DIG 4104C: WEB DESIGN WORKSHOP

Lecture 05: Extending web apps: HTML5 Geolocation; Drag & Drop

Summer 2015

D. Novatnak

Outline

I. Geo-location Basics
II. Privacy
III. Error Handling
IV. Positioning
V. Fall-Back
VI. Mapping

Objectives

- Students will be exposed to geolocation methods in HTML5 to extend web based applications based on user location
- Students will be exposed to drag & drop methods in HTML5 to extend web based applications to permit more natural interactions
- Students will receive instruction on the language and commands necessary to get location and drag and drop within the client's browser
- View and analyze code examples

At the end of this lecture, you should be able to….

- Store data on the client's browser for later retrieval with the Web Storage API
- Access data on local file systems from the client's browser for later retrieval with the File API
- Access data on the client's browser for processing through the Web Storage and File APIs

GEOLOCATION

FIND YOURSELF!

Mapping

- Google
- Yahoo
- Open Street Map
- Microsoft Bing Maps
- OpenLayers
- Cloudmade
- Mapstraction

Geo-location Basics

- Standard Methods
  - Cell Tower Triangulation or Trilateration
    - 1 – X Towers Used (Accuracy Varies) ~3000m
    - Quick Startup and lower power consumption
    - No sky view required
  - GPS
    - 4 - X Satellites Used of 24 available
    - Slower startup and higher battery consumption
    - Sky view required
    - Better Accuracy ~3m
    - Elevation Possible

Basics

- Wi fi Triangulation
  - Requires Wi fi Access
  - Higher Battery Consumption
- IP Lookup
  - Requires IP Access
  - Can be very inaccurate
- A-GPS
  - Combination of GPS and Land Based System
  - Altitude Accuracy

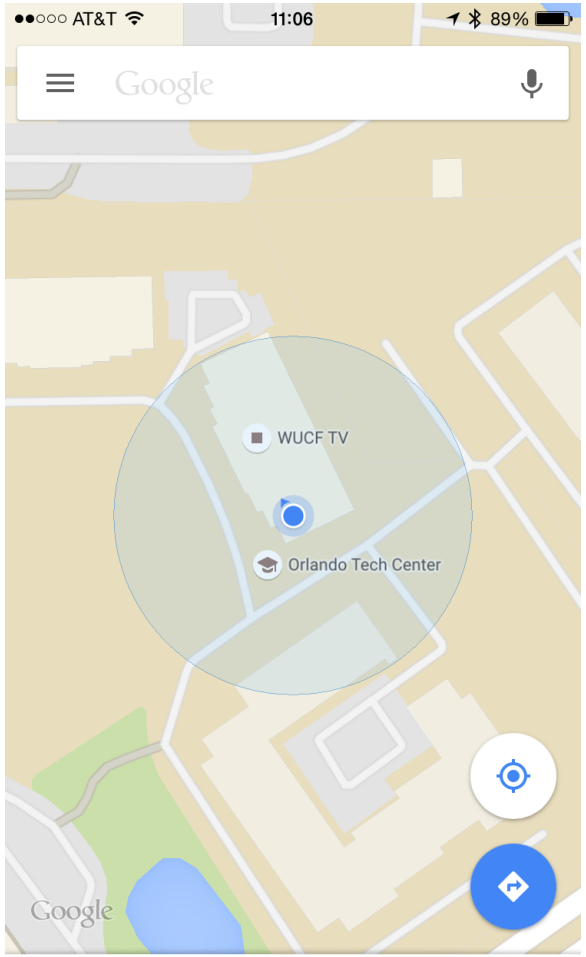
Cell Tower Techniques

- CID – Cell tower Id
- AOA – Angle of Arrival
- TOA – Time of Arrival
- TDOA – Time Difference of Arrival
- RSS – Received Signal Strength
- Multi path Fingerprint
- OTDOA – Observed Time Difference of Arrival
- U-TDOA – Uplink Time Difference Of Arrival

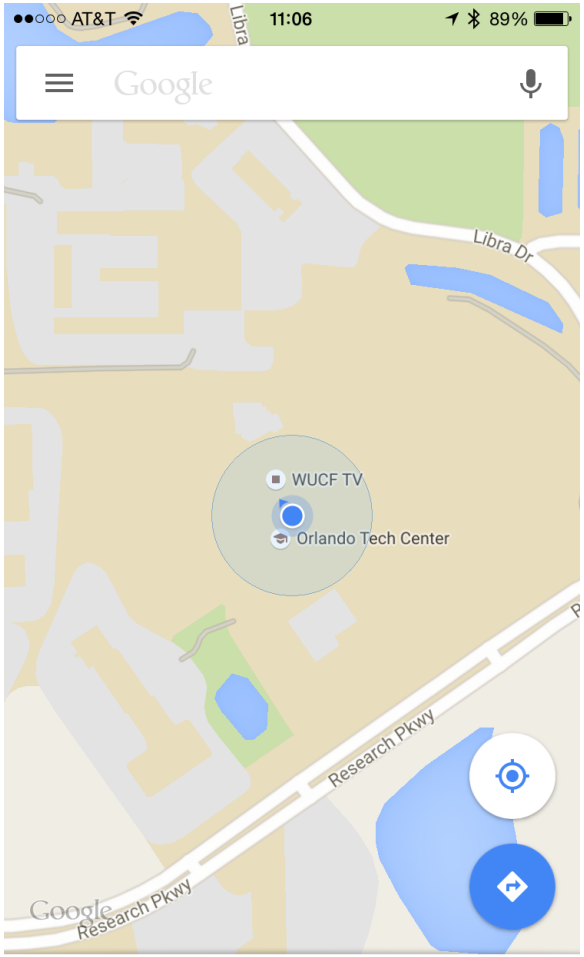
Example

- Google Maps
- 3 circles
  - Outside Circle (One tower)
  - Inside Circle (multiple tower)
  - Push Pin GPS

