

# Tesi - Geolocation API

Daniele Rigon - 857319

2 agosto 2018

## Indice

<b>1</b>	<b>Overview</b>	<b>2</b>
<b>2</b>	<b>Specifiche</b>	<b>3</b>
2.1	Oggetto della geolocalizzazione . . . . .	3
2.2	Metodi . . . . .	3
2.2.1	GetCurrentPosition . . . . .	3
2.2.2	WatchPosition . . . . .	7
2.2.3	ClearPosition . . . . .	7
<b>3</b>	<b>Problemi sicurezza/privacy</b>	<b>8</b>
<b>4</b>	<b>Supporto compatibilità web</b>	<b>10</b>
<b>5</b>	<b>Conclusioni</b>	<b>11</b>

# 1 Overview

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 }  
4 else {  
5   /* la geolocalizzazione non è disponibile */  
6 }
```

### 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 ogni qualvolta la posizione cambi, o ogni tot intervallo di tempo. La sintassi per invocare questi metodi è la seguente:

```
1 navigator.geolocation.getCurrentPosition(success, error, options);  
2 navigator.geolocation.watchPosition(success, error, options);
```

#### 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<sup>1</sup>:

---

<sup>1</sup>viene chiesto il permesso all'utente nell'uso della posizione quando si chiama il metodo GetCurrentPosition e WatchPosition; se negata apparirà un messaggio di errore in console, se consentita verranno mostrati i dati dell'utente

```

1  /*Il codice mostra tutte le informazioni estraibili da Position; a
   seconda del device sul quale viene effettuata l'interrogazione non
   saranno tutte 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      document.getElementById('altitude').innerHTML = position.coords.
        altitude ? position.coords.altitude : 'unavailable';
7      document.getElementById('altitude-accuracy').innerHTML = position.
        coords.altitudeAccuracy ? position.coords.altitudeAccuracy : '
        unavailable';
8      document.getElementById('heading').innerHTML = position.coords.heading
        ? position.coords.heading : 'unavailable';
9      document.getElementById('speed').innerHTML = position.coords.speed ?
        position.coords.speed : 'unavailable';
10 }

```

Che produrrà la seguente pagina:

## Geolocation API

Get current position
Watch position
Stop watching position

### Information

- Your position is **unavailable**° latitude, **unavailable**° longitude (with an accuracy of **unavailable** meters)
- Your altitude is **unavailable** meters (with an accuracy of **unavailable** meters)
