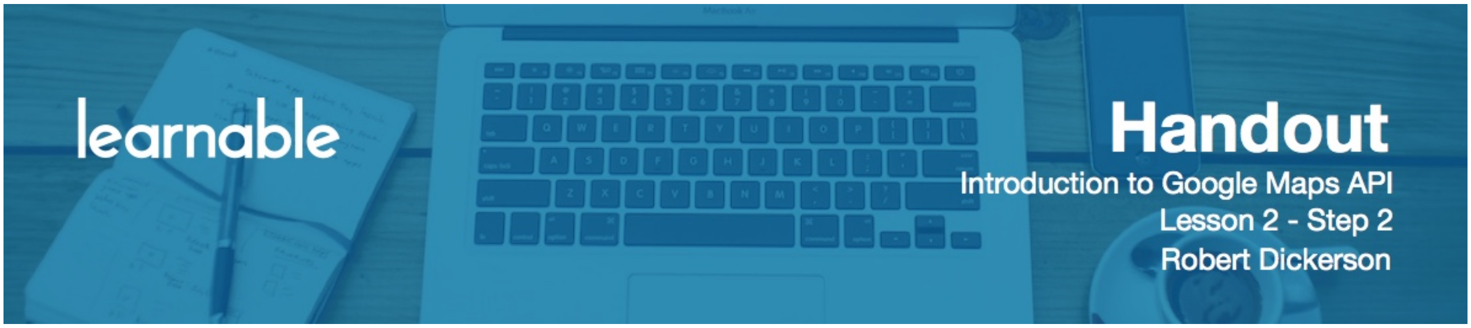

learnable

Handout

Introduction to Google Maps API

Lesson 2 - Step 2

Robert Dickerson

Introduction

At this step you will learn how to include Google Maps JavaScript API in your code, and learn about the parameters that allow you to set the version, your unique API key, and any optional API libraries.

Include Google API in code

To start using the Google Maps JavaScript API you will need to include it in your HTML file using a `script` tag.

```
<script type="text/javascript" src="https://maps.google.com/maps/api/js?v=3&key=API_KEY"></script>
```

The URL takes three parameters:

- Your API key, specified by the keyword `key`.
- The version, specified by `v`.
- The optional libraries you want to include, specified by the keyword `library`. If you are not using any of the optional libraries, then the `library` parameter does not need to be included.

None of these parameters are required, but the version and the API key are recommended before deploying to production.

When specifying the version number, you can provide a specific version, use the latest stable release, or use the experimental channel:

- If the version is **not specified**, then the experimental release would be the default version. The experimental version includes the latest features and enhancements, but might not be as stable and thoroughly tested.
- Using a **specific version** number allows you to test against a specific release and only update after you have performed regression testing. The downside here is that you will not receive automatic updates and features as Google promotes them to the stable release.
- When using the **stable version**, you will receive updates as Google promotes builds to the stable release channel. By using the stable release, you can take advantage of automatic updates for new features and bug fixes as they get updated.

We will be using the stable version in the examples in this course.

There are a couple useful libraries than can be included when retrieving the JavaScript API file. We will not be using these libraries throughout the course but I wanted to give you a brief overview of them.

You can include single or multiple libraries by using the optional `library` URL parameter when retrieving the API. Including multiple libraries is accomplished by using a comma-separated list of the libraries' names.

- The **Adsense** library allows you to monetize your application by displaying advertisements on the map. Ads will be linked to your Adsense account, and the ad content will be served based on the areas your users are viewing on the map. Using this library allows for an additional revenue stream for your site.
- The **Drawing** library provides a set of tools for allowing users to interactively place markers and shapes, including lines, circles, polygons, and squares. Some examples of this library in use include allowing users to place markers indicating points of interest, performing searches based on shape bounds, or drawing routes towards their destination on the map.
- The **Geometry** library provides a set of methods for computer distances, areas, and angles from latitudes and longitudes. These methods are useful if you want to display distance measurements between points or compute the area of a polygon. The library also provides encoding methods for compressing and decompressing polyline paths using the encoded polyline algorithm. The encoded polylines store the latitude and longitude coordinates and the maximum zoom levels to display the points. These methods are useful for efficiently storing and transmitting large sets of data.
- The **Places** library enables you to search for establishments, locations, or prominent points of interest in the Google database. You can search for places based on the defined area, within the current bounds of the map view, or relative to a fixed point. The library search functions will return an array of places with details about each place. This library is useful for integrating searches for places, such as restaurants and businesses, onto your

map.

- The **Visualization** library provides classes for turning your data into visualizations, such as a heat map. This library is useful if you want to take point data and use a heat map to display weighted intensity, instead of just markers on the map.