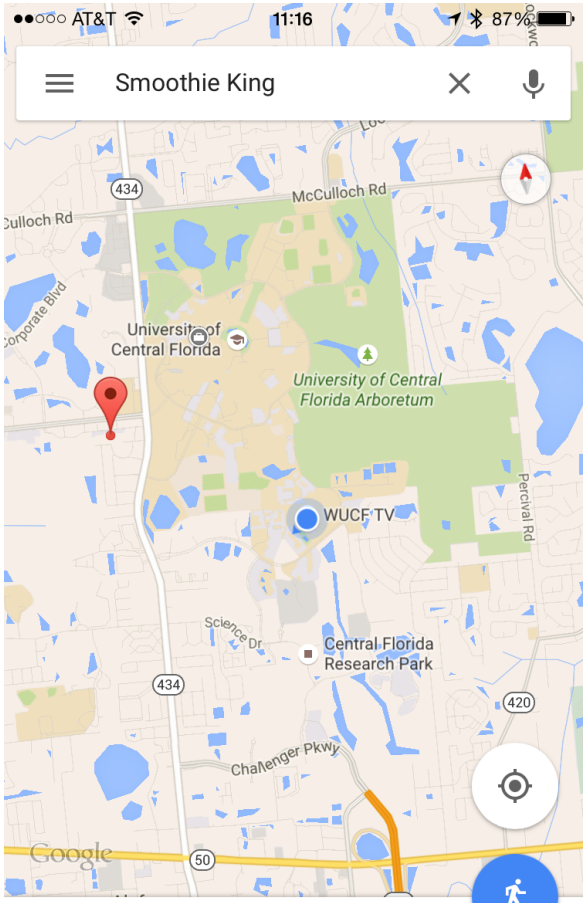
Google Maps



Explore around you



Around The Marquee



Smoothie King

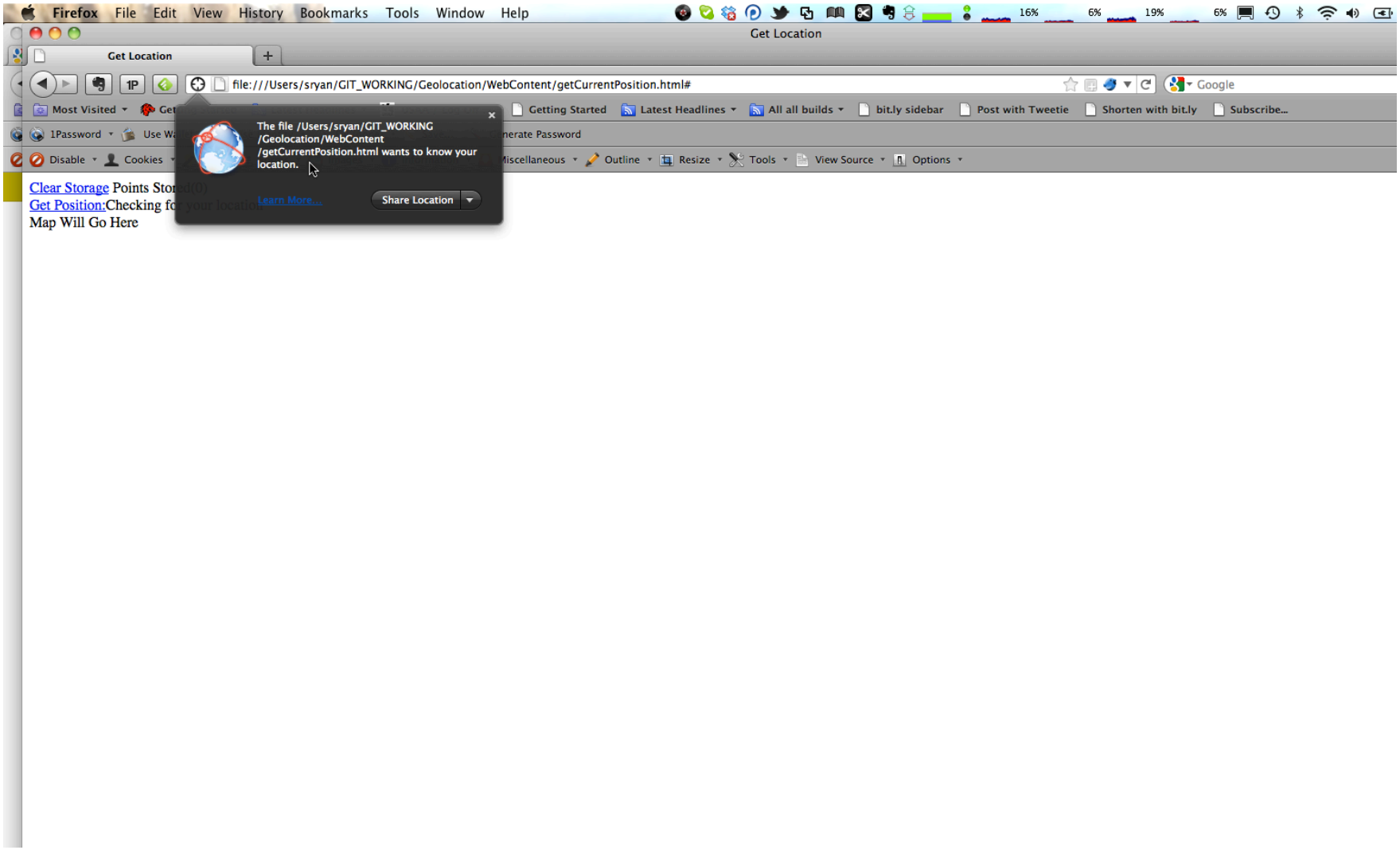
3.7 ★★★½☆ (8) · $$

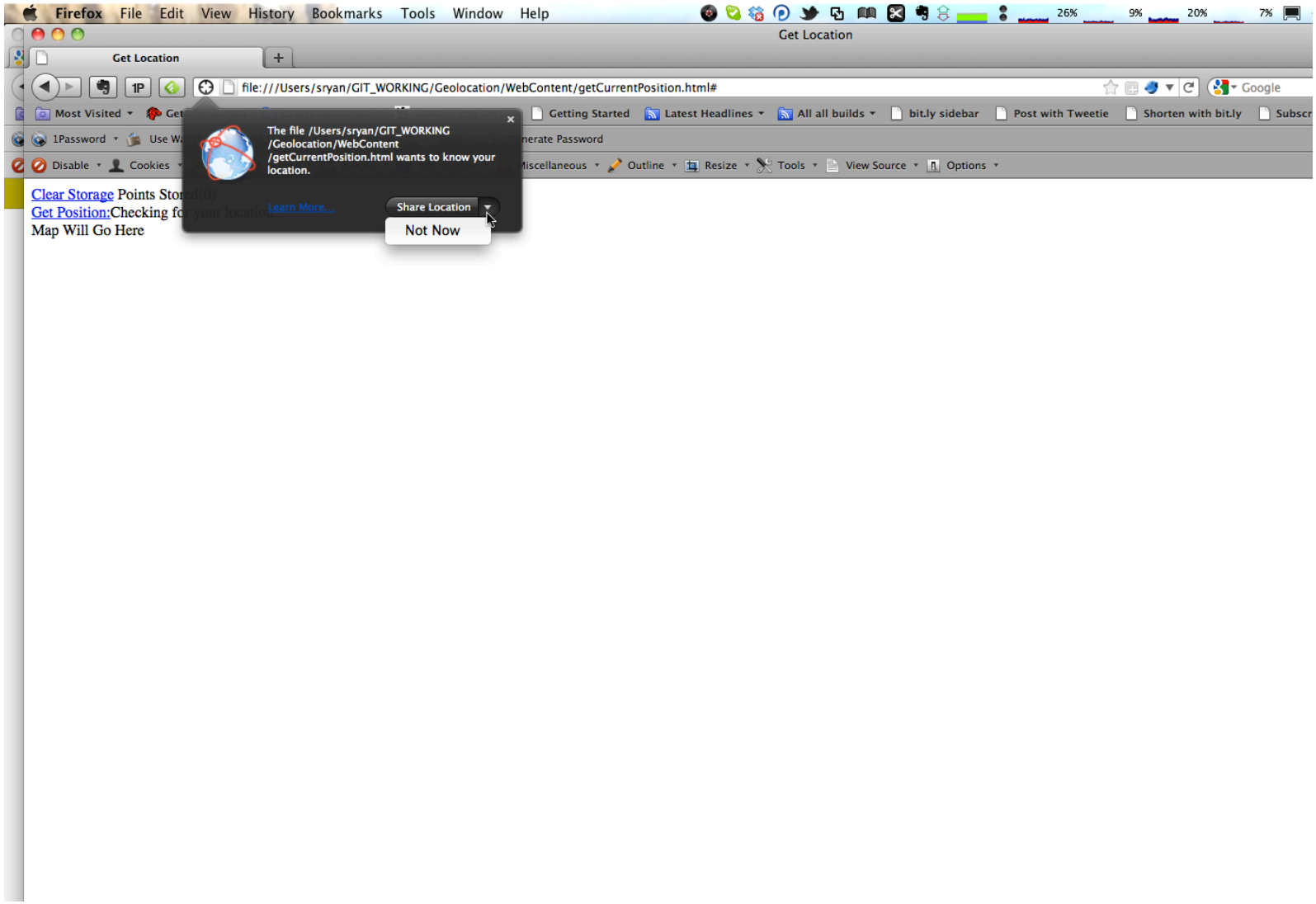
34 min

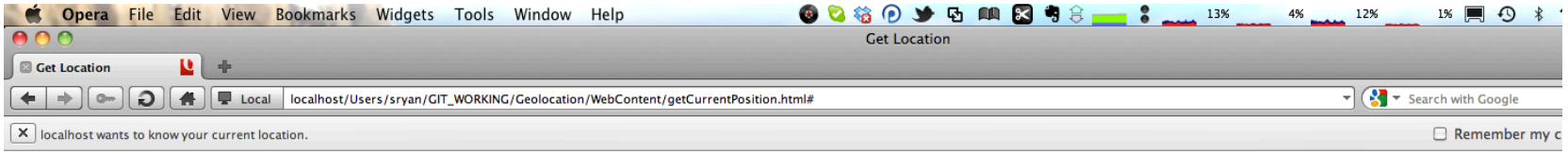
Privacy

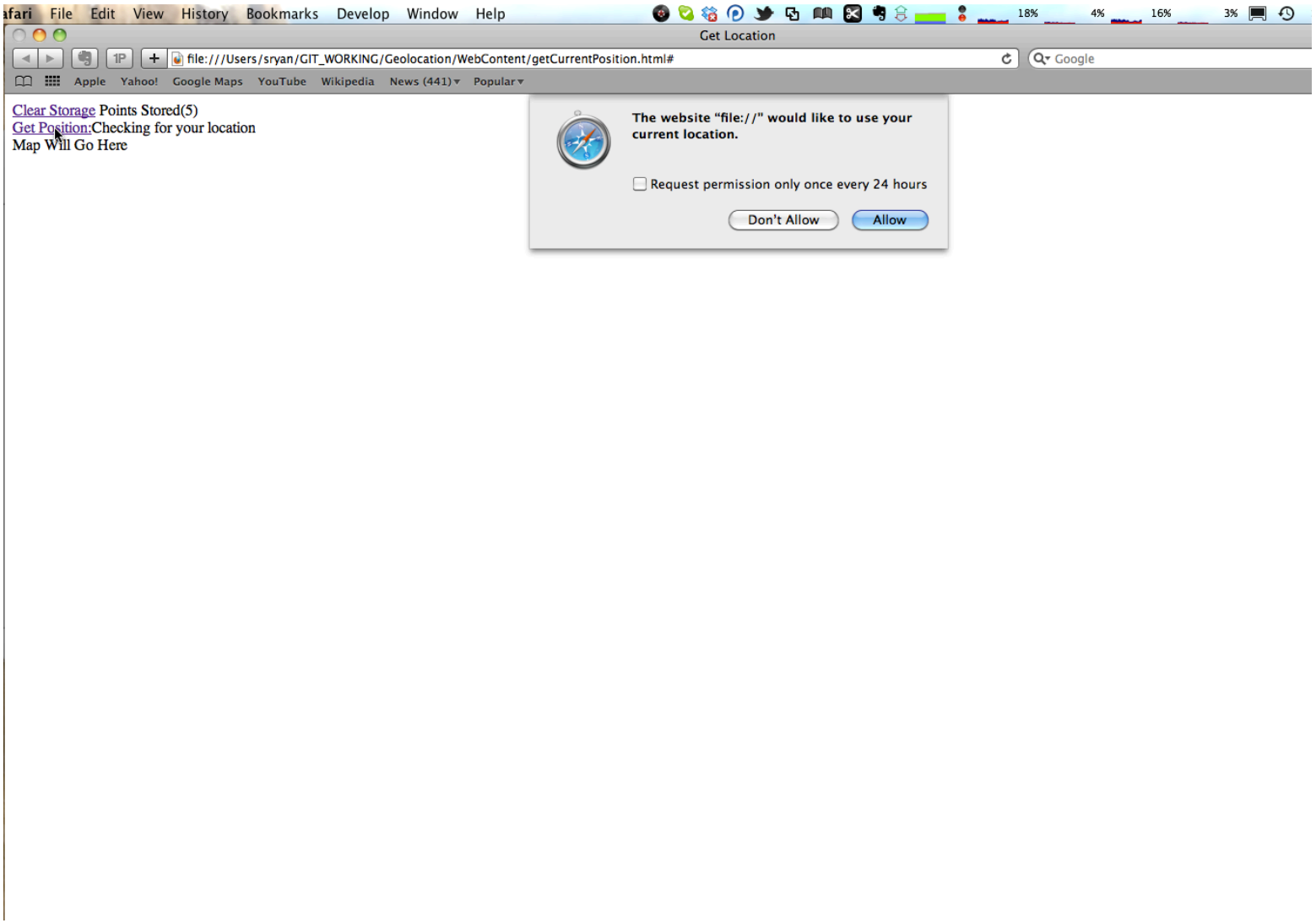
- Prompt for every domain
- Cannot bypass via code
- Non modal and tab specific
- Display requesting website with help
- Allow, Don’t Allow and Remember
- Errors are handled in the error callback

Sample Output




Firefox File Edit View History Bookmarks Tools Window Help
Get Location
file:///Users/sryan/GIT_WORKING/Geolocation/WebContent/getCurrentPosition.html#
Most Visited
Getting Started
Latest Headlines
All all builds
bit.ly sidebar
Post with Tweetie
Shorten with bit.ly
1Password
Disable
Cookies
Miscellaneous
Outline
Resize
Tools
View Source
Options
The file /Users/sryan/GIT_WORKING /Geolocation/WebContent /getCurrentPosition.html wants to know your location.
Learn More...
Share Location
Not Now
Clear Storage Points Stored(0)
Get Position:Checking for your location...
Map Will Go Here


Opera File Edit View Bookmarks Widgets Tools Window Help
13% 4% 12% 1%
Get Location
Get Location
Local
localhost/Users/sryan/GIT_WORKING/Geolocation/WebContent/getCurrentPosition.html#
Search with Google
localhost wants to know your current location.
