W3C Geolocation API

Michael(tm) Smith
mike@w3.org

Overview

- API for use in Web applications (through JavaScript binding)
- enables Web application running on a remote server to access user location (latitude/longitude) data; e.g., for a user accessing the Web from a browser running on a mobile device (mobile handset, laptop, etc.)

How does a Web app get access to user location data?

The API requires browser support; either the device a browser is running on must directly expose one or more location-sensing mechanisms (e.g., GPS) to the browser, or some other third-party application must expose it to the browser (e.g., WiFi positioning through SkyHook)

Who's doing the work?

- W3C Geolocation Working Group
- the editor for the spec works for Google
- Google Gears has already shipped with support for the API

Where to get it

http://dev.w3.org/geo/api/spec-source.html

Requirements

- must provide location data in terms of a pair of latitude and longitude coordinates
- must provide information about the accuracy of the retrieved location data
- must support "one-shot" position updates

Requirements (more)

- must allow an application to register to receive repeated position updates
- must allow an application to cheaply query the last known position
- must provide a way for the application to receive updates about errors that may have occurred while obtaining a location fix

Requirements (more)

- must allow an application to specify a desired accuracy level of the location information
- must be agnostic to the underlying sources of location information
- should allow an application to request address information as part of the location data

Agnostic about location method

- The API does not expect or depend on any particular location method.
- Can work with any location-sensing mechanism exposed to the browser; might be on GPS, IP address, RFID, WiFi, Bluetooth MAC address, GSM/ CDMA cell IDs,...

Some use cases

- Show the user's position on a map
- Find points of interest in the user's area
- Turn-by-turn route navigation (like NaviTime)
- Push alerts when points of interest are in the user's vicinity
- Location-tagged status updates in social networking applications

Example: “One-shot” query

```
function showMap(position) {
  // Show a map centered at
  // at (position.latitude, position.longitude). }
// One-shot position request.
navigator.geolocation.getCurrentPosition(showMap);
```

Example: Position updates

```
function scrollMap(position) {
  // Scrolls the map so it is centered
  // at (position.latitude, position.longitude). }
// Request repeated updates.
var watchId =
navigator.geolocation.watchPosition(scrollMap);
function buttonClickHandler() {
  // Cancel when user clicks button
  navigator.geolocation.clearWatch(watchId); }
```

IDL: Geolocation interface

```
interface Geolocation {
  readonly attribute Position lastPosition;
  void getCurrentPosition(in
    PositionCallback successCallback);
  …
  int watchPosition(in
    PositionCallback successCallback);
  …
  void clearWatch(in int watchId); };
```

IDL: PositionCallback interface

```
interface PositionCallback {
  void handleEvent(in Position position); };
```

IDL: Position object

```
interface Position {
  readonly attribute double latitude;
  readonly attribute double longitude;
  readonly attribute double accuracy;
  readonly attribute double altitude;
  readonly attribute double altitudeAccuracy;
  readonly attribute double heading
  readonly attribute double velocity
  readonly attribute DOMTimeStamp timestamp; };
```

The End

Thanks!

mike@w3.org