LESSON 3: DISTANCE CALCULATIONS + ASSIGNMENT

OUTLINE

- The first half of this lesson is going to be going over the distance/travel time services Google Maps provides for us
- The second half will be available to begin working on the assignment after we discuss it in class. This is so you can ask questions about prior material/ the assignment and get answers right away.

API KEYS: CLARIFICATION

- Some API keys are client API keys, and have to be stored in the browser (and hence public). This is the case with Google Maps. You can however protect the API key by limiting the domain/IP that is allowed to make requests with a particular key.
- Secret API keys should never be used from a client and need to be stored and kept secret on a server.

GEOMETRY LIBRARY

- Google maps lets us calculate the exact distance between two LatLng coordinates using their geometry library.
- This gets the true, geometric distance and does not take into consideration travel times, modes of transportation, directions etc.
- Useful for getting a manageable number of possible destinations

DISTANCE MATRIX

- Distance matrix is more complex and allows us to compute travel times/distances for common modes of transportation between a list of origins and destinations.
- Limited to 25 origins/destinations per request, and 100 total routes per request.
- Returns many rows for each possible route between origins/destinations
- Number of routes = # of origins * # of destinations

JSONP & CORS

- When we use fetch, our web app is (sometimes) requesting data from a server that is different than the one that sent the web page to the user.
- This USED to be considered a security risk, and wasn't allowed (except for script tags), which is troublesome for accessing JSON data.
- Three solutions for this exist: JSONP, CORS, and server-side calls.

JSONP

- JSON with Padding is a workaround and could be called the biggest “hack” of all time.
- JSONP would call a server, disguising the request as a script tag.
- The server would dynamically generate such a script, that when run, would simply return the data you actually wanted.
- Limited to GET requests, don’t handle errors well. Generally not used anymore.

CORS

- Cross Origin Resource Sharing allows a server to restrict the access of content to external domains, or make it completely public.
- This essentially means the server will can allow web apps served from a different domain to access content on it’s own domain.
- MUST be set by the server. Modern alternative to JSONP, much safer.

SERVER SIDE REQUEST

- Although JSONP and CORS are useful, they must be set by the server serving the data. If JSONP or CORS are not configured on the destination server, there is nothing your web app can do. You will be unable to access the content from within a JS application.
- In this case it is necessary to create a server that isn't limited by the same restrictions as a browser and can make the cross origin request.
- Then you can server your web app from that same server, or enable CORS to allow external request.

SERVER SETUP

- Setting up a server is not covered in this course, but I made code available for one that we will use in Assignment 1, as the Bikeshare Toronto endpoint is not setup with JSONP or CORS.