

# Tesi - Geolocation API

Daniele Rigon - 857319

24 luglio 2018

## Indice

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Overview geolocation</b>               | <b>2</b> |
| <b>2</b> | <b>Specifiche</b>                         | <b>2</b> |
| 2.1      | Oggetto della geolocalizzazione . . . . . | 2        |
| 2.2      | Metodi . . . . .                          | 2        |
| 2.2.1    | GetCurrentPosition . . . . .              | 2        |
| 2.2.2    | WatchPosition . . . . .                   | 4        |
| 2.2.3    | ClearPosition . . . . .                   | 5        |
| 2.3      | Supporto compatibilità web . . . . .      | 5        |
| 2.4      | Problemi sicurezza/privacy . . . . .      | 5        |

# 1 Overview geolocation

La Geolocation API viene utilizzata per ottenere la posizione geografica di un utente. Poiché questo può compromettere la privacy la posizione non è disponibile a meno che l'utente non la approvi: su un dispositivo mobile avremo un set di coordinate provenienti dal sensore GPS mentre su un portatile potremo usare il posizionamento legato all'ip della connessione internet.

## 2 Specifiche

### 2.1 Oggetto della geolocalizzazione

Le API di geolocalizzazione sono pubblicate tramite l'oggetto navigator.geolocation. Se l'oggetto esiste, il servizio di geolocalizzazione è disponibile. Per testare l'esistenza di tale oggetto:

```
1  if ("geolocation" in navigator) {  
2      /* la geolocalizzazione è disponibile */  
3  } else {  
4      /* la geolocalizzazione NON è disponibile */  
5  }
```

### 2.2 Metodi

Ci sono solamente tre metodi a disposizione: getCurrentPosition, watchPosition e clearWatch. I primi due sono utili a ottenere la posizione corrente mentre il terzo serve ad annullare la ricerca della posizione corrente. La differenza tra i primi due va ricercata nella loro periodicità, mentre il primo metodo fornisce il dato una sola volta, il secondo si attiva automaticamente ogniqualvolta la posizione cambi, o ogni tot intervallo di tempo. La sintassi per invocare questi metodi è la seguente:

```
1  navigator.geolocation.getCurrentPosition(inCasoDiSuccesso, opzInCasoDiErrore, opzioni);  
2  navigator.geolocation.watchPosition(inCasoDiSuccesso, opzInCasoDiErrore, opzioni);
```

#### 2.2.1 GetCurrentPosition

```
1  navigator.geolocation.getCurrentPosition(function(position) {  
2      do_something(position.coords.latitude, position.coords.longitude);  
3  });
```

L'esempio chiama la funzione dosomething() quando la posizione viene calcolata. Un esempio concreto potrebbe essere il seguente:

```

1  /* Nel frammento di codice appena illustrato possiamo vedere tutte le informazioni
   *   estraibili dalla struttura Position. Chiaramente, a seconda del device sul quale
   *   viene effettuata l'interrogazione, non tutte saranno sempre disponibili, in tal caso
   *   il loro valore sarà impostato a null.*/
2  function success(position) {
3      document.getElementById('latitude').innerHTML = position.coords.latitude;
4      document.getElementById('longitude').innerHTML = position.coords.longitude;
5      document.getElementById('position-accuracy').innerHTML = position.coords.accuracy;
6
7      document.getElementById('altitude').innerHTML = position.coords.altitude ? position.
8          coords.altitude :
9          'unavailable';
10     document.getElementById('altitude-accuracy').innerHTML = position.coords.
11         altitudeAccuracy ? position.coords.altitudeAccuracy :
12         'unavailable';
13     document.getElementById('heading').innerHTML = position.coords.heading ? position.
14         coords.heading :
15         'unavailable';
16     document.getElementById('speed').innerHTML = position.coords.speed ? position.coords.
17         speed :
18         'unavailable';
19 }

