

API de geolocalización

Introducción

Una de las APIs relacionadas con HTML5 que el W3C se está encargando de especificar es la de geolocalización, cuyo objetivo es solicitar al agente de usuario (el navegador) información sobre su posición (latitud, longitud, dirección, velocidad, ...). Sin embargo, esta API no indica cómo puede obtener el navegador esa información.

Consecuentemente queda en manos de los fabricantes de dispositivos/navegadores establecer cuál va a ser ese sistema. Actualmente se aprovechan los siguientes parámetros:

- Sensores del dispositivo (GPS, GSM, brújula digital, ...).
- Si el cliente está conectado a Internet por WiFi puede rastrearse el SSID de su conexión entre las que ha recopilado Google durante la elaboración de las fotografías para su servicio Street View y proporcionar una ubicación aproximada.
- La dirección IP. Nuestro proveedor de acceso a Internet nos asigna una dirección IP cada vez que conectamos; esta dirección se escoge de entre la libres del *pool* (depósito) del proveedor. Estos pools están registrado en los enrutadores de Internet para que sepan como encaminar las peticiones procedentes de ellos, y entre la información que disponen sobre los pools se encuentra su ubicación aproximada.
- Otros tipos de señales de radio, como Bluetooth y RFID.

¿Para qué puede interesarnos conocer la posición del agente de usuario?

- Para ofrecerle servicios publicitarios próximos a su ubicación.
- Para evitar usurpaciones de identidad.
- Controlar el acceso a los contenidos en función del área geográfica.

Obviamente, la ubicación del agente de usuario es un dato privado y no podremos acceder a él sin la autorización expresa del usuario. Cuando desde JavaScript solicitemos la posición, el navegador mostrará un mensaje al usuario solicitándole su aprobación para compartir su ubicación.

El objeto *navigator.geolocation*

La API de geolocalización está disponible para JavaScript a través del objeto [navigator.geolocation](#). Este objeto dispone de los 3 métodos siguientes:

- **getCurrentPosition**(*gestionarExito*,*gestionarFallo*,*opciones*): Este método solicita la información de posición al navegador de forma asíncrona (tenga en cuenta que el dispositivo puede tardar cierto tiempo en comunicar esta información al navegador; por ejemplo, un GPS puede tardar minutos en ubicarse) y, en caso de obtenerla correctamente ejecuta la función especificada en el argumento *gestionarExito*, a la que envía como argumento un objeto con las siguientes propiedades.
 - **timestamp**: Devuelve el instante (referido a la época UNIX) en el que se obtuvo la información de ubicación. Esta parámetro puede utilizarse para determinar si al solicitar posteriormente información de ubicación es necesario adquirirla de nuevo o podemos aprovechar la última ubicación obtenida (tenga en cuenta que obtener una ubicación implica la puesta en servicio de distintos sensores cuyo consumo de corriente puede ser significativo en dispositivos alimentados a batería).
 - **coords**: Es a su vez un objeto con las siguientes propiedades:
 - **latitude**: Latitud en grados
 - **longitude**: Longitud en grados
 - **accuracy**: Precisión de la ubicación expresada como un radio en metros.
 - **altitude**: Altitud (no altura) en metros respecto al elipsoide de referencia WGS84.
 - **altitudeAccuracy**: Precisión de la altura en metros.
 - **heading**: Dirección de avance expresada en grados.
 - **speed**: Velocidad expresada en metros por segundo.

Por el contrario, si el navegador no es capaz de adquirir la información de ubicación, ejecutará la función indicada en el argumento *gestionarFallo* enviándole como argumento un objeto con las siguientes propiedades:

- **message**: Es el mensaje de texto que explica el error.
- **code**: Es un código numérico que puede adoptar los siguientes valores:
 - **0**: Error desconocido
 - **1**: El usuario no ha autorizado la obtención de la ubicación.
 - **2**: No se pudo determinar la ubicación.
 - **3**: Se ha superado el tiempo máximo para obtener la ubicación. Este tiempo máximo se establece a través del tercer argumento del método **getCurrentPosition()**.

