

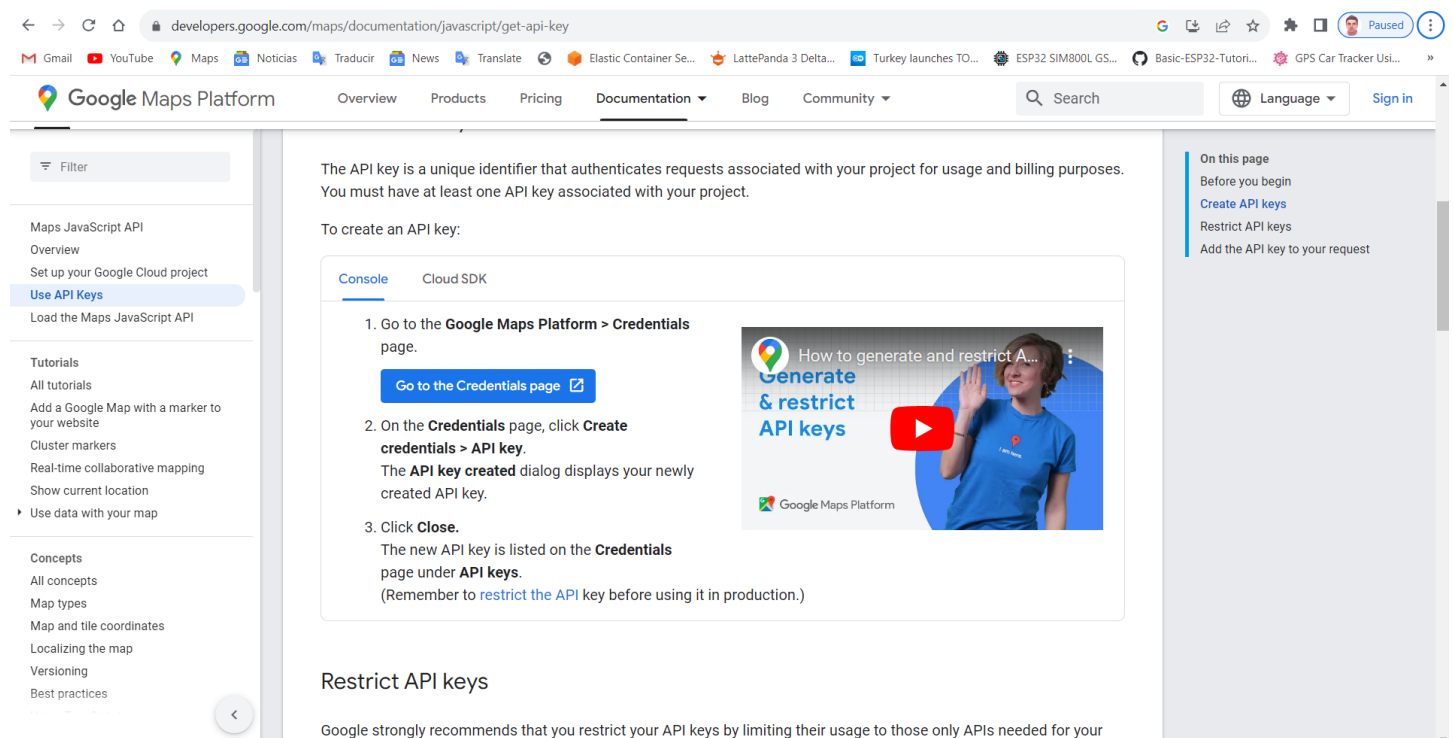
Angular with GoogleMaps integration.

https://github.com/luiscoco/AngularWithGooglemaps_Sample1-SimpleMap

How to create the Google Maps API Key credentials

Navigate to this link : <https://developers.google.com/maps/documentation/javascript/get-api-key>

Then press the button "Go to the credentials page":



The screenshot shows the Google Maps Platform documentation page for getting an API key. The page is titled "The API key is a unique identifier that authenticates requests associated with your project for usage and billing purposes. You must have at least one API key associated with your project." It includes a section "To create an API key:" with three steps: 1. Go to the Google Maps Platform > Credentials page. 2. On the Credentials page, click Create credentials > API key. 3. Click Close. A video thumbnail titled "How to generate and restrict A... Generate & restrict API keys" is also visible. The left sidebar shows the navigation menu with "Use API Keys" selected. The right sidