HTML5 Geolocation

Sang Shin
Founder & Chief Instructor
JPassion.com
"Learn with Passion!"

Topics

- What is and why Geolocation?
- How it works
- Location data sources
- Accuracy
- Privacy
- Geolocation API
- Using maps
- Geolocation examples

What is and Why Geolocation?

What is Geolocation?

- W3C JavaScript API
- Built into the browser
  - All modern browsers support it
  - http://caniuse.com/#search=geolocation (~90% adoption: 12/2015)
- Must give an “opt-out” option to a user
  - To protect user privacy
  - “opt-out” option can be configurable in browser setting, however
- Provides approximate location
  - If you need high accuracy location, you might have to use native application

Geolocation Use-cases

- Show the user's position on a map
- Find points of interest in the user's area
- Annotating content with location information
- Turn-by-turn navigation
- Location-tagged status updates in social networking applications
- Auto-select country/state/city/zipcode

Geolocation Sample App Ideas

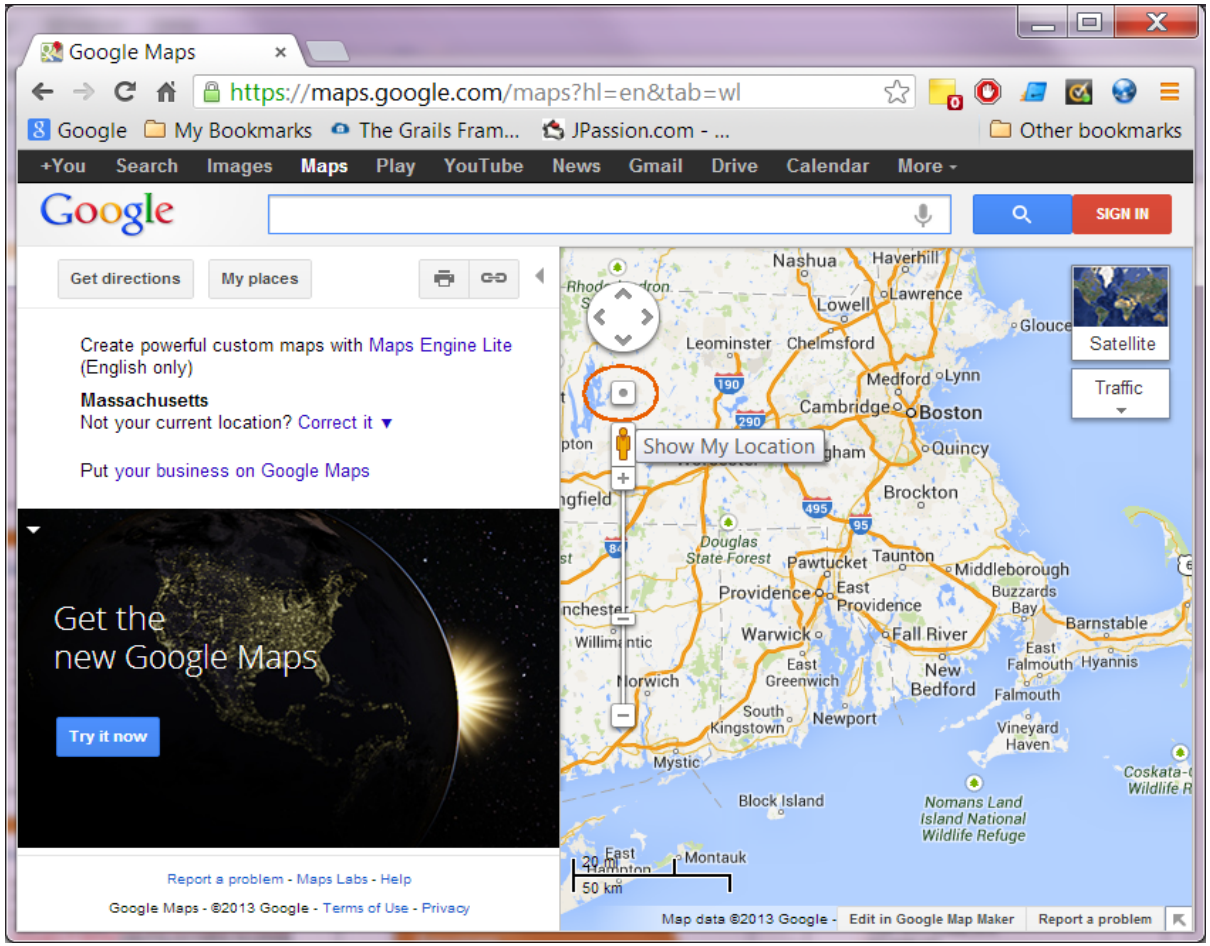
- Public transportation sites can list nearby bus stops and metro locations (Many cities are already offering these services)
- Late night out? Taxi or car service Web sites can find where you are, even if you don't know (Uber and Lyft are already doing this.)
- Shopping sites can immediately provide estimates for shipping costs.
- Content sites can more accurately determine the language and dialect of search queries.
- Real estate sites can present average house prices in a particular area, a handy tool when you're driving around to check out a neighborhood or visit open houses. (Zillow is already doing this)
- Online games can blend reality into the game play by giving users missions to accomplish in the real world.