```

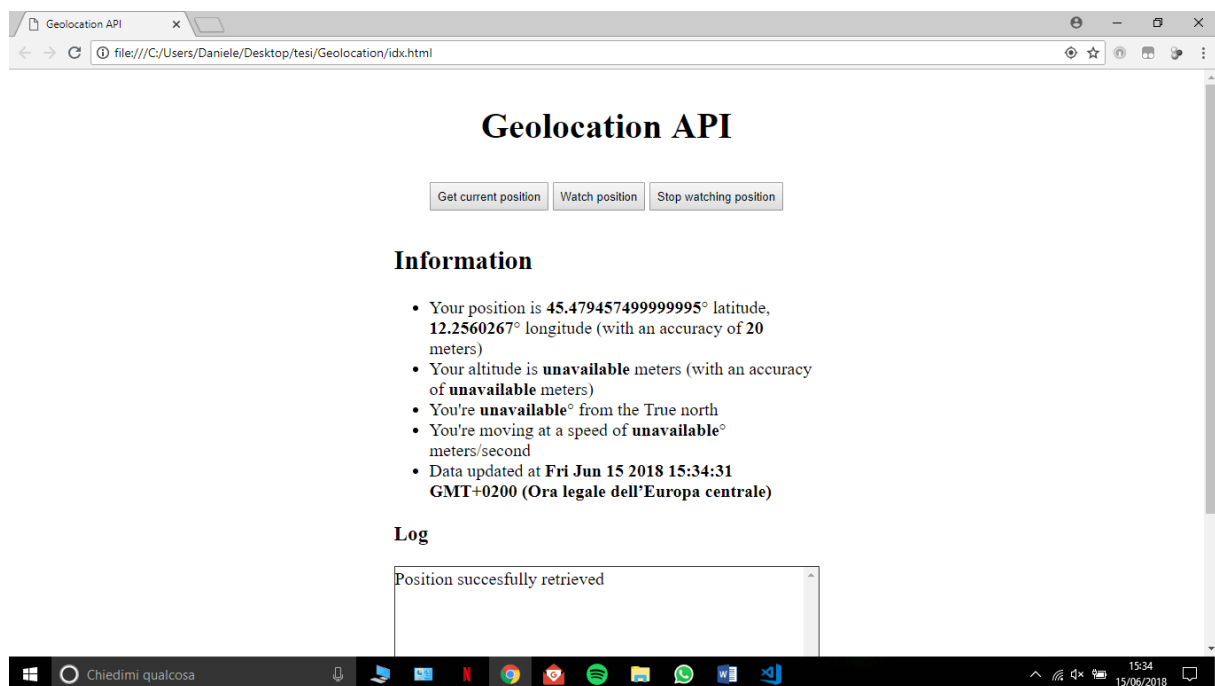


Figura 1: Prova info Geolocation.

In caso invece si verifichi un errore la funzione preposta deve accettare anch'essa un parametro, un oggetto di tipo `PositionError` contenente un codice di errore ed un messaggio ad uso di debug, ad esempio:

```

1  message.opzInCasoDiErrore = function(error){
2      alert( "Errore " + error.code + ": " + error.message );
3  }

```

### 2.2.2 WatchPosition

Se la posizione cambia (perché il dispositivo si sposta o perché viene calcolata una posizione più accurata), si può settare una funzione che viene chiamata quando la posizione attuale si aggiorna. Basta usare la funzione `watchPosition()`, che ha gli stessi parametri di input di `getCurrentPosition()`. Questa funzione viene chiamata più volte così da permettere al browser di sapere sempre la posizione del dispositivo. La funzione di errore è opzionale come lo era per `getCurrentPosition()`.

```
1 var watchID = navigator.geolocation.watchPosition(function(position) {
2     do_something(position.coords.latitude, position.coords.longitude);
3 });
```

Il metodo `watchPosition()` ritorna un ID numerico che può essere usato per identificare univocamente il controllo della posizione;

Nota: si può usare questo valore insieme al metodo `clearWatch()` per fermare il controllo della posizione.

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

Infatti, un'invocazione del metodo `watchPosition` può essere successivamente interrotta utilizzando la funzione `clearWatch`, come nell'esempio seguente:

```
1 <script>
2 var id_watch = null;
3 /*some code*/
4 watchId = navigator.geolocation.watchPosition(success, error, positionOptions);
5 // ...
6 </script>
```

### 2.2.3 ClearPosition

Viene usato il metodo `clearWatch()` per annullare il monitoraggio della posizione.

```
1 navigator.geolocation.clearWatch(watchId);
2 /*Il numero ID e quello restituito dal metodo watchPosition() descritto precedentemente*/
```

## 2.3 Supporto compatibilità web

### 2.4 Problemi sicurezza/privacy

<https://developers.google.com/web/updates/2016/04/geolocation-on-secure-contexts-only>

Può essere eseguito solo in HTTPS o localhost. Se eseguito in HTTP, o si è in una pagina HTTPS ospitata da HTTP (vedi secure context), viene gestito l'errore.

(vedere se levare questa parte)

```
1     navigator.geolocation.getCurrentPosition(function(success) {
2         /* Do something */ },
3         function(failure) {
4             if(failure.message.indexOf("Only secure origins are allowed") == 0) {
5                 // Secure Origin issue.
6             }
7         });
8     });
```

Geolocation API | Geolocation - Web APIs |

Sicuro | https://developer.mozilla.org/en-US/docs/Web/API/Geolocation

## Browser compatibility

**!** We're converting our compatibility data into a machine-readable JSON format. This compatibility table still uses the old format, because we haven't yet converted the data it contains. [Find out how you can help!](#)

|                     | Desktop | Mobile |                 |                   |  |        |
|---------------------|---------|--------|-----------------|-------------------|--|--------|
| Feature             | Chrome  | Edge   | Firefox (Gecko) | Internet Explorer | Opera  | Safari |
| Basic support       | 5       | (Yes)  | 3.5 (1.9.1)     | 9                 | 10.60<br>Removed in 15.0<br>Reintroduced in 16.0 | 5      |
| Secure context-only | 47      | ?      | 55 (55)         | No support        | (Yes)  | (Yes)  |

## See also

Chiedimi qualcosa

15:36 15/06/2018

Figura 2: Browser compatibility