Remember my c
Clear Storage Points Stored(5)
Get Position:Checking for your location
Map Will Go Here


afari File Edit View History Bookmarks Develop Window Help
18% 4% 16% 3%
Get Location
file:///Users/sryan/GIT_WORKING/Geolocation/WebContent/getCurrentPosition.html#
Google
Apple Yahoo! Google Maps YouTube Wikipedia News (441) Popular
Clear Storage Points Stored(5)
Get Position:Checking for your location
Map Will Go Here
The website "file://" would like to use your current location.
Request permission only once every 24 hours
Don't Allow
Allow

Error Handling

- Handled in Error Callback
  - Code
    - PERMISSION_DENIED (1)
    - POSITION_UNAVAILABLE (2)
    - TIMEOUT (3)
    - UNKNOWN_ERROR (0)
  - Message

Get Position

- getCurrentPosition(function, error_function,options)
  - Latitude, longitude, altitude (optional)
  - Accuracy
  - Altitude Accuracy(optional)
  - Heading (optional)
  - Speed (optional)
  - timestamp

Watch Position

- Same call as getPosition
  - Returns a number that should be stored
  - Returns position when location changes
    - Device determines sample interval
  - Use number to stop watching
    - ClearWatch(number)

Options

- Javascript object
  - enableHighAccuracy
    - Can fail on some phones due to permission
  - Timeout (After permission is handled)
  - maximumAge (caching)

Sample Code

```
function showMap(position) {
      // Show a map centered at (position.coords.latitude,
position.coords.longitude).
    }

    // One-shot position request.
    navigator.geolocation.getCurrentPosition(showMap);
```

```
function scrollMap(position) {
      // Scrolls the map so that it is centered at
(position.coords.latitude, position.coords.longitude).
    }

    // Request repeated updates.
    var watchId = navigator.geolocation.watchPosition(scrollMap);

    function buttonClickHandler() {
      // Cancel the updates when the user clicks a button.
      navigator.geolocation.clearWatch(watchId);
    }
```

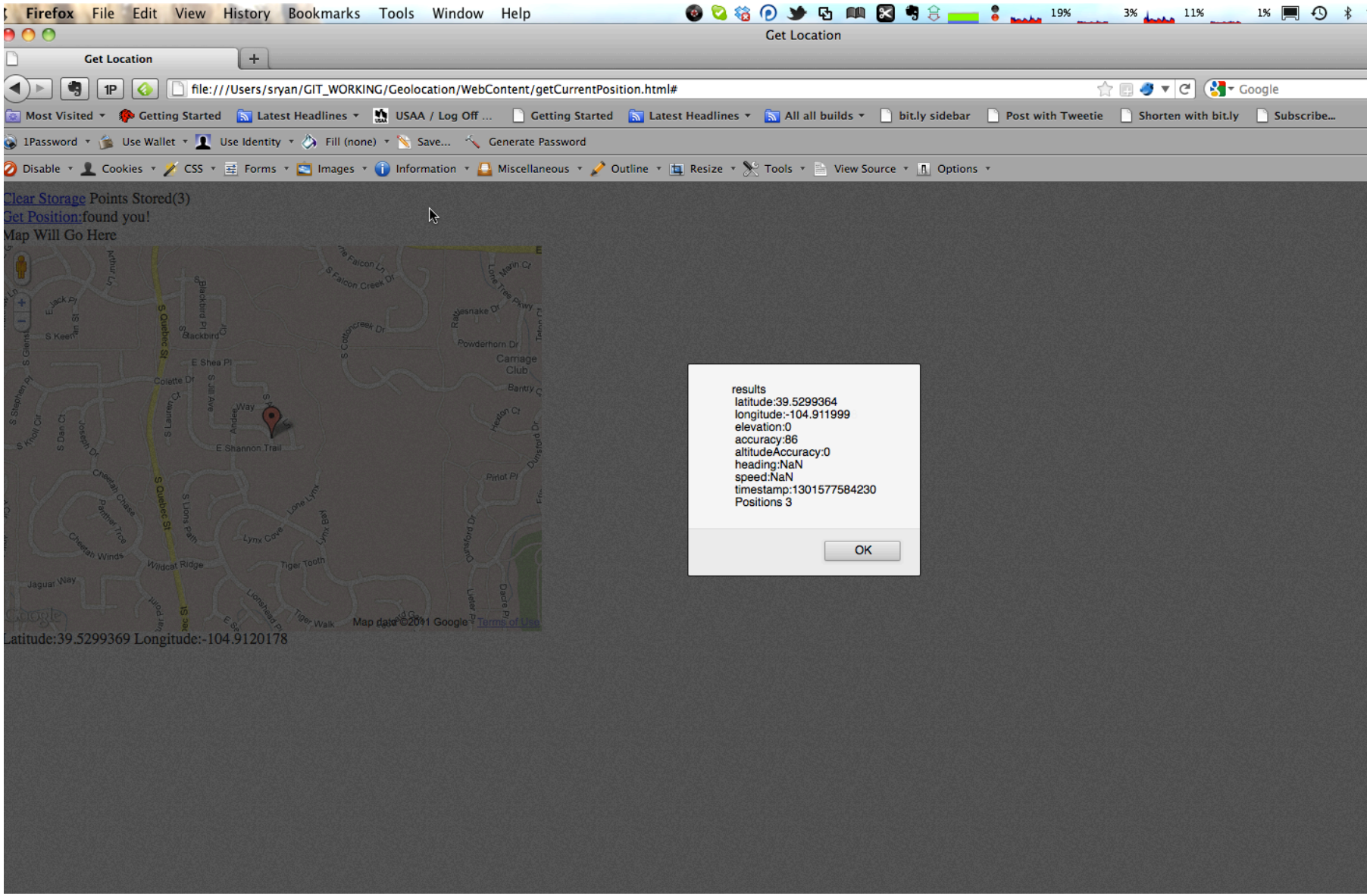
Sample Code

```
 function scrollMap(position) {
      // Scrolls the map so that it is centered at
(position.coords.latitude, position.coords.longitude).
    }

    function handleError(error) {
      // Update a div element with error.message.
    }

    // Request repeated updates.
    var watchId = navigator.geolocation.watchPosition(scrollMap,
handleError);

    function buttonClickHandler() {
      // Cancel the updates when the user clicks a button.
      navigator.geolocation.clearWatch(watchId);
    }
```

Sample Code

```
function success_callback(position) {
        var latitude = position.coords.latitude;
        var longitude = position.coords.longitude;
        var elevation = position.coords.altitude;
        var accuracy = position.coords.accuracy;
        var altitudeAccuracy = position.coords.altitudeAccuracy;
        var heading = position.coords.heading;
        var speed = position.coords.speed;
        var timestamp = position.timestamp;
}
```

```
function error_callback(positionError) {
        alert("Error" + positionError.code + " Message "
                        + positionError.message);
}
```


Firefox File Edit View History Bookmarks Tools Window Help
Get Location
Get Location
file:///Users/sryan/GIT_WORKING/Geolocation/WebContent/getCurrentPosition.html#
Clear Storage Points Stored(3)
Get Position:found you!
Map Will Go Here
Map data ©2011 Google Terms of Use
Latitude:39.5299369 Longitude:-104.9120178
results
latitude:39.5299364
longitude:-104.911999
elevation:0
accuracy:86
altitudeAccuracy:0
heading:NaN
speed:NaN
timestamp:1301577584230
Positions 3
OK

NEED MORE?

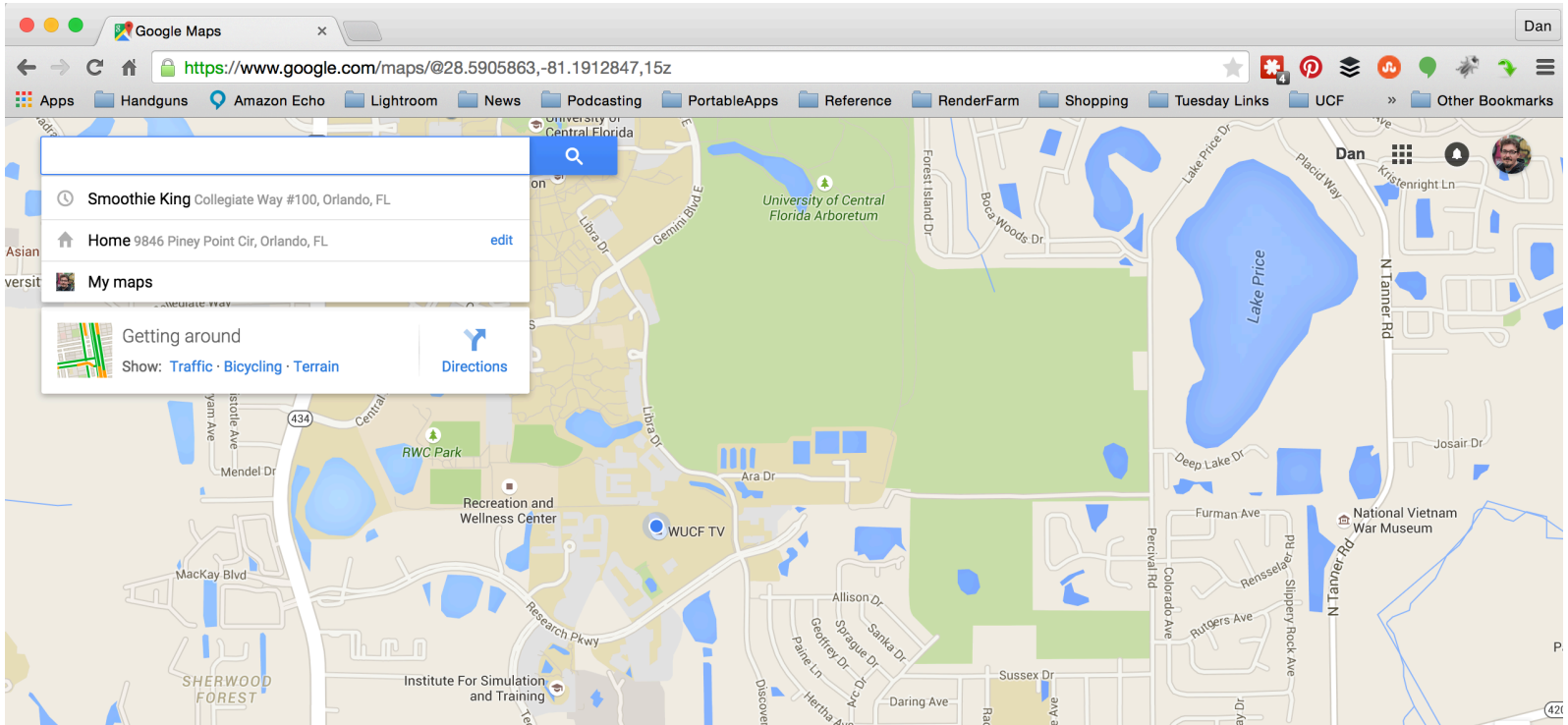
Google it! No, seriously, Google can help!

Enter the Google API

- Google Code is Google's site for developer tools, APIs and technical resources. The site contains documentation on using Google developer tools and APIs—including discussion groups and blogs for developers using Google's developer products.
- There are APIs offered for almost all of Google's popular consumer products, like Google Maps, YouTube, Google Apps and others.

What is Google Maps?

- Like I have to tell you!
  - An enormous amount of searchable geographic information (gazetteer)
  - A way of organizing the worlds information geographically



Google Maps API

- API – Application Programming Interface
  - Allows different software components to communicate with each other
  - Library that includes tools, variables, data structures, object classes, etc.
- Google Maps API
  - Web version utilizes JavaScript
  - Versions also available for Android and iOS

Google Maps API, cont.

- No cost usage limits and restrictions
  - Service must be freely and publicly accessible to end users
  - 25,000 map loads per day
  - 2,500 Geocoding requests per day
  - 2,500 Directions requests per day
- Google Maps API for Business available for those that do not meet, or exceed these requirements

Google Maps API, cont.

- Obtain API Key
  - Instructions to create your API key:
    - Visit the APIs Console at https://code.google.com/apis/console and log in with your Google Account.
    - Click the **Services** link from the left-hand menu.
    - Activate the **Google Maps API** service.
    - Click the **API Access** link from the left-hand menu. Your API key is available from the **API Access** page, in the **Simple API Access** section. Maps API applications use the **Key for browser apps**.

Loading Google Maps API

- The URL contained in the script tag is the location of a JavaScript file that loads all of the symbols and definitions you need for using the Google Maps API
- The key parameter contains your application's API key
- The sensor parameter indicates whether this application uses a sensor (such as a GPS locator) to determine the user's location

```
<script type="text/javascript" src="https://maps.googleapis.com/maps/
api/js?key=YOUR_API_KEY&sensor=SET_TO_TRUE_OR_FALSE">
</script>
```

Map DOM Element

- Creates a named ***div*** element and obtains a reference to this element in the browser's document object model (DOM)
- A ***div*** element defines a section of a webpage

```
<div id="map_canvas"></div>
```

CSS

- Indicates that the map container (named map-canvas) should take up 95% of the height and 75% of the width of the HTML body

```
<style type="text/css">
     html { height: 100% }
     body { height: 100%; margin: 0; padding: 0 }
     #map_canvas { height: 95%; width: 75% }
</style>
```

Map Options

- Creates a map options object to contain map initialization variables

```
var myOptions = {
        center: new google.maps.LatLng(41.638026,-87.48000),
        zoom: 12,
        mapTypeId: google.maps.MapTypeId.ROADMAP
};
```

Map Options: Latitudes and Longitudes

```
center: new google.maps.LatLng(41.638026,-87.48000)
```

- Centers the map on a specific point
- Creates a LatLng object to hold this location by passing the location's coordinates in the order { latitude, longitude }
- Must be in decimal degrees

Map Options: Zoom Levels

```
zoom: 12
```

- Initial resolution at which to display the map
- 0 corresponds to a map of the Earth fully zoomed out
- Higher zoom levels zoom in at a higher resolution

Map Options: Map Type

- ROADMAP displays the normal, default 2D tiles of Google Maps.
- SATELLITE displays photographic tiles.
- HYBRID displays a mix of photographic tiles and a tile layer for prominent features (roads, city names).
- TERRAIN displays physical relief tiles for displaying elevation and water features (mountains, rivers, etc.).

```
mapTypeId: google.maps.MapTypeId.ROADMAP
```

Map Object

- Map class is the JavaScript class that represents a map
- Creates a new instance of this class using the JavaScript new operator
- Obtains a reference to this element via the document.getElementById() method.
- This code defines a variable, map, and assigns that variable to a new Map object, also passing in options defined within the mapOptions object

```
var map = new
google.maps.Map(document.getElementById("map_canvas"),myOptions);
```