source: http://html5center.sourceforge.net/12-Cool-HTML5-Geolocation-Ideas

How it works

How Desktop Browser determines your location

- Google Map – Show My Location



How Browser determines your location

- When you visit a location-aware website (like Google Map), Browser will ask you if you want to share your location.
- If you consent, Browser gathers information using various means of the device nearby wireless access points (assuming your device has wireless connection) and/or your computer's IP address
- Then Browser sends this information to the default geolocation service provider, Google Location Services, to get an estimate of your location
- That location estimate is then shared with the requesting website
- If your wireless connection is turned off, or you're at a stationary computer, all calculations are based on the IP address

Location Information Sources

Sources of Location Metadata

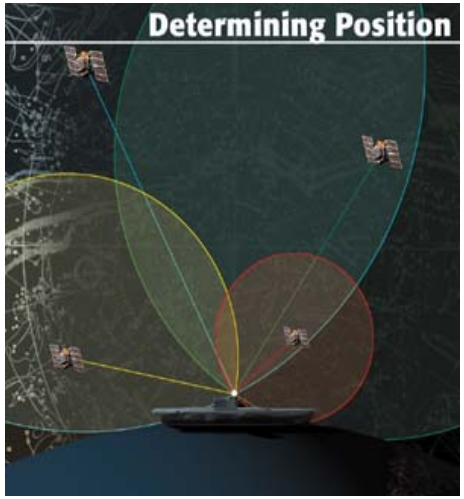
- A device can use any combination of the following sources
  - IP address
  - Coordinate triangulation
    - Global Positioning System (GPS)
    - Wi-Fi access point
    - GSM or CDMA cell phone IDs
  - User defined

IP address-based Location Data

- How it works
  - HTTP request has IP address of the sender
- Pros
  - Available everywhere
- Cons
  - Might not be accurate – accuracy granularity is only to the city level
  - The IP address might be the one of the service provider not the client machine itself
- Example
  - Website selects a default language (or customized contents) based on the IP addresses of its clients

GPS-based Location Data

- How it works
  - Devices passively receive signals from multiple GPS satellites and then compute its location
- Pros
  - High accuracy
- Cons
  - Require an unobstructed view of the sky, so they generally can be used outdoors and they often do not perform well within forested areas or near tall buildings
  - Might take long time to get the location data
  - Require GPS-hardware



Wi-Fi based Location Data

- How it works
  - Devices computes the location using the distances from multiple Wi-Fi access points (It is called WiFi-triangulation)
- Pros
  - High accuracy
  - Works indoors
  - Fast
- Cons
  - Does not work well where Wi-Fi access points are scarce (rural areas)

Cell Phone Location Data

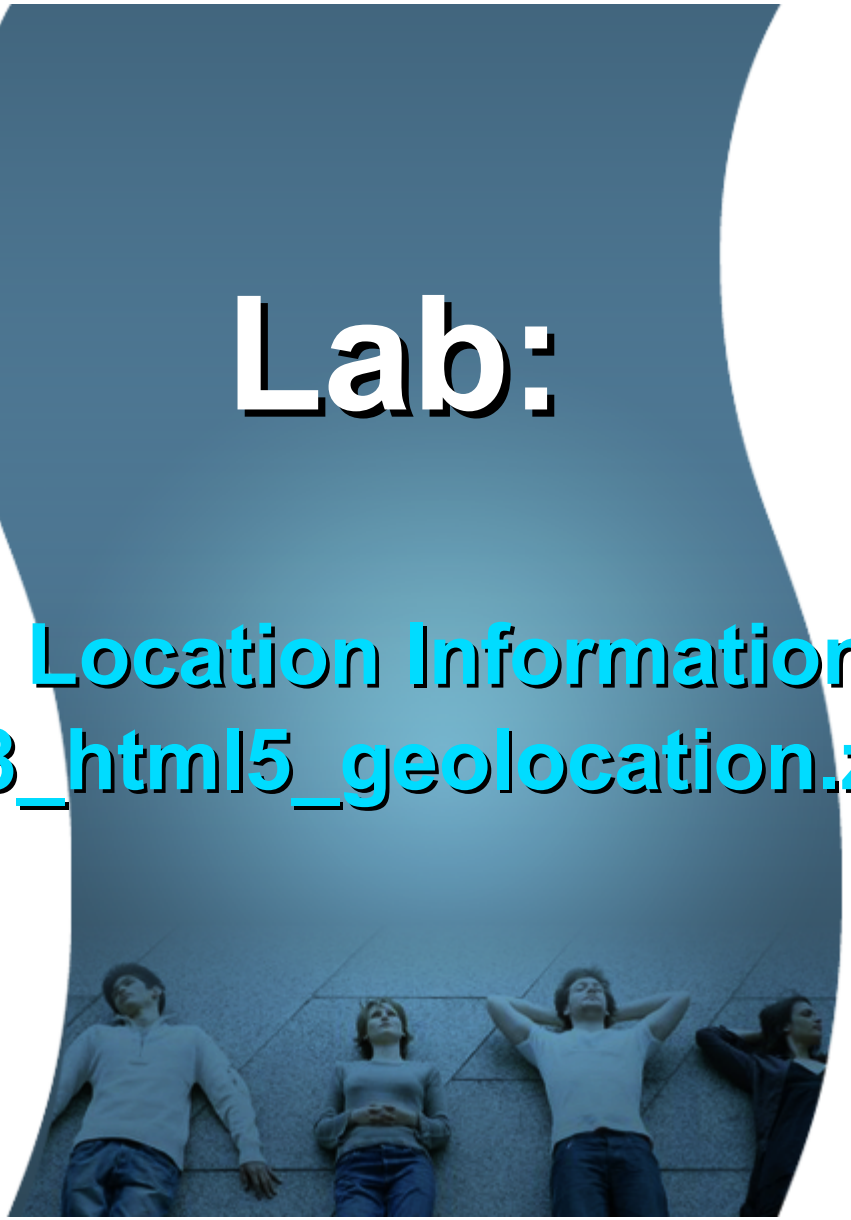
- How it works
  - Devices computes the location using the distances from multiple Cell phone towers
- Pros
  - Fair accuracy
  - Works indoors
  - Fast
- Cons
  - Does not work where cell phone towers are scarce (rural areas)

