

Tesi - Geolocation API

Daniele Rigon - 857319

1 agosto 2018

Indice

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

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  } 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 ogni qualvolta 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.

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  }
```

Un esempio concreto potrebbe essere il seguente. Nota: 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  /* 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      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 }
```

Questo codice produrrà la seguente pagina.

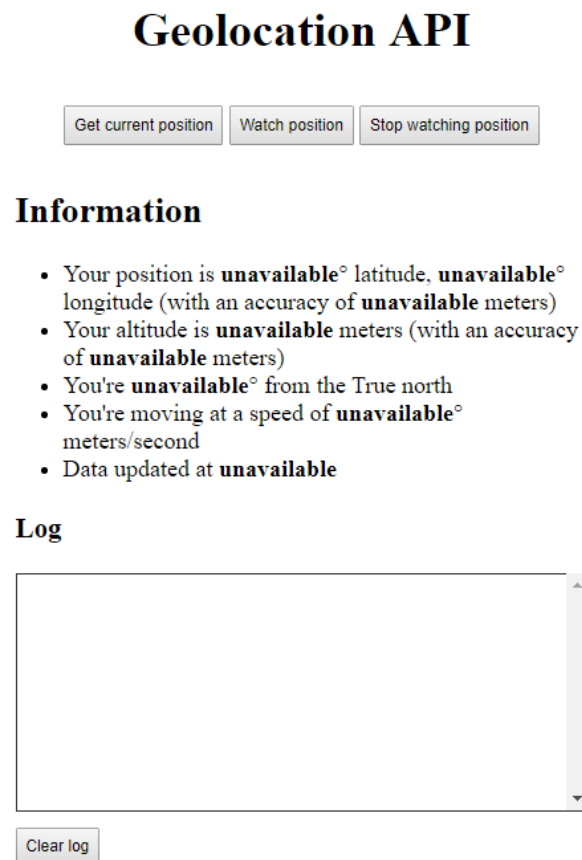


Figura 1: Pagina info Geolocation

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

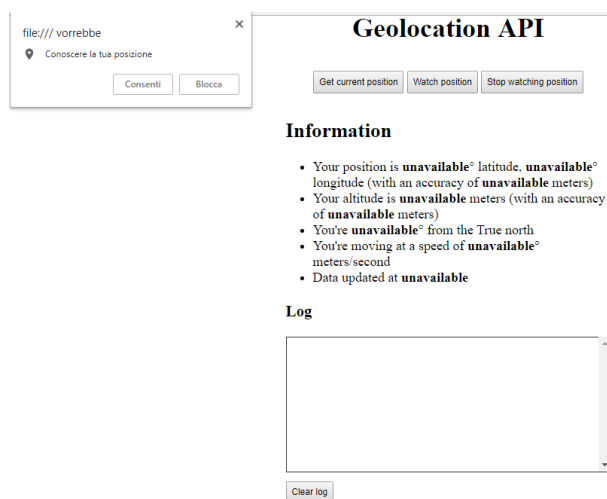


Figura 2: Richiesta permesso

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

