Tracking a position in real time

INTRODUCTION

In order to track the current position, the geolocation API provides a method similar to the`getCurrentPosition(onSuccess, onError)` named`watchPosition(onSuccess, onError)`.

When `getCurrentPosition` gives a position when called,`watchPosition` does the following:

- **It gets the callback function only when the current position changes**. If you stay in the same location, the callback function won't be called regularly.
- It returns an `id` so that you can use the `clearWatch(id)` method to stop the current tracking.

TYPICAL USE

```
// get an id of the current tracking, the showPosition
callback is like the one we saw in earlier examples.
var watchPosId =navigator.geolocation.watchPosition(showPosition);
...
// stop the tracking
navigator.geolocation.clearWatch(watchPosId);
```

As an exercise, you may just try to change`getCurrentPosition` to `watchPosition` in the previous examples, and try this code using a mobile phone or tablet, walk for 20 meters and see the position changing.

External resource

- An article on html5rocks.com that shows how to write a simple trip meter using the geolocation API: in particular, you will find a JavaScript function that computes the

distance (in meters) between two positions defined by their longitude and latitude.

OPTIONS AVAILABLE WHEN USING THE GEOLOCATION API, IN PARTICULAR REAL TIME TRACKING

Several options are available when using HTML5 geolocation. We can pass a third parameter to the `getCurrentPosition`and `watchPosition` methods, that will hold one or several of the following options:

Properties of the coords object	
enableHighAccuracy	A boolean (true/false) which indicates to the device that you wish to obtain its most accurate readings. in other words: use the GPS please! (However, this parameter may or may not make a difference, depending on your hardware, GPS availability, etc.)
maximumAge	The maximum amount of time (in milliseconds) the position may remain in the cache (this is appropriate as the device may cache readings to save power and/or bandwidth).
timeout	The maximum time (in milliseconds) for which you are prepared to allow the device to try to obtain a Geo location. After this timeout value has elapsed, the onError callback is called.

Example of use (see the explanations in the lines of comment):

```
// Just ask to turn GPS on, if available
navigator.geolocation.getCurrentPosition(onSuccess,onError,

 {enableHighAccuracy:true});
// maximumAge = 10 mins, the position can be cached for 10
mins,
// useful when in tunnels...When the device tries to get
// a position, if it does not succeed, then go on error
// immediately
navigator.geolocation.getCurrentPosition(onSuccess,onError,

{maximumAge:600000, timeout:0});
// Position will never come from the cache (maximumAge: 0),
and
// if after 0.1s the position could not be computed, then go
on
// error
navigator.geolocation.getCurrentPosition(onSuccess,onError,

 {maximumAge:0, timeout:100});
// Ask for GPS, cache for 30s, 27s before going on error...
watchId=navigator.geolocation.watchPosition(onSuccess,onError,

{enableHighAccuracy:true,maximumAge:30000, timeout:27000});
```