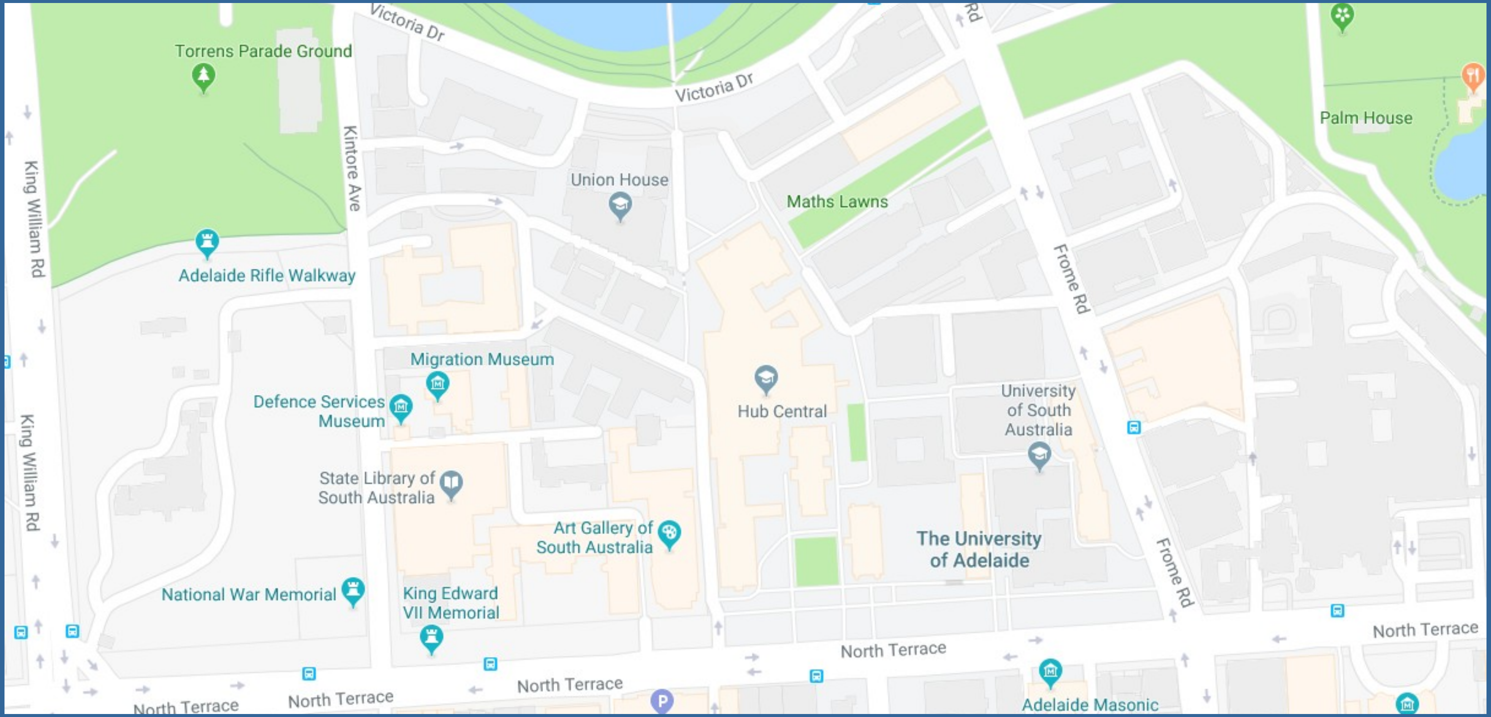


School of Computer Science

COMP SCI 2207/7207 Web and Database Computing
Lecture 20 – Client Side APIs (Google Maps)

adelaide.edu.au

seek LIGHT

Client Side APIs

- Provide methods to access and integrate services external services into our site.
- Are run on the client –
  - Saves our server resources
- Can be accessed different ways
  - REST vs Library

You can use any Map API

with your project

But today we're using...

