Per quanto riguarda le Geolocation Api, 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.

Le API per la geolocalizzazione permettono agli utenti di fornire la propria posizione alle applicazioni web. Per proteggere la privacy all'utente viene richiesta l'autorizzazione all'uso della posizione.

**L'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:

if ("geolocation" in navigator) {

/\* la geolocalizzazione è disponibile \*/

} else {

/\* la geolocalizzazione NON È disponibile \*/

}

**LE SPECIFICHE**

Esistono praticamente solo due metodi a disposizione, getCurrentPosition e watchPosition (+ clearWatch), entrambi utili a ottenere la posizione corrente.

La differenza tra i 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:

navigator.geolocation.getCurrentPosition(inCasoDiSuccesso, opzInCasoDiErrore, opzioni);

navigator.geolocation.watchPosition(inCasoDiSuccesso, opzInCasoDiErrore, opzioni);

**GETCURRENTPOSITION**

navigator.geolocation.getCurrentPosition(function(position) {

do\_something(position.coords.latitude, position.coords.longitude);

});

L'esempio qui sopra chiama la funzione do\_something() quando la posizione viene calcolata.

!=

La funzione invocata in caso di successo con tutte le info:

inCasoDiSuccesso = function(position){

alert( "Posizione delle: " + position.timestamp.getHours() + ":" +

position.timestamp.getMinutes() + "n" +

"Accuratezza delle coordinate: " + position.coords.accuracy + " mt; n" +

"Latitudine: " + position.coords.latitude + " gradi; n" +

"Longitudine: " + position.coords.longitude + "gradi; n" +

"Accuratezza dell'altezza: " + position.coords.altitudeAccuracy + " mt; n" +

"Altezza: " + position.coords.altitude + " mt; n" +

"Direzione: " + position.coords.heading + " gradin " +

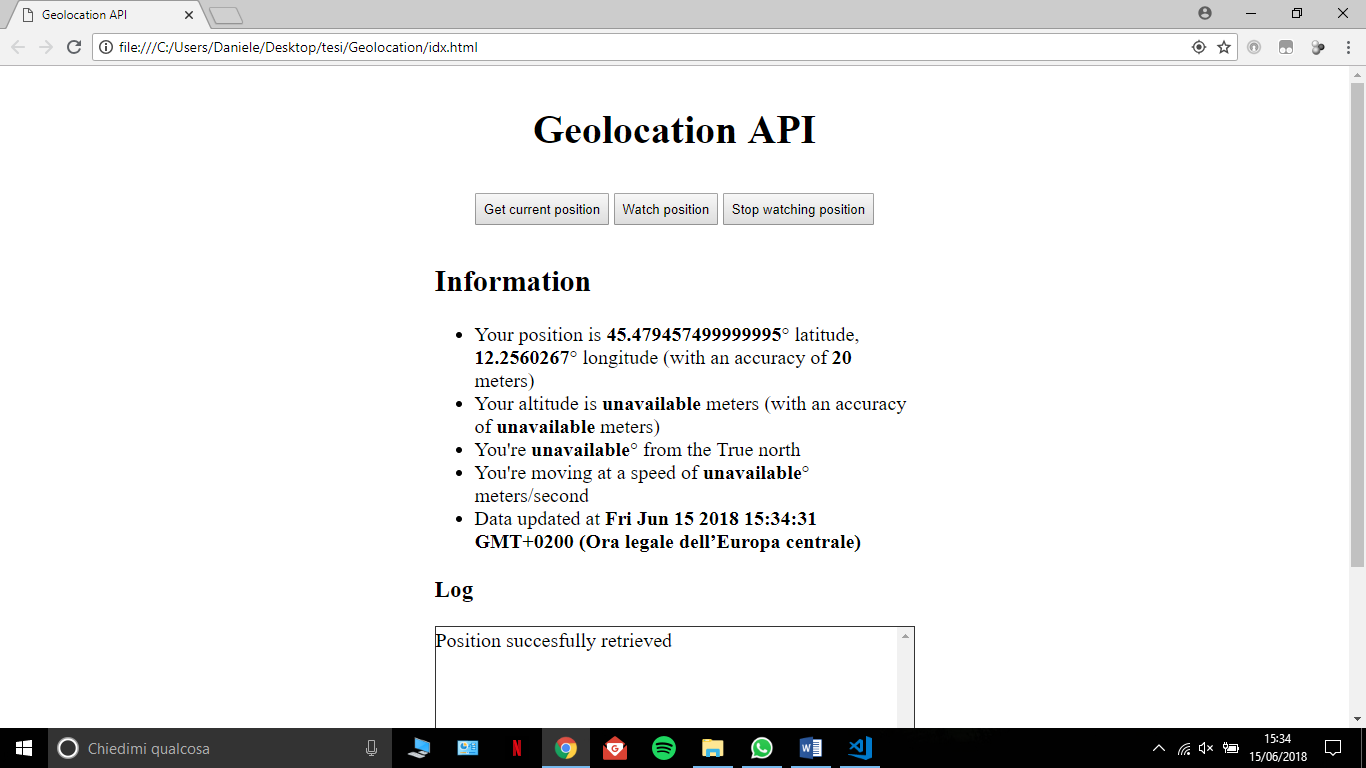
"(0 = Nord, 90 = Ovest, 180 = Sud, 270 = Est);n" +

"Velocita: " + position.coords.speed + " m/s;"

);

}

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.