User-defined Location Data

- How it works
  - Users provide their location related data – Address, zipcode, etc – to the application
- Pros
  - Might be more accurate if more specific location data is provided
- Cons
  - Might not be accurate if wrong data is provided

Lab:

Exercise 1: Location Information Sources
1228_html5_geolocation.zip



Accuracy

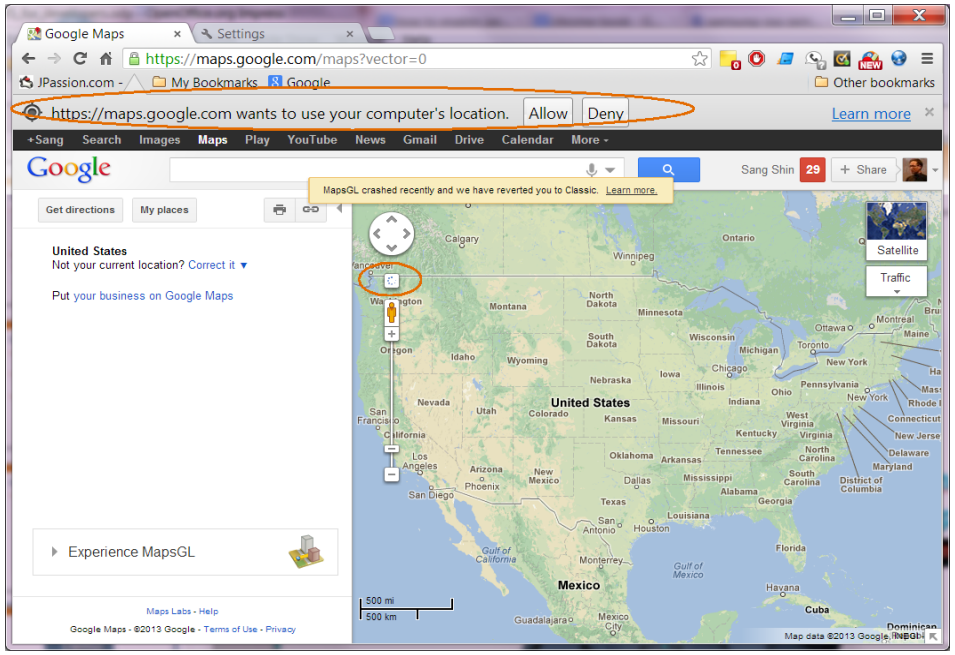
Factors Influencing Accuracy

- Type of browser
  - Each browser provides its own implementation
- Type of device
  - Phone, Tablet, Laptop, desktop, etc
- Wi-Fi enabled or not
- GPS enabled or not
- Internet connectivity
- VPN

Privacy

Privacy

- HTML5 Geolocation specification “mandates” that a mechanism is provided to protect the privacy
  - > Location information should not be made available unless the user grants the permission



Geolocation API

Geolocation API

- One-time position request

```
void getCurrentPosition(PositionCallback successCallback,
                        optional PositionErrorCallback errorCallback,
                        optional PositionOptions options);
```

- Repeating position request

```
long watchPosition(PositionCallback successCallback,
                   optional PositionErrorCallback errorCallback,
                   optional PositionOptions options);

void clearWatch(long watchId);
```

One-time position request

- The *getCurrentPosition()* method takes one, two or three arguments.
- When called, it must immediately return and then asynchronously attempt to obtain the current location of the device.
  - If the attempt is successful, the *successCallback* must be invoked with a new *Position* object, reflecting the current location of the device
  - If the attempt fails, the *errorCallback* must be invoked with a new *PositionError* object, reflecting the reason for the failure.

```
if (navigator.geolocation) {
  navigator.geolocation.getCurrentPosition(
      function(position) {      // successful callback
          var latLng = new google.maps.LatLng(
              position.coords.latitude, position.coords.longitude);
          var marker = new google.maps.Marker({position: latLng, map: map});
          map.setCenter(latLng);
      },
      errorHandler              // error callback
  );
}
```