Google Maps API

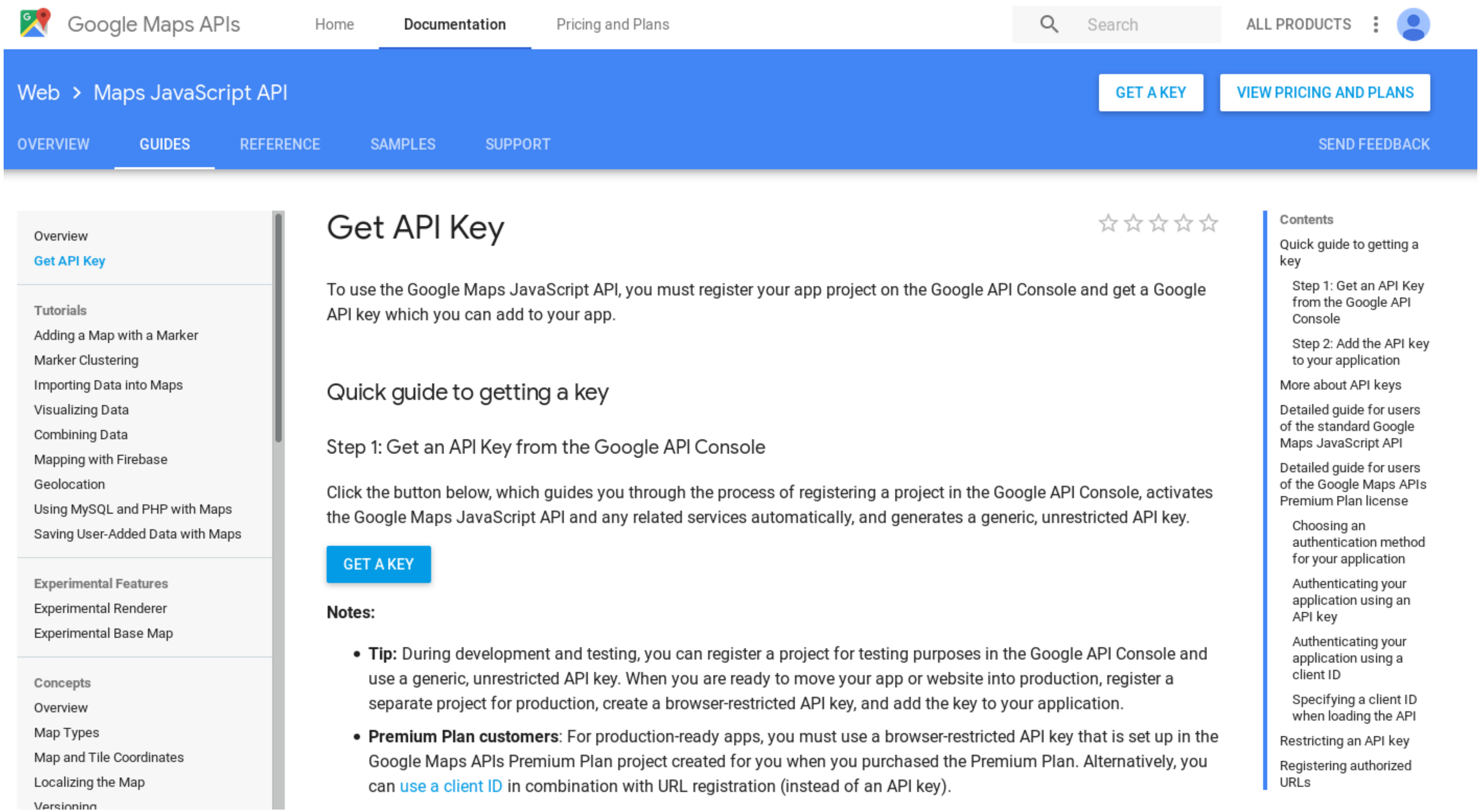
- Like jQuery or Vue.js, Google Maps is a library of pre-written javascript objects that you can use to create and manipulate maps in your application
- Also like jQuery, we can get this library from Google

… so how do we use it?

1. Get an API key

Google Maps API Key

- Google protects their API from misuse by requiring developers to obtain a unique code called an API Key
- Keys can be used to restrict access and need to be registered to the site they're used on.



Get API Key

To use the Google Maps JavaScript API, you must register your app project on the Google API Console and get a Google API key which you can add to your app.

Quick guide to getting a key

Step 1: Get an API Key from the Google API Console

Click the button below, which guides you through the process of registering a project in the Google API Console, activates the Google Maps JavaScript API and any related services automatically, and generates a generic, unrestricted API key.

GET A KEY

Notes:

- **Tip:** During development and testing, you can register a project for testing purposes in the Google API Console and use a generic, unrestricted API key. When you are ready to move your app or website into production, register a separate project for production, create a browser-restricted API key, and add the key to your application.
- **Premium Plan customers**: For production-ready apps, you must use a browser-restricted API key that is set up in the Google Maps APIs Premium Plan project created for you when you purchased the Premium Plan. Alternatively, you can use a client ID in combination with URL registration (instead of an API key).