El tercer argumento, *opciones*, es un objeto que podemos aportar para configurar la obtención de la ubicación. Sus propiedades son las siguientes:

- **enableHighAccuracy**: Si establecemos el valor true, obligaremos al dispositivo a recabar la información más precisa; esto puede suponer

poner en marcha sensores cuyo consumo es considerable. Por el contrario, con el valor `false` indicamos que también nos vale una ubicación aproximada.

- **timeout:** Es la caducidad, expresada en milisegundos, que queremos asignar a la petición de la ubicación. Si se supera este valor sin que el navegador haya recibido la información se ejecutará la función indicada en el argumento *gestionarFallo*.
- **maximumAge:** Indica al navegador que estamos dispuestos a aceptar una información de ubicación obtenida previamente si su antigüedad no es mayor al número de milisegundos indicado en esta propiedad.
- **watchPosition(gestionarExito,gestionarFallo,opciones):** Este método funciona exactamente igual que el anterior, con la única diferencia de que la ubicación se va actualizando de forma periódica automáticamente, llamando a la función *gestionarExito* cada vez que se produce un cambio. Este método devuelve un identificador que puede utilizarse con el método siguiente para interrumpir su actualización automática.
- **clearWatch(id):** Anula las peticiones asíncronas de posición que va generando automáticamente el método *watchPosition* identificado por *id*.

Por ejemplo:

```
001 <!DOCTYPE html>
002 <html>
003 <head>
004   <meta charset="utf-8" />
005   <title>HTML5</title>
006   <!--<link rel="stylesheet" href="css/main.css" type="text/css">-->
007   <style>
008   </style>
009   <!--<script src='jquery-1.7.2.min.js'></script>-->
010   <script>
011     function obtenerPosicion(){
012       var opciones = {
013         enableHighAccuracy: true,
014         timeout: 10000,
015         maximumAge: 30000};
016
017       navigator.geolocation.getCurrentPosition(gestionarExito,gestionarFracaso,opciones);
018     }
019     function gestionarExito(posicion){
020       var nombre;
021       document.getElementById('timestamp').innerHTML=posicion.timestamp;
022       document.getElementById('latitude').innerHTML=posicion.coords.latitude;
023       document.getElementById('longitude').innerHTML=posicion.coords.longitude;
024       document.getElementById('accuracy').innerHTML=posicion.coords.accuracy;
025       document.getElementById('altitude').innerHTML=posicion.coords.altitude;
026       document.getElementById('altitudeAccuracy').innerHTML=posicion.coords.altitudeAccuracy;
027       document.getElementById('heading').innerHTML=posicion.coords.heading;
028       document.getElementById('speed').innerHTML=posicion.coords.speed;
029     }
030     function gestionarFracaso(error){
031       alert('Se ha producido el siguiente error\n' + error.message);
032     }
033   </script>
034 </head>
035 <body>
036   <button onclick='obtenerPosicion();'>Obtener posición</button>
```

```

036     <table>
037     <tr>
038         <th>Propiedad</th><th>Valor</th>
039     </tr>
040     <tr><td>timestamp</td><td id="timestamp">--</td></tr>
041     <tr><td>coords.latttude</td><td id="latitude">--</td></tr>
042     <tr><td>coords.longitude</td><td id="longitude">--</td></tr>
043     <tr><td>coords.accuracy</td><td id="accuracy">--</td></tr>
044     <tr><td>coords.altitude</td><td id="altitude">--</td></tr>
045     <tr><td>coords.altitudeAccuracy</td><td id="altitudeAccuracy">--</td></tr>
046     <tr><td>coords.heading</td><td id="heading">--</td></tr>
047     <tr><td>coords.speed</td><td id="speed">--</td></tr>
048     </table>
049 </body>
050 </html>

```