HTML5 - locate user with geolocation API

Introduction

On mobile devices, where the browser supports HTML5, it will offer good possibilities for creating implementations using the local storage API, geolocation API and any existing web service incorporating AJAX to form compelling appliances.

The Local Storage databases functionality has been introduced in a previous Wiki article here handling the basic usage of it, opening, accessing and creating the database.

Main idea for the local storage in HTML5 lies behind the fact that user related data can be easily stored and accessed. This without the need of building a separate server database and implementing back and forth communication with it to accomplish the required task of acting as a persistent data storage.

This method will bring the usual desktop/laptop world approach to the mobile devices.

For mashups, combining HTML5 and any of the vast amount of web services, which are there for almost any need or use case, could be transformed to full-blown web applications with good looking user interfaces and ease-of-use.

The more the HTML5 and CSS3 support is found in mobile phones and the incorporated web browsers, the more sense it makes to consider building great mashups.

Locating the user

The need to know where you are is one of the most used cases for mobile applications, and no wonder. When we think about the data that determines where we are, it often ends up with the two abbreviations ”lat” & ”lon”, so latitude and longitude.

However, these numeric values by themselves are not too easy to convert for anyone to just determine where one exactly or about is. Therefore this information is combined with a service that provides more value to that information, by determining something readable and understandable back; e.g. The street where the user actually is provides much more value, that is for sure.

As most of the services in web will take the latitude and longitude values as input, we will now check how to get them using the HTML5 geolocation API. For most of the use cases the getCurrentPosition method of the navigator object is enough. The code below shows how to use it to get the user's location. And this is the object and method definition:

```
navigator.geolocation.getCurrentPosition(successCallback, errorCallback, options);
```

The callbacks are there for handling the cases of success / error. In case of success, the function defined in ”success” will be passed with one parameter, which is type of Position. The call in itself for the getCurrentPosition is asynchronous.

The parameter that reaches the success function, will contain a timestamp property and a coords property. The coord property is type of Coordinates.

Then, the Coordinates object itself has many properties as follows:

```
longitude, latitude, altitude, accuracy, altitudeAccuracy, heading, speed
```

Note that this does not mean all of this may be accessible, as not all devices will necessarily support them. The lat, lon, accuracy however will most definitely be available, if the device supports fetching the geolocation at all.

In case of an error, the function that was defined for the error will get just one parameter, which is type of ”PositionError”. When you have an instance of PositionError, it will have two properties in itself, which are : code and message.

It is device dependent what will be in the message, but could be used for debugging. The code received most likely will be either value of 1,2 or 3 as follows:

```
1=PERMISSION_DENIED, 2=POSITION_UNAVAILABLE, 3=TIMEOUT
```

The timeout refers to the duration of what it took to define the user's location. The support is device dependent.

Example: use of getCurrentPosition

Please check the code on this wiki article for how to query the location for the user, it extends the use with pulling in Ovi Maps API to put an image on the location where the user has been located.

Note: this is supported on desktop/laptop use at the moment of writing (view the code by show-source-code on your browser).

(C) Copyright Nokia 2013. All rights reserved.

http://mapswidgets.com/html5geolocation.html

Timed location updates

When there is a need for periodic location updates, the watchPosition API can be used. The basic usage resembles to how the getCurrentPosition was used. Actually even the same parameters can be used. Main difference is that the there is an value of ID, which will be returned when watchPosition is called.

With the watchPosition, the clearWatch API comes useful. It uses the ID of the watchPosition – the browser sends updates using the function that was created for the success callback. This will be performed until clearWatch is called.

browser will keep sending updates to the success callback function that you passed it, until you call clearWatch.

Note: Continuous calls to location will not be mobile device battery-friendly operation.

Where the location updates are supported, this could be used to e.g. show a map and display the user location as it changes.

Note: using these APIs require support from the mobile browser on device.

(C) Copyright Nokia 2013. All rights reserved.

(C) Copyright Nokia 2013. All rights reserved.