Contents

https://developers.google.com/maps/documentation/javascript/get-api-key

2. Load maps script

Google Maps Script

- Use a script tag at the end of your page's body

```
<script async defer
    src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=myMapFunction">
</script>
```

- Need to ensure it loads in correct order.

3. Setup Placeholder

Google Maps Placement

- Create a <div> tag
  - This will act as a container for the map.

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

- Set its size in your stylesheet

```
#map {
    width: 100%;
    height: 200px;
}
```

No, really

Set it's size

- Especially it's height
- Divs have default height of 0!
- The map will fill this container.

4. Initialise the Map

Initialise the Map in your code

- In your javascript file, create a function that matches the function in the maps URI

```
<script async defer
    src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&callback=myMapFunction">
</script>


// Your separate JS file

function myMapFunction() {
    // Map stuff
}
```

Initialise the Map in your code

- The map creation and changes are all done through javascript.
- When we create a map we pass two values: the options and the element that the map should be placed in (ie the div).
- Options we can set:
  - **center** - sets the location that is the center of the map (based on latitude, longitude)
  - **zoom** - 0 is max zoom out, higher numbers zoom in more
  - **mapTypeId** one of HYBRID ROADMAP SATTELITE TERRAIN
  - Options to turn off scrolling, remove controls, etc
  - center and zoom are required
- For a list of all the options, see https://developers.google.com/maps/documentation/javascript/reference#MapOptions

Initialise the Map in your code

```
// Your separate JS file
var map = null;

function myMapFunction() {
    // Set options
    var mapOptions = {
        center: { lat: -34.9285, lng: 138.6007 },
        zoom: 12
    };

    // Load map
    map = new google.maps.Map(
        document.getElementById("map"),
        mapOptions
    );
}
```

- The map creation and changes are all done through javascript.
- How can we work out the latitude and longitude of a location?
- The map constructor requires the options that we just created and the div that will contain it.

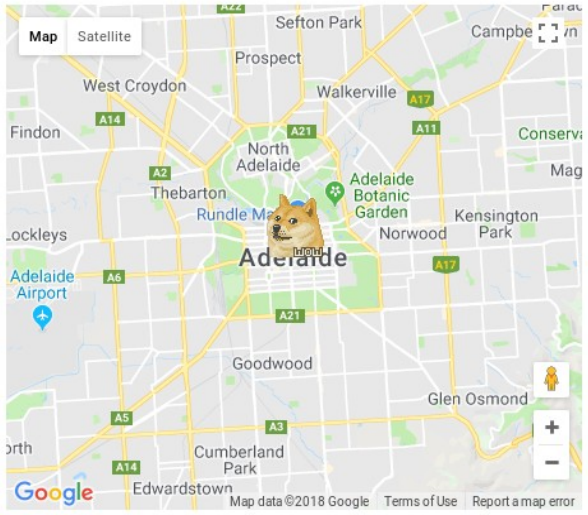
That's a Map

My Google Maps Demo

Map
Satellite
West Lakes
Woodville
Sefton Park
Paradise
Campbelltown
Black Hill
Prospect
West Croydon
Walkerville
Montacute
Grange
Findon
North Adelaide
Morialta Conservation Park
Magill
Fulham Gardens
Thebarton
Adelaide Botanic Garden
Rundle Mall
Henley Beach
Lockleys
Norwood
Kensington Park
Adelaide
Adelaide Airport
West Beach
Ashton
Goodwood
Summertown
Glen Osmond
Cleland Conservation Park
Glenelg North
Cumberland Park
Edwardstown
Glenelg
Map data ©2018 Google
Terms of Use
Report a map error

Adding a Marker…

```
var place = { lat: -34.9285, lng: 138.6007 };
var marker = new google.maps.Marker({
                    position: place,
                    map: map
                 });
```



- Use google.maps.Marker constructor

- What else can we do with markers?

https://developers.google.com/maps/documentation/javascript/reference/3.exp/marker https://developers.google.com/maps/documentation/javascript/

THE UNIVERSITY of ADELAIDE

What's happening?

- Prac Exercise 5 due today
- Prac Exercise 6 due Monday
- Prac Exercise 7 available tonight, due Mon week 9
  - Websub available Monday

The rest of this lecture will be answering your questions posted on the Survey, plus a second attempt at Prac 3 using Vue.js