Repeating Position request

- The watch operation must first attempt to obtain the current location of the device
- It then must continue to monitor the position of the device and invoke the appropriate callback every time this position changes
- The watch operation must continue until the clearWatch method is called with the corresponding identifier.

```
//Repeated position updates
var watchId = navigator.geolocation.watchPosition(
                                                  updateLocation,
                                                  handleLocationError);

// do something with the location updates!
// ...

// Stop receiving location updates
navigator.geolocation.clearWatch(watchId);
```

Optional Request Parameters

- enableHighAccuracy
  - Request the HTML5 Geolocation service to use a higher accuracy-detection mode if available (only supports true or false)
- Timeout
  - Tells the browser the maximum amount of time it is allowed to calculate the current location
- maximumAge
  - Indicates how old a location value can be before the browser must attempt to recalculate (default is zero, meaning that the browser must attempt to recalculate a value immediately.)

Position Data Returned (1)

- latitude and longitude (in decimal degrees)
  - Geographic coordinates specified
- altitude (in meters)
  - Height of the position
- accuracy (in meters)
  - Accuracy level of the latitude and longitude coordinates
  - Values returned by an implementation should correspond to a 95% confidence level.
- altitudeAccuracy (in meters)
  - Values returned by an implementation should correspond to a 95% confidence level.

Position data returned (2)

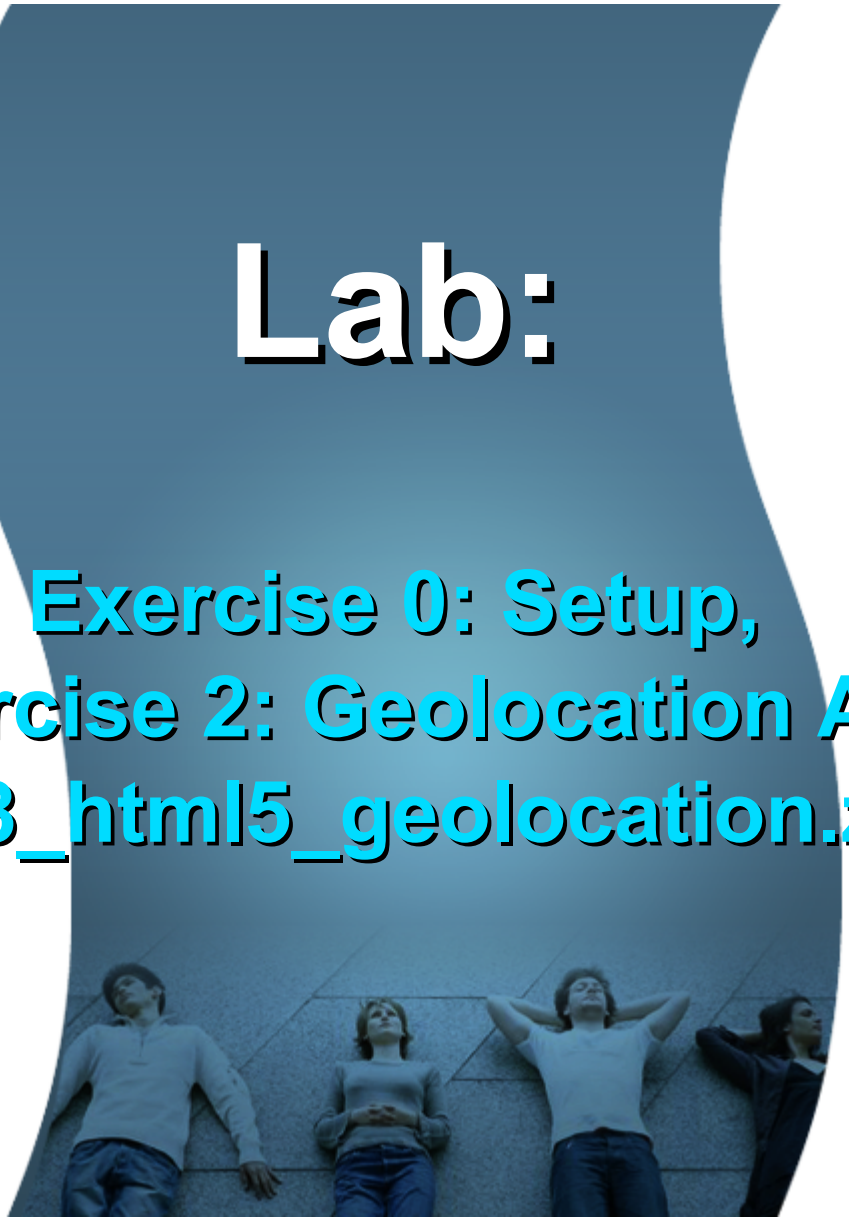
- heading attribute (in degrees)
  - direction of travel of the hosting device and is specified in degrees, where 0° ≤ heading < 360°, counting clockwise relative to the true north
  - If the hosting device is stationary (i.e. the value of the speed attribute is 0), then the value of the heading attribute must be NaN
  - If not available, can be computed by the application using data from watchPosition(..)
- speed attribute (in meters per second)
  - magnitude of the horizontal component of the hosting device's current velocity and is specified in meters per second
  - If not available, can be computed by the application using data from watchPosition(..)