Loading the Map

```
<body onload="initialize()">
```

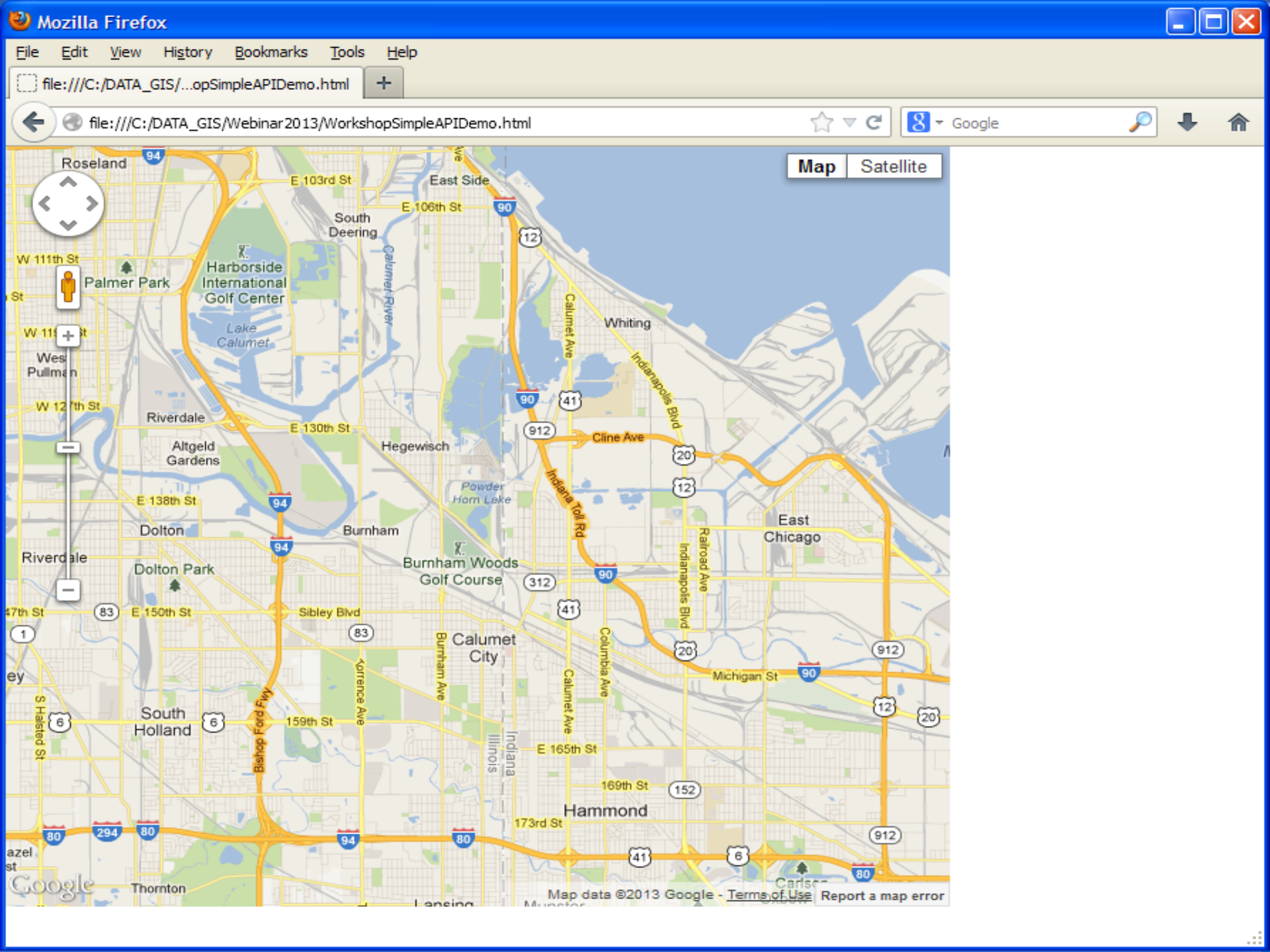
- Places map on the page after the page has fully loaded
- Executes the function, initialize(), which constructs the Map object once the <body> element of the HTML page receives an onload event
- The body tag's onload attribute is an example of an event handler

WorkshopSimpleAPICode - Notepad

File Edit Format View Help

```
<!DOCTYPE html>
<html>
  <head>
        <style type="text/css">
                html { height: 100% }
                body { height: 100%; margin: 0; padding: 0 }
                #map_canvas { height: 95%; width: 75% }
        </style>

        <script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?key=YOUR_UNIQUE_KEY&sensor=false">
        </script>

        <script type="text/javascript">
                function initialize() {
                        var myOptions = {
                                center: new google.maps.LatLng(41.638026,-87.48000),
                                zoom: 12,
                                mapTypeId: google.maps.MapTypeId.ROADMAP
                        };

                        var map = new google.maps.Map(document.getElementById("map_canvas"),myOptions);

                };

        </script>

   </head>

   <body onload="initialize()">

         <div id="map_canvas"></div>

   </body>

</html>
```

Ln 10, Col 96


Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/DATA_GIS/...opSimpleAPIDemo.html
file:///C:/DATA_GIS/Webinar2013/WorkshopSimpleAPIDemo.html
Google
Map
Satellite
Roseland
E 103rd St
East Side
E 106th St
South Deering
Calumet River
W 111th St
Palmer Park
Harborside International Golf Center
Lake Calumet
Whiting
Calumet Ave
Indianapolis Blvd
Riverdale
E 130th St
Hegewisch
Cline Ave
Altgeld Gardens
Powder Horn Lake
Indiana Toll Rd
East Chicago
E 138th St
Dolton
Burnham
Railroad Ave
Burnham Woods Golf Course
Dolton Park
E 150th St
Sibley Blvd
Calumet City
Columbia Ave
Michigan St
Burnham Ave
South Holland
159th St
Bishop Ford Fwy
Indiana Illinois
E 165th St
169th St
Hammond
173rd St
Thornton
Map data ©2013 Google - Terms of Use
Report a map error

Maps API – Layers

- Your Data:
  - KML layer
  - Fusion Tables layer
  - Heatmap layer
  - GeoRSS layer
- Google Data:
  - Traffic layer
  - Transit layer
  - Bicycling Layer
  - Weather and Cloud layer
  - Panoramio layer

Maps API – Layers (cont'd)

- To add a layer to a map, you only need to call setMap(), passing it the map object :
  - trafficLayer.setMap(map);
- To remove a layer, call setMap(), passing null :
  - trafficLayer.setMap(null);

Layer Code - KML

- KML can be stored locally or anywhere on the web
- Multiple (>10) KML layers can be displayed on a single map

```
var url1 = https://sites.google.com/site/hsdgis/CityBoundary.kml;
var BoundaryLayer=new google.maps.KmlLayer(url1);
BoundaryLayer.setMap(map);
```

Layer Code – Fusion Tables

- Selecting the name of the column in the Fusion Table that contains geographic information
- Long cryptic string is the ID of the Fusion Table
- 5 Fusion Tables can be displayed on a single map

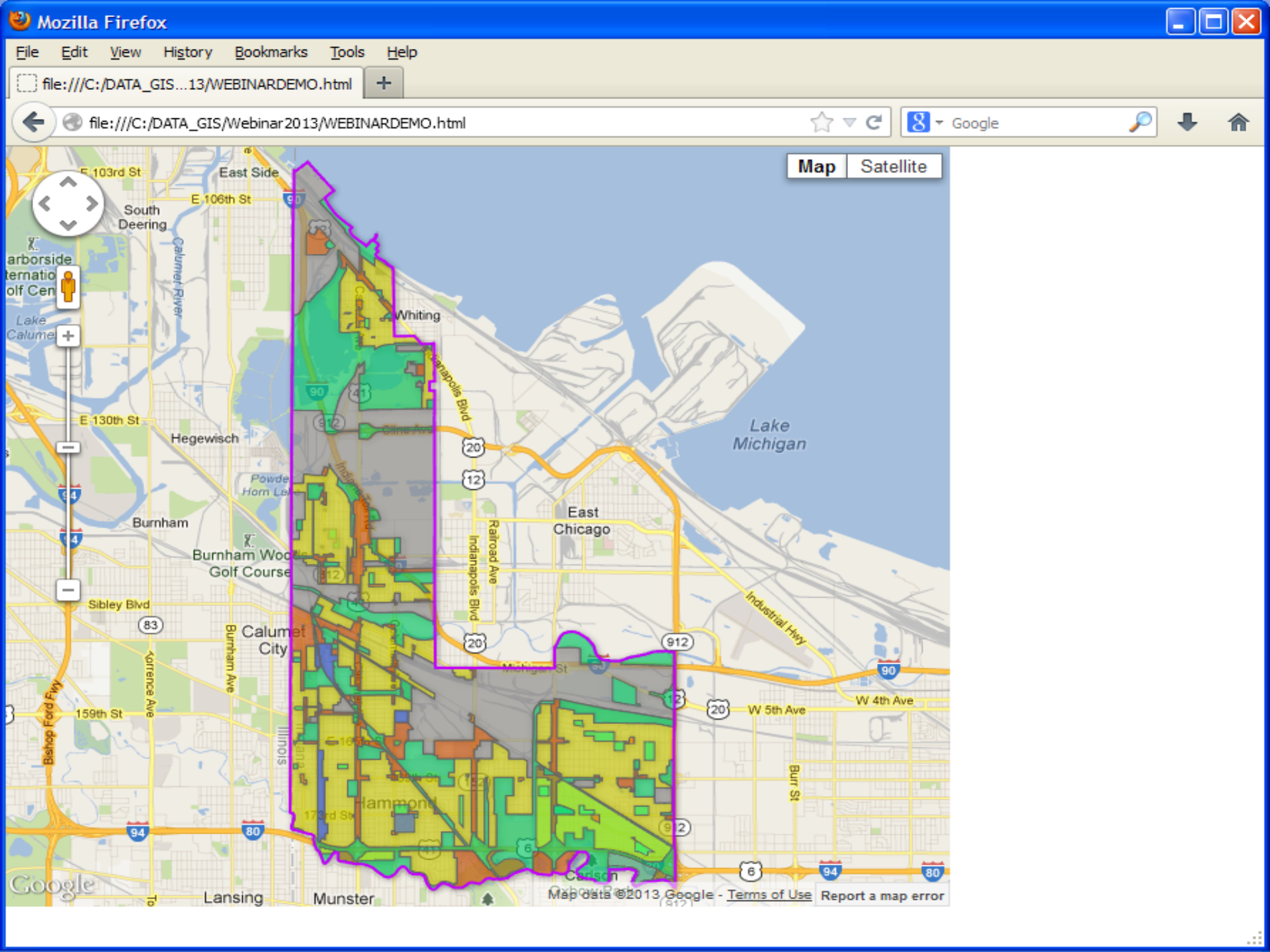
```
var ZoningLayer=new google.maps.FusionTablesLayer({
        query: {
                select: 'geometry',
                from: '1BVsTFSd16ggYS7BMM2NJ1r5Ua5j2w6NIzX6a_zk'
        }
});
ZoningLayer.setMap(map);
```

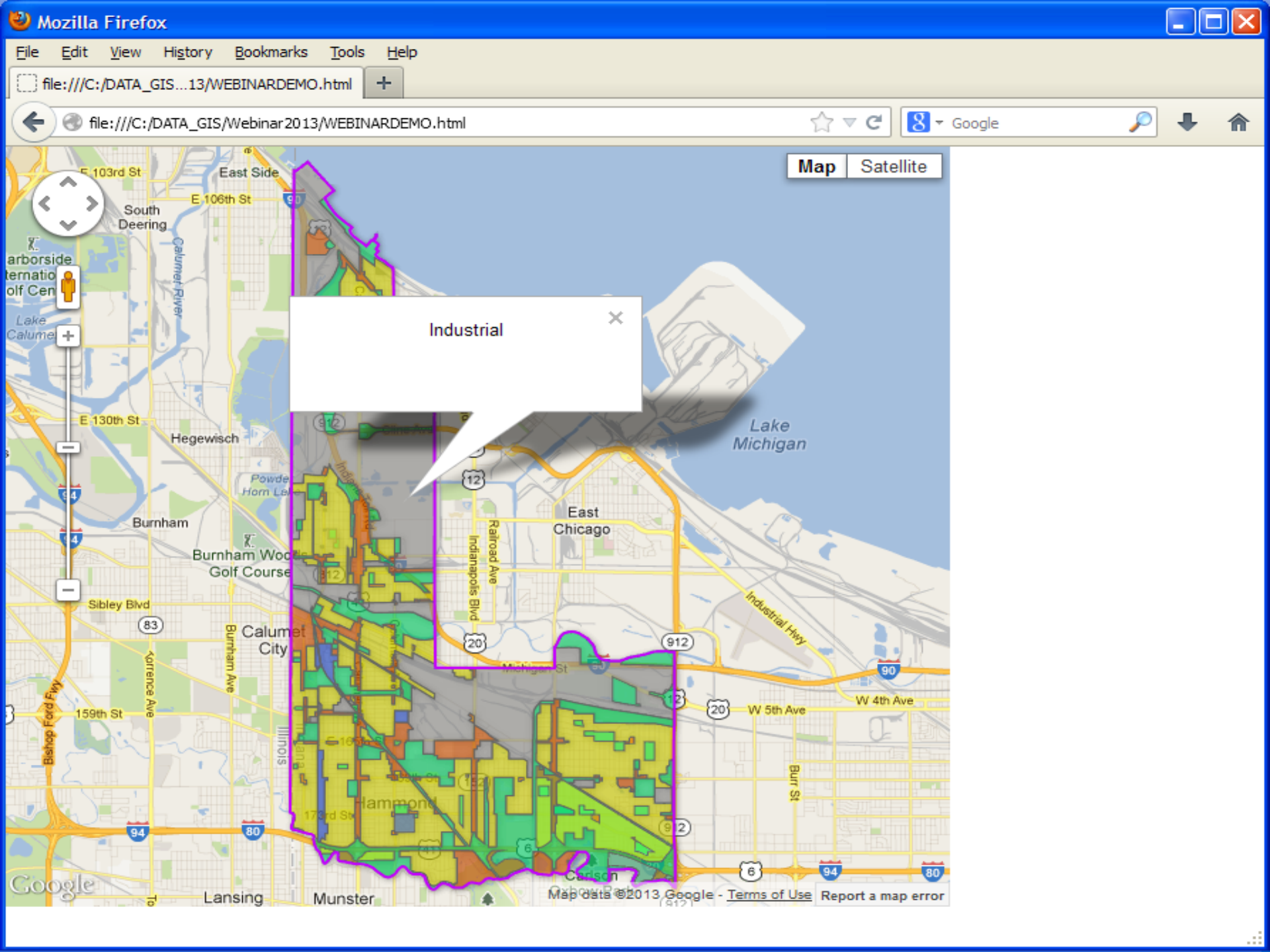
WebinarAPICode - Notepad

File Edit Format View Help

```
<!DOCTYPE html>
<html>
  <head>

    <style type="text/css">
                html { height: 100% }
                body { height: 100%; margin: 0; padding: 0 }
                #map_canvas { height: 95%; width: 75% }
    </style>

    <script type="text/javascript" src="http://maps.googleapis.com/maps/api/js?key=YOUR_UNIQUE_KEY&sensor=false">
    </script>

    <script type="text/javascript">
        function initialize() {

                var myOptions = {
                        center: new google.maps.LatLng(41.638026,-87.48000),
                        zoom: 12,
                        mapTypeId: google.maps.MapTypeId.ROADMAP
                };

                var map = new google.maps.Map(document.getElementById("map_canvas"),myOptions);

                var ZoningLayer=new google.maps.FusionTablesLayer({
                        query: {
                                select: 'geometry',
                                from: '1BVsTFSd16ggYS7BMM2NJ1r5Ua5j2w6NIzX6a_zk'
                                }
                });

                ZoningLayer.setMap(map);

                var url1 = "https://sites.google.com/site/hsdgis/CityBoundary.kml";

                var BoundaryLayer=new google.maps.KmlLayer(url1,{preserveViewport:true});

                BoundaryLayer.setMap(map);
        };
    </script>

  </head>

  <body onload="initialize()">
        <div id="map_canvas"></div>
  </body>

</html>
```