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:

message.opzInCasoDiErrore = function(error){

alert( "Errore " + error.code + ": " + error.message);

}

**WATCH POSITION**

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().

var watchID = navigator.geolocation.watchPosition(function(position) {

do\_something(position.coords.latitude, position.coords.longitude);

});

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.

navigator.geolocation.clearWatch(watchID);

Infatti, un'invocazione del metodo watchPosition può essere successivamente interrotta utilizzando la funzione clearWatch,

come nell’esempio seguente:

<!doctype html>

<html>

<head>

<title> Un esempio di watchPosition</title>

<script>

var id\_watch = null;

inCasoDiSuccesso = function(position){

document.getElementById("posizione\_corrente").insertAdjacentHTML('beforeend',

"<li> Lat: " + position.coords.latitude + ", Lon: " + position.coords.longitude + );

"</li>"

);

}

sospendiLaRicezione = function(){

navigator.geolocation.clearWatch(id\_watch);

}

window.onload = function(){

id\_watch = navigator.geolocation.watchPosition(inCasoDiSuccesso);

}

</script>

</head>

<body>

<h1>La tua posizione attuale</h1>

<menu type="toolbar">

<button type="button" onclick="sospendiLaRicezione()">

sospendi la ricezione di dati geospaziali

</button>

</menu>

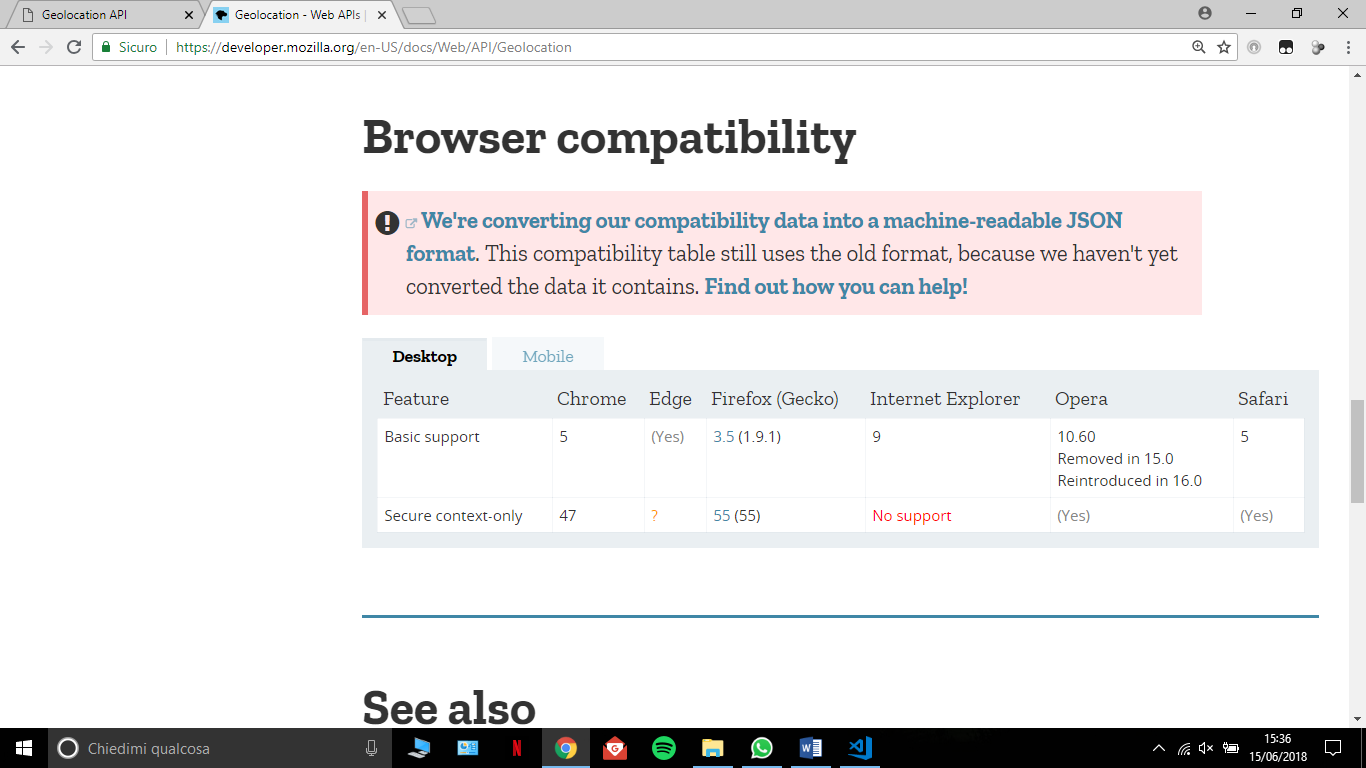
<ul id="posizione\_corrente">

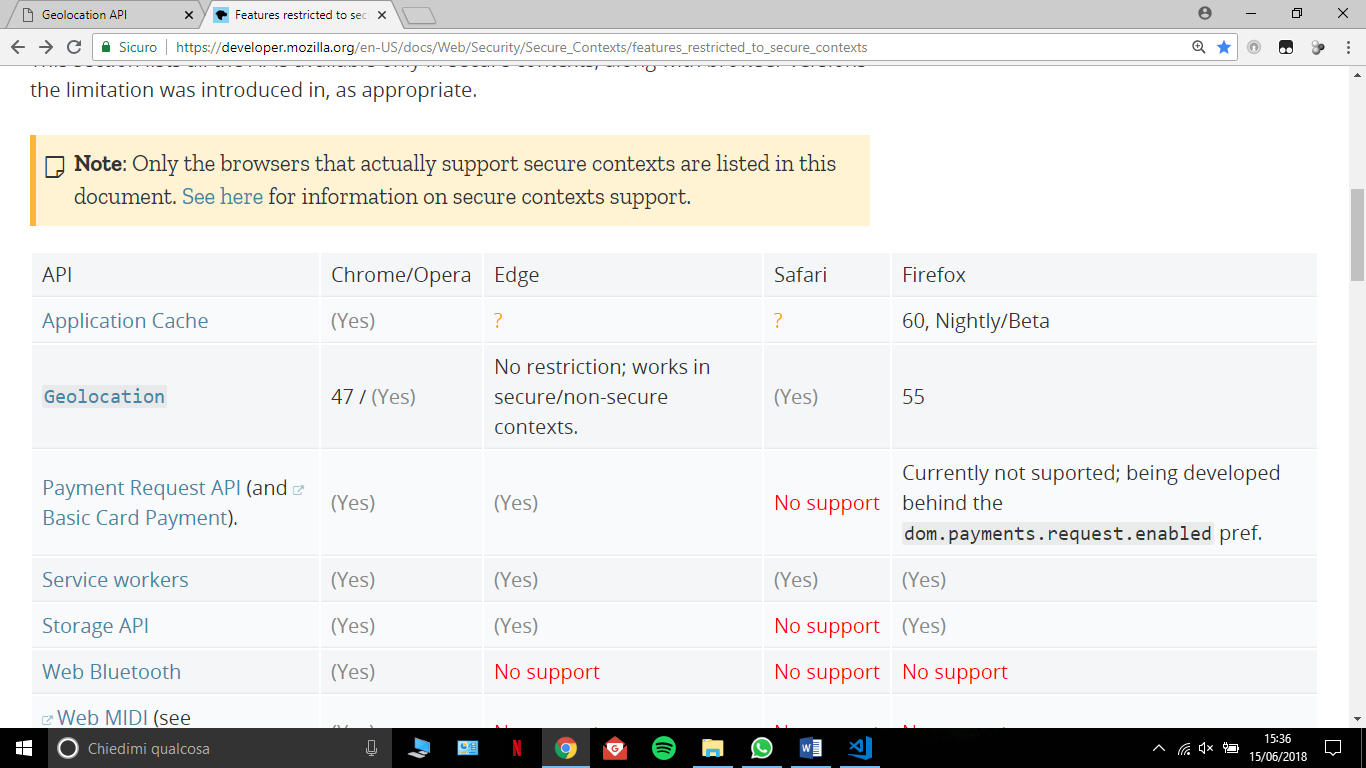
</ul>

</body>

</html>

**SUPPORTO COMPATIBILITÀ BROWSER**





**PROBLEMI SICUREZZA/PRIVACY GEOLOCATION API**

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

**COME TROVA LA POSIZIONE ESATTA**

<https://stackoverflow.com/questions/4213410/how-does-html5-geolocation-work>

**TEST RESULT**

<https://wpt.fyi/results/geolocation-API>

**TEST REPOSITORY**

<https://github.com/web-platform-tests/wpt/tree/master/geolocation-API>