



# CS 50 Walkthrough 8.1

## Problem Set 8: Mashup

Keito Uchiyama

# Topics

- JavaScript and Ajax
- Using the Google Maps API
- Parsing XML using PHP

# Steps

1. Import data from \$79.99 database
2. Embed a Google Map
3. Using the geocoder
4. Writing the server-side handler
5. Writing the client-side Ajax code
6. Write code to deal with XML response
7. Add Google News to the equation

# Step 1: Import data

- Writing command-line PHP scripts
  - Shebang (`#!/usr/bin/php`)
  - `$argv`, `$argc`
- Reading CSV files using PHP
  - What's a CSV file?
  - `fgetcsv()`

## Step 2: Embed a Google Map

- Get a Google Maps API key  
<http://code.google.com/apis/maps/>
- Copy the sample code

## Step 3: Using the Geocoder

- What's latitude and longitude?
- The Geocoder: Converts human-readable addresses to latitude/longitude
- Geocoder (GClientGeocoder) uses Ajax

## Step 4.1: Getting the bounds

- You can get the current bounds (i.e., what's visible right now) of the map by using **map.getBounds()**.
- That returns an object that has functions **getSouthWest()**, **getNorthEast()**, etc.
- Now we can tell PHP what portion of the map we're looking at!
- <http://code.google.com/apis/maps/documentation/reference.html>

## Step 4.2: Getting cities

- Now that PHP knows what part of the map we're looking at...
- Use our database to retrieve the 5 largest cities (i.e., zips and latlngs) within our bounds
- Some cities have multiple zip codes:  
**GROUP BY** and **SUM()**
- We want the 5 largest cities:  
**ORDER BY** and **LIMIT**



# Step 5: Writing the client-side JavaScript

- Registering events:
  - moveend, zoomend
  - GEvent.addListener()
- Our event handler:
  - Calls handler we wrote in Step 4
  - Sends bounds to server handler and gets zips and latlngs via Ajax. Same as in Supersection's example (but we can use GXmlHttp for simplicity in creating the XmlHttpResponse)

## Step 6.1: Deal with XML response

- Read the **XMLHttpRequest** (or **GXmlHttp**) object's **responseXML** property
- **responseXML** is a DOM-like tree representation of the XML that was returned by the server.

## Step 6.1: Deal with XML response

```
var cities =
    request.responseXML.getElementsByTagName("city");

// iterate over cities
for (var i = 0; i < cities.length; i++)
{
    // grab current city
    var city = cities[i];

    // debug
    alert(city.getAttribute("name") + "has zip code "
        + city.getAttribute("zip"));
}
```

## Step 6.2: Adding markers

- We can create a new GMarker
- It requires a GLatLng (the position)
- We can add a GMarker to a map using the `addOverlay()` function of the map

## Step 7.1: Adding Google News to the equation

- We can get an RSS feed of news items relevant to 02138: <http://news.google.com/news?geo=02138&output=rss>
- RSS feeds are written in XML
- You can parse XML in PHP using the SimpleXML functions (<http://www.php.net/SimpleXML>)
- We then mix the news items into the lat/longs we returned in Step 4.2

# Step 7.1: GInfoWindow

**RTFM** 😊

<http://code.google.com/apis/maps/documentation/reference.html#GInfoWindow>



# CS 50 Walkthrough 8.1

## Problem Set 8: Mashup

Keito Uchiyama