BMap.Marker在地图上创建标注。如下图所示：



图7 地图上的位置

接着是为地图添加需要的控件。我在本地图中添加了定位控件，地面平移缩放控件和比例尺控件。本应用还添加了关键字检索功能和城市定位功能，以及单个点沿线运动功能。这些功能都是基于百度地图进行实现的。

**6.2** 实际应用展示：

6.2.1 城市定位功能：



图8 定位到新乡 图9 定位到北京

给应用传城市的名字，自动定位到该城市。

6.2.2 周边搜索功能：



图10 搜索周边餐厅

利用获取的地理位置，搜索周边。此处，给的关键字是餐厅，搜索周边餐厅。

6.2.3 单个点沿线运动：



图11 单个点沿线运动

利用获取的位置当做起点，然后从起点到目的地自动规划路线，沿路线由当前位置去往终点。在这里，我将终点设为河南师范大学。

**7 Geolocation的不足**

尽管使用HTML5的Geolocation方法可以顺利的获取当前的位置信息，可是在地图上还是有一定的误差。图7展示的是Geolocation获取到的地理位置信息，可实际的位置并不是地图上所标注的。



图12 实际位置

实际位置如图中的蓝色原点所示，是经过百度地图的定位控件转化后的真实坐标。由此可见，通过Geolocation API获取到的经纬度，即便是开启了enableHi-

-ghAcuracy为true，也难免存在着较大的误差。这种误差有多方面的原因，因为在移动设备上，可以通过WIFI、IP地址、基站或者是GPS来获取位置信息，这几种方法中GPS是最为接近的。还有原因是因为Geolocation获取到的经纬度按标准的来说是google的经纬度，而并不是百度地图上所表示的经纬度。这两者之间的转化需要调用百度地图的convert.js将实际的经纬度转化为百度地图的经纬度，这样才能减小误差。

**8 结束语**

HTML5的出现对于目前飞速发展的互联网的发展来说具有划时代的意义，它带来的众多新技术将重新定义Web标准，影响力巨大。

本文介绍的HTML5的新技术之一Geolocation API就是一个很好的证明。它的出现降低了地理定位的开发难度，提供了浏览器用户精确的地理位置信息，催生了各种各样的基于互联网地理位置服务Web应用[6]。许多用这个新技术开发的移动导航、公交查询、周边搜索等相关应用都极大的方便了我们的生活。这种基于浏览器的地理定位和导航技术能避免多平台多操作系统的制约，能满足多种类地理位置信息的定位，不需要开发APP应用程序以及下载安装工作，将是基于浏览器实现地理定位和导航的有效手段。[7]

参考文献：

[1] Anthony.T.Holdener．HTML5地理位置定位.(HTML5.Geolocation)[M] the United States of America，2011

[2]W3C．Geolocation API Specification[EB/OL].http://dev.w3c/geo/spec-source.html，2012.05.10

[3]易晓飞．[HTML5 Geolocation API 研究与应用][J].计算机光盘软件与应用，2012.06

[4]王志刚．HTML5移动开发即学即用[M].北京：电子工业出版社，2012

[5]Bruce Lawson．HTML5用户指南[M].北京：机械工业出版社，2011

[6]梁莉菁．基于Geolocation API的HTML5地理位置追踪定位的实现[J].萍乡高等专科学校学报，2014.06

[7]鲁立，刘桢．基于HTML5地理定技术的移动终端导航设计[J].电子设计工程，2012.11