Ln 49, Col 1


Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/DATA_GIS...13/WEBINARDEMO.html
file:///C:/DATA_GIS/Webinar2013/WEBINARDEMO.html
Google
Map
Satellite
East Side
South Deering
E 103rd St
E 106th St
Calumet River
Whiting
Lake Michigan
E 130th St
Hegewisch
Burnham
Burnham Woods Golf Course
East Chicago
Indianapolis Blvd
Railroad Ave
Industrial Hwy
Sibley Blvd
Calumet City
159th St
Burnham Ave
Bishop Ford Fwy
W 5th Ave
W 4th Ave
Burr St
Hammond
Lansing
Munster
Google
Map data ©2013 Google - Terms of Use
Report a map error


Mozilla Firefox
File Edit View History Bookmarks Tools Help
file:///C:/DATA_GIS...13/WEBINARDEMO.html
file:///C:/DATA_GIS/Webinar2013/WEBINARDEMO.html
Google
Map
Satellite
Industrial
East Side
South Deering
E 103rd St
E 106th St
Calumet River
Lake Michigan
E 130th St
Hegewisch
Burnham
Burnham Woods Golf Course
Sibley Blvd
Calumet City
159th St
East Chicago
Railroad Ave
Indianapolis Blvd
Industrial Hwy
W 5th Ave
W 4th Ave
Hammond
Lansing
Munster
Map data ©2013 Google - Terms of Use
Report a map error

API Map Methods

- getCenter()
- getBounds()
- setCenter(*latlng:LatLng*)
- setMapTypeId(mapTypeId:*MapTypeId*|*string*)
- setZoom(zoom:*number*)
- fitBounds(bounds:*LatLngBounds*)

Example: ***map.setZoom(12);***

API Map Overlays

- Markers
- Polylines
- Polygons
- Ground Overlays
- Info Windows

API Map Markers

- Markers identify locations on the map
- By default, they are represented with a standard icon, a red pin
- Example:

```
var marker = new google.maps.Marker({
  position: new google.maps.LatLng(-25.363, 131.044),
  map: map
});
```

Fall Back Alternatives

- Geo.js
- IP Location Resolution

HTML5 DRAG & DROP

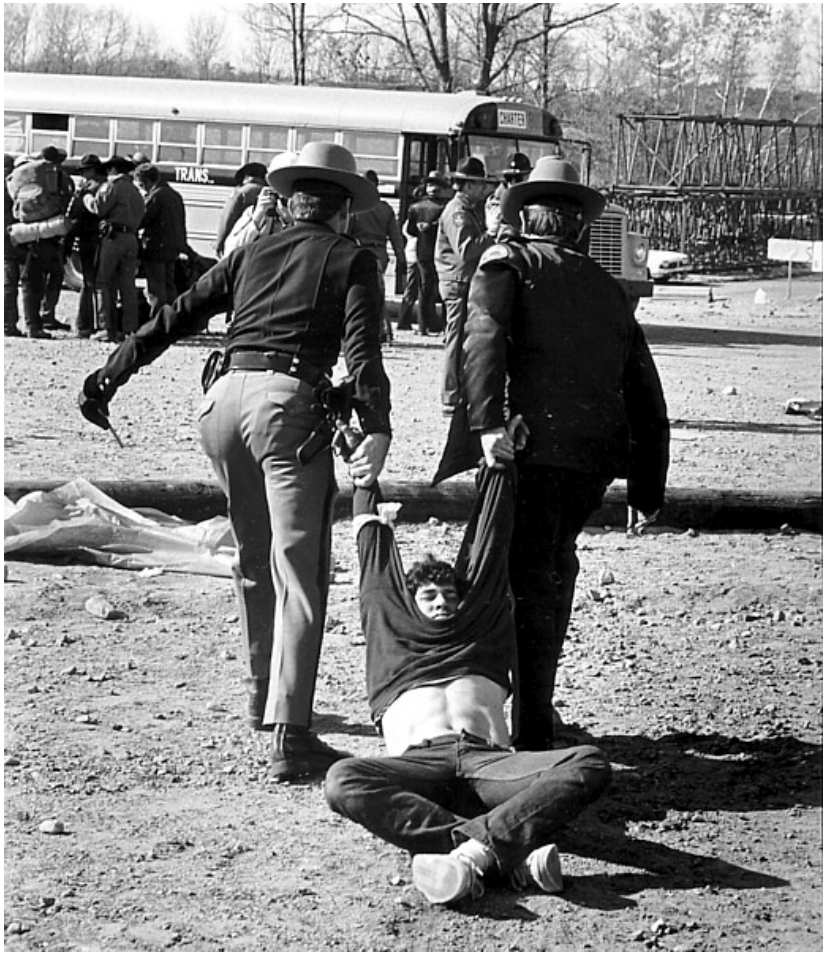
Throw things around…

Drag and Drop API

- HTML5 comes with a Drag and Drop API that brings native Drag and Drop support to the browser
- This API can also drag elements from browser to external applications and vice-versa

At minimum, drag and drop takes ...

1. Something to drag
2. A dragstart event
3. A dragover event
4. A drop event



Drag and Drop

- <element> attribute draggable="true"
- Events:
  - dragstart
  - dragstop
  - dragenter
  - dragleave
  - Dropend

- Let the user know they can drag it In CSS:
  - [draggable=true] {
  - cursor: move;
  - }

1. Something to drag

- Anything can be dragged!
- Set the `draggable` attribute to true

```
<div id="dragMe" draggable="true"> This is draggable
</a>
```

2. Handle the dragstart event

```
$('[draggable]').each(function ()
   { this.addEventListener('dragstart',
      dragStartHandler, false);
});
```

- In this event you must signal what the thing being dragged can do

```
function dragStartHandler(e)
   { e.dataTransfer.effectAllowed = 'move';
}
```

3. Handle the dragover event

- Dropping is prevented by default, so you have to tell the target to stop stopping

```
function dragOverHandler(e){
e.preventDefault();
}
```

4. Handle the drop event

```
$('.dropzone').each(function () {
this.addEventListener('dragover',
     dragOverHandler, false);
  this.addEventListener('drop',
     dropHandler, false);
});
```

- The default drop events already usually countermand whatever we're trying to do, so cancel them

```
function dropHandler(e) {
  e.preventDefault();
}
```

Transmitting data

- You're transferring an item, from one place to another, so how do you get the data associated with it?
  - The dataTransfer object

Load it up when beginning the drag

```
dragStartEventHandler = function(e) {
e.dataTransfer.effectAllowed = 'move';
e.dataTransfer.setData('id', e.target.getAttribute("data-
   id"));
e.dataTransfer.setData('title',
$('.title', this).text());
}
```

Pull that same data out of it when dropping

```
dropEventHandler = function (e) {
var id = e.dataTransfer.getData('id');
var t = e.dataTransfer.getData('title');
$('#titleP').text(t);
var url = '/api/GetDetails/' + id;

$('#detailsDiv').load(url);
}
```

Drag and Drop in a nutshell

- When boiled down, it is easy to drag and drop
- Make things draggable with 'draggable="true"'
- Wire up the dragstart, dragover, and drop events
- Write and read data in the event.dataTransfer object with setData and getData

Questions?

References

- Crowther, R., & Lennon, J. (n.d.). File editing and management: rich formatting , file storage, drag and drop. In HTML5 in action.

- Geo API: Directories and System. (n.d.). Retrieved May 6, 2015, from http://www.w3.org/TR/geolocation-API/

- User Interaction HTML5. (n.d.). Retrieved May 6, 2015, from http://www.w3.org/html/wg/drafts/html/master/editing.html#dnd