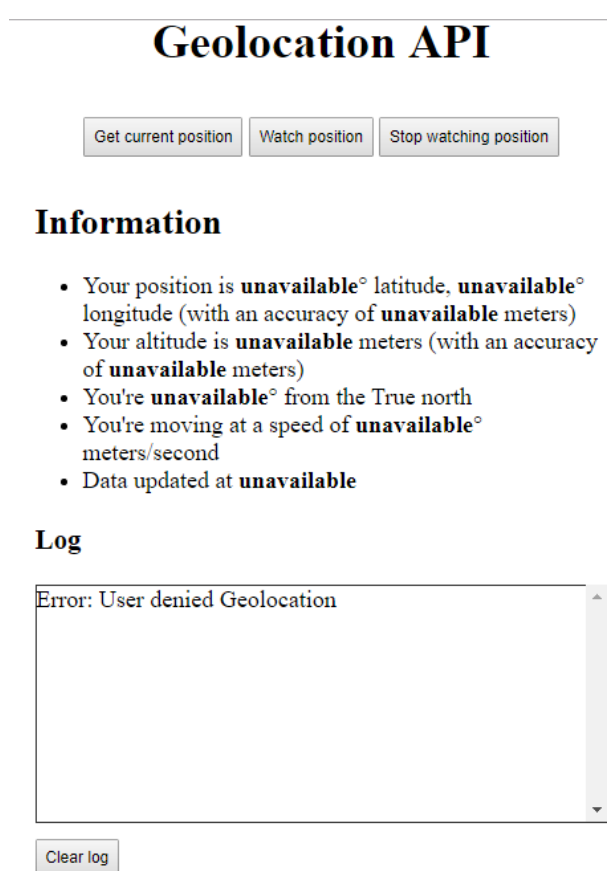


Figura 3: Rifiuta

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: Accetta

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);
```

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*/
```

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 problema importante è che se l'utente non disabilita il tracciamento il browser continuerà a esporre la posizione dell'utente all'attaccante.

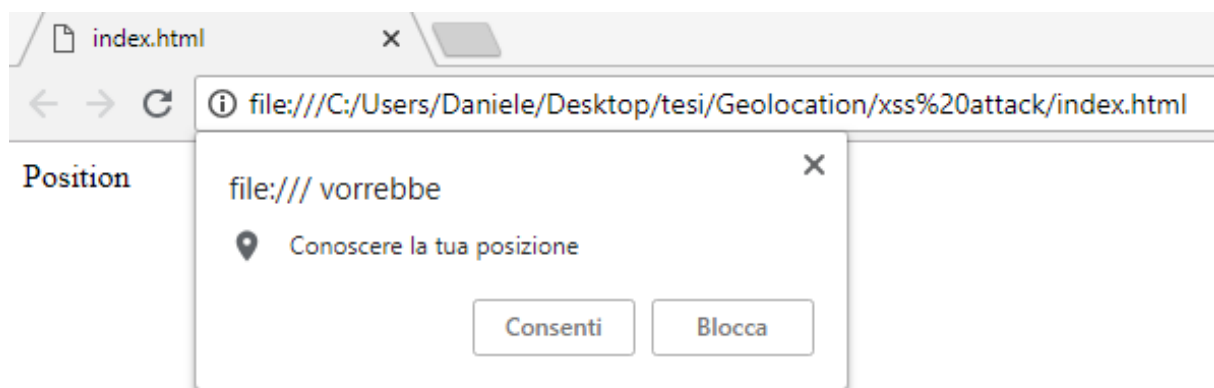


Figura 5: Richiesta posizione

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

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.

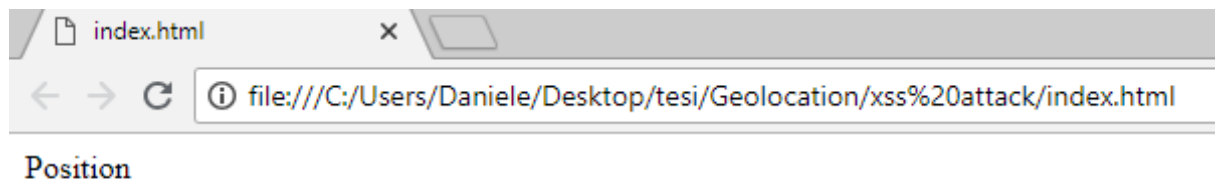


Figura 6: Blocca

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

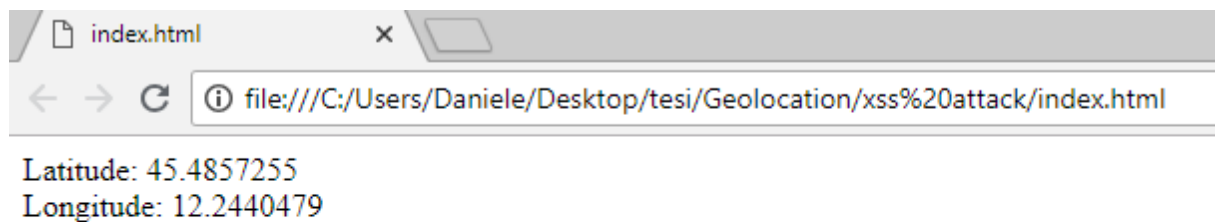




















Figura 7: Consenti

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