- You're **unavailable**° from the True north
- You're moving at a speed of **unavailable**° meters/second
- Data updated at **unavailable**

### Log

Clear log

Figura 1: Pagina info Geolocation

Quando saranno chiamati i metodi `getCurrentposition` o `WatchPosition` verrà chiesto all'utente il permesso per usare la posizione.

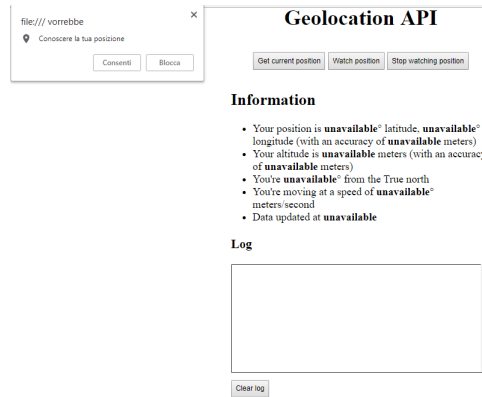


Figura 2: Richiesta permesso

Se l'utente rifiuta la posizione non viene calcolata e viene mostrato un messaggio di errore in console.

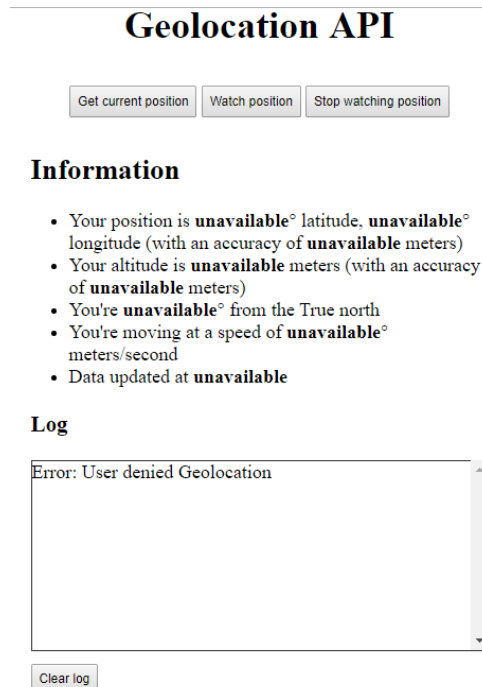


Figura 3: Rifiuto autorizzazione

Se l'utente acconsente all'utilizzo della posizione verrà mostrato un messaggio di conferma in console e saranno mostrate a video tutte le informazioni disponibili dell'utente.

---

## Geolocation API

### Information

- Your position is **45.4857255°** latitude, **12.2440479°** longitude (with an accuracy of **872** meters)
- Your altitude is **unavailable** meters (with an accuracy of **unavailable** meters)
- You're **unavailable°** from the True north
- You're moving at a speed of **unavailable°** meters/second
- Data updated at **Wed Aug 01 2018 10:23:20 GMT+0200 (Ora legale dell'Europa centrale)**

### Log

Position succesfully retrieved

---

Figura 4: Accetto autorizzazione

### 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.

### 2.2.3 ClearPosition

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

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

### 3 Problemi sicurezza/privacy

Uno dei principali problemi con la Geolocation API è rappresentato dagli attacchi di cross-site scripting (XSS) dovuto al fatto che gli oggetti per tracciare le coordinate (latitudine e longitudine) risiedono all'interno del DOM, il quale è accessibile con JavaScript e attraverso il quale potrebbe essere rubata la posizione dell'utente. Dato che gli utenti si fidano del sito web, si fidano anche della richiesta di posizione e la condividono. Un problem importante è che se l'utente non disabilita il tracciamento il browser continuerà a esporre la posizione dell'utente all'attaccante.

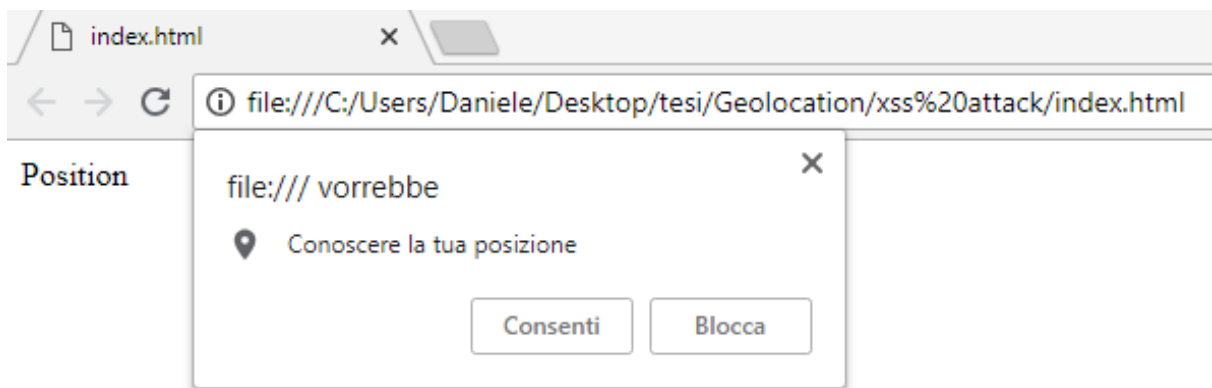


Figura 5: Richiesta posizione

Supponiamo che un utente malintenzionato abbia rilevato una vulnerabilità XSS in un sito Web; tutto ciò che dovrà fare è fare in modo che la vittima esegua il seguente codice JavaScript per rubare la posizione.

```
1 function showPosition(position){
2     var pos="Latitude: " + position.coords.latitude + "<br>Longitude: "
3     + position.coords.longitude;
4     document.getElementById("mydiv").innerHTML = pos;
5 }
6 function getLocation(){
7     navigator.geolocation.getCurrentPosition(showPosition);
8 }
9 getLocation();
```

Il codice utilizza le proprietà del DOM `coords.latitude` e `coords.longitude` per determinare rispettivamente la latitudine / longitudine e le memorizza in una variabile. Successivamente il codice JavaScript invia i dati al dominio dell'attaccante, in modo diverso a seconda di come è stato configurata la richiesta.



Se l'utente non consente all'utilizzo della posizione non succederà nulla nella pagina.

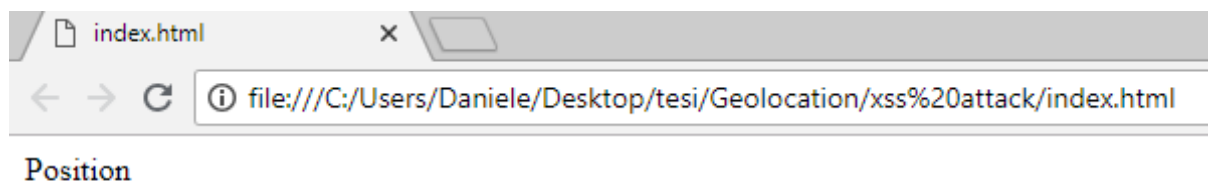


Figura 6: Blocca autorizzazione

Se l'utente acconsente all'utilizzo della posizione essa sarà calcolata e, risiedendo nel DOM, può esser facilmente rubata.

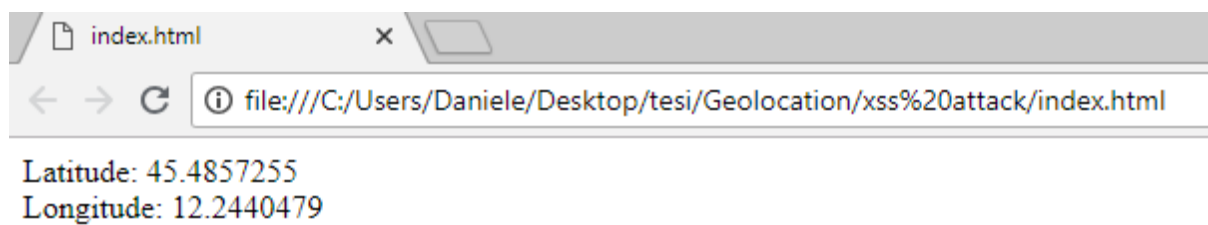

















Figura 7: Consenti autorizzazione

## 4 Supporto compatibilità web

														
														
Basic support	5	12	3.5	9	16	5	Yes	Yes	12	4	15	Yes	Yes	
Secure context required	50	?	55	No	37	Yes	51 *	50	?	55	37	Yes	?	
<code>clearWatch</code>	5	Yes	3.5	9	16	Yes	Yes	Yes	Yes	4	15	Yes	Yes	
<code>getCurrentPosition</code>	5	Yes	3.5	9	16	Yes	Yes	Yes	Yes	4	15	Yes	Yes	
<code>watchPosition</code>	5	Yes	3.5	9	16	Yes	Yes	?	Yes	4	15	Yes	Yes	

..

Full support



No support

..

Compatibility unknown

\*

See implementation notes.

Figura 8: Desktop and mobile compatibility

## 5 Conclusioni

## Elenco delle figure

1	Pagina info Geolocation . . . . .	4
2	Richiesta permesso . . . . .	5
3	Rifiuto autorizzazione . . . . .	5
4	Accetto autorizzazione . . . . .	6
5	Richiesta posizione . . . . .	8
6	Blocca autorizzazione . . . . .	9
7	Consenti autorizzazione . . . . .	9
8	Desktop and mobile compatibility . . . . .	10

## Riferimenti bibliografici

- [1] Connorshea, Chris David Mills, HeilKing, northvanhooser, erikadoyle, fscholz, Alhadis, teoli, FabioMagnoni (2018), *Features restricted to secure contexts*, *Mozilla Developer*, <https://developer.mozilla.org/en-US/docs/Web/API/Geolocation>
- [2] heppy, Nisarg-Shah, tacsipacsi, chrisdavidmills, andysh, aneditor, smalllong, edent, divyan-shu013, erikadoyle, VAggrippino, bsvensson, ewape, jpmedley, bjohnson, jsx, mugsydylan, Sebastianz, bizzzybetz, shaneriley, atrama, nikifor, openjck, teoli, rebloor, Jeremie, jswisher, GARAAD, Zupper, jyz19880823, krishnachandra, markg, kohei.yoshino, kscarfone, SarahWalrus, BrandonLove, kmaglione, wlach, trevorh, ronj, ethertank, lmorchard, DavidWalsh, JohnKarahalis, mikerhodes, dflanagan, fcheslack, eberon, inma610, paul.irish, sebmozilla, dynamis, Steffen, stevep98, Dougt, Soupdragon, Chtitux, Bzbarsky *Geolocation API*, *Mozilla Developer*, [https://developer.mozilla.org/en-US/docs/Web/API/Geolocation\\_API](https://developer.mozilla.org/en-US/docs/Web/API/Geolocation_API)
- [3] W3C (2018), *Geolocation API Specification 2nd Edition*, <https://www.w3.org/TR/geolocation-API/>
- [4] Ioannis Krontiris, Andreas Albers and Kai Rannenberg, *W3C Geolocation API calls for Better User Privacy Protection*, Chair of Mobile Business and Multilateral Security, Goethe University, Frankfurt, Germany
- [5] Doty, Nick, Mulligan, Deirdre K., Wilde, Erik (2010), *Privacy Issues of the W3C Geolocation API*, UC Berkeley School of Information, <https://escholarship.org/uc/item/0rp834wf>
- [6] Ruadhán O'Donoghue (2013), *HTML5 for the Mobile Web – a guide to the Geolocation API*, <https://mobiforge.com/design-development/html5-mobile-web-a-guide-geolocation-api>
- [7] OccupyTheWeb, WonderHowTo (2015), *How to Find the Exact Location of Any IP Address*, <https://null-byte.wonderhowto.com/how-to/hack-like-pro-find-exact-location-any-ip-address-0161964/>
- [8] Kipkay(2017), *Trace Any IP Address*, <https://internet.gadgethacks.com/how-to/trace-any-ip-address-1916/>
- [9] Aurelio De Rosa (2014), *An Introduction to the Geolocation API*, <https://code.tutsplus.com/tutorials/an-introduction-to-the-geolocation-api--cms-20071>
- [10] Rafay Baloch, HTML5, *HTML5 Modern Day Attack And Defense Vector*, <http://www.xss-payloads.com/papers/HTML5AttackVectors.pdf>