Error Codes

- PERMISSION_DENIED (numeric value 1)
  - The location acquisition process failed because the document does not have permission to use the Geolocation API.
- POSITION_UNAVAILABLE (numeric value 2)
  - The position of the device could not be determined. For instance, one or more of the location providers used in the location acquisition process reported an internal error that caused the process to fail entirely.
- TIMEOUT (numeric value 3)
  - The length of time specified by the timeout property has elapsed before the implementation could successfully acquire a new Position object.

Lab:

Exercise 0: Setup,
Exercise 2: Geolocation API
1228_html5_geolocation.zip



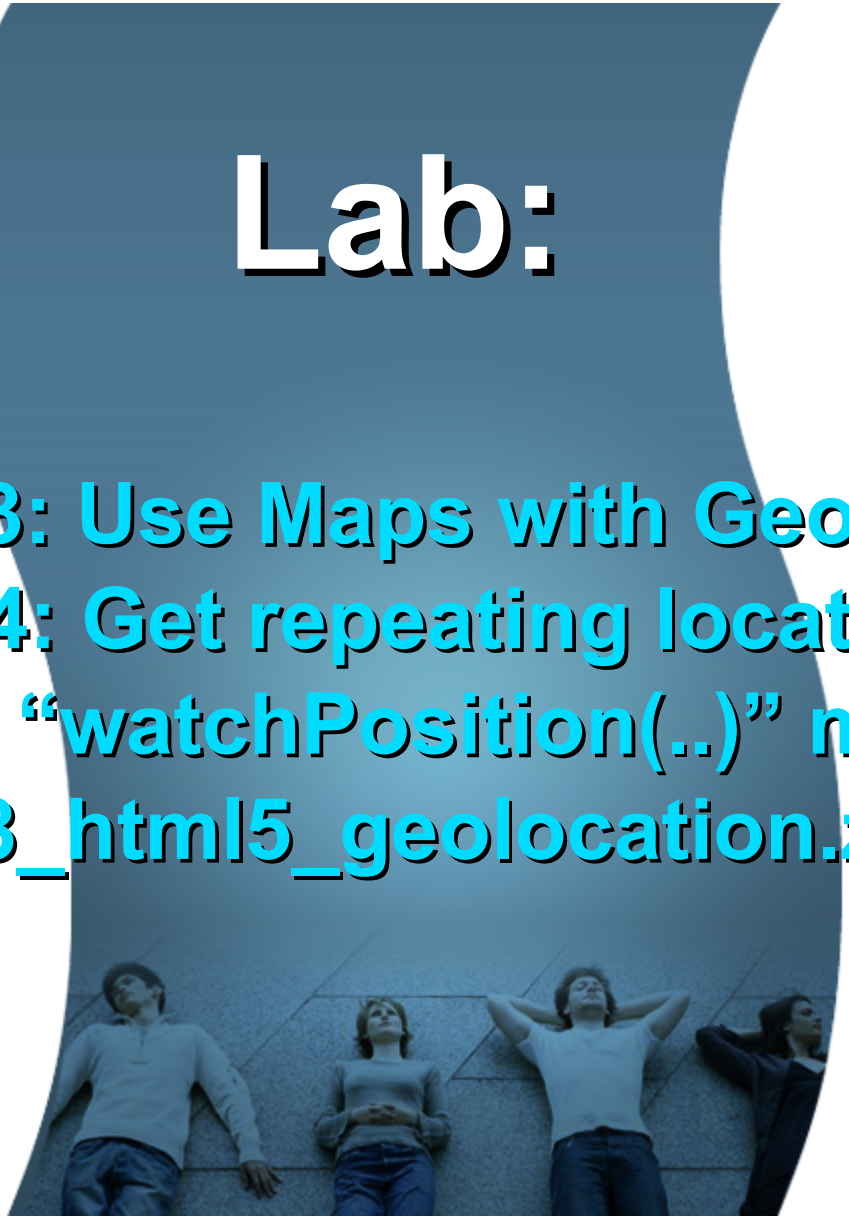
Using Maps

Map sources

- Google map
- Bing map
- Yahoo map
- Nokia map

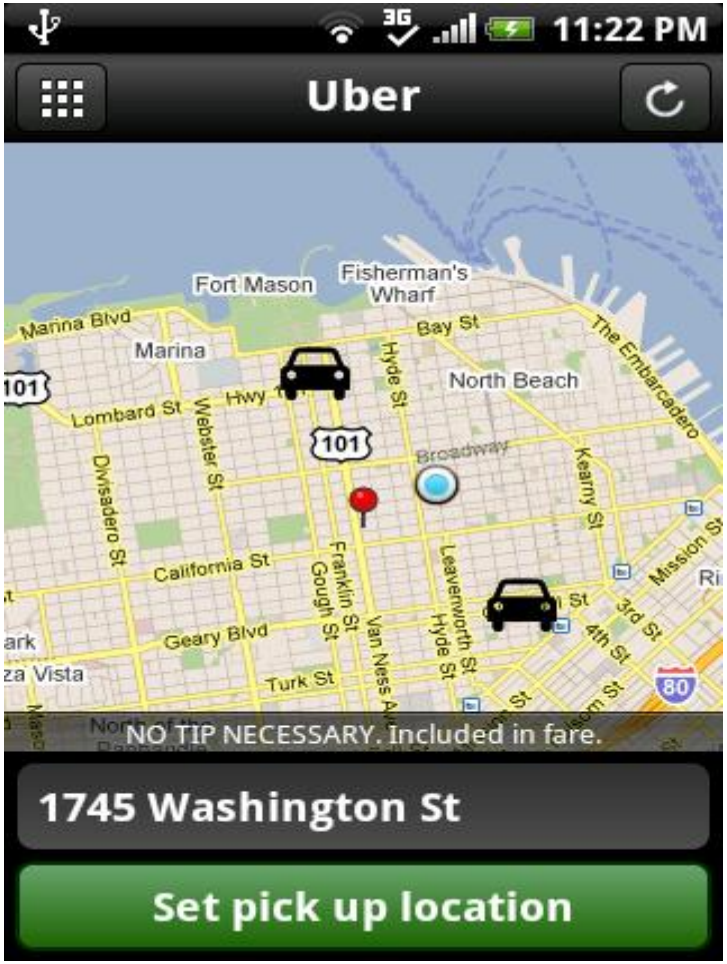
Lab:

Exercise 3: Use Maps with Geolocation
Exercise 4: Get repeating location data through “watchPosition(..)” method
1228_html5_geolocation.zip



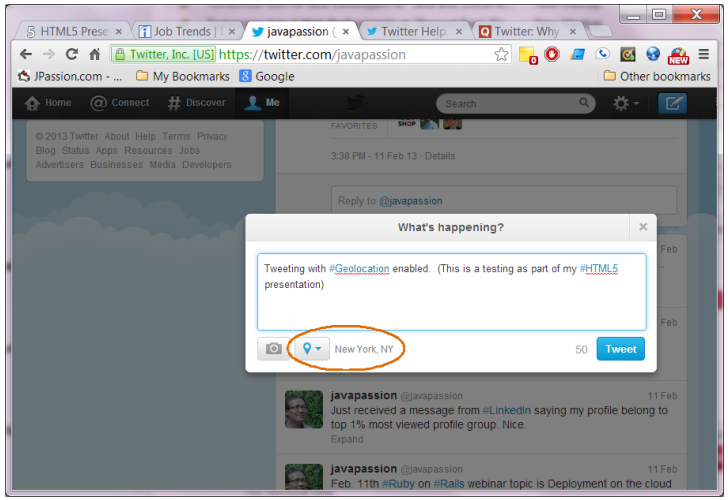
Geolocation Examples

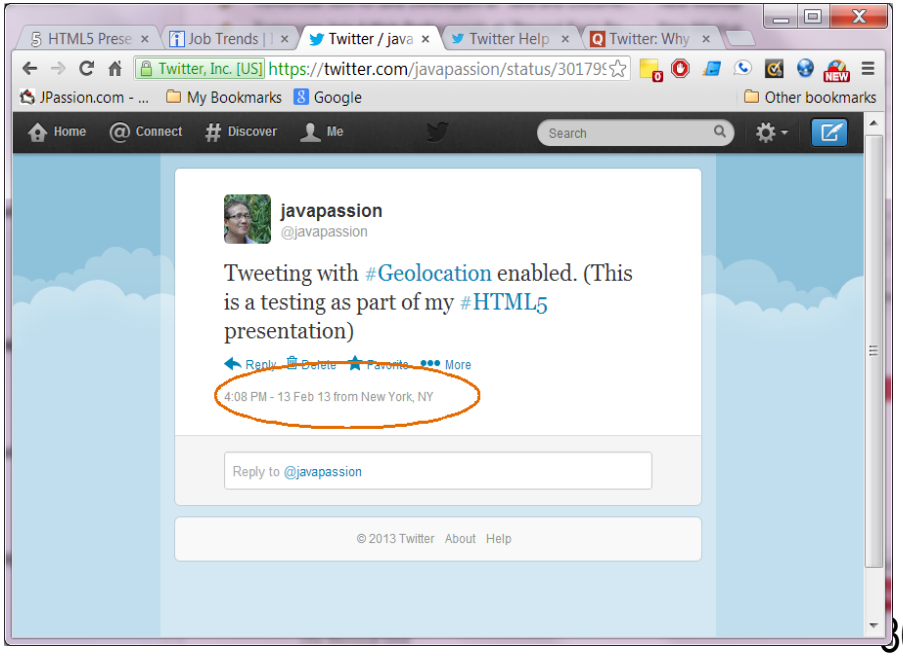
Geolocation Example: Uber



Geolocation Example: tweeter

- Twitter will be able to:
  - Show your followers the location you are tweeting from as part of your Tweet
  - Enable delivery of location-specific trends and Tweets, that are personalized for your location,





Learn with Passion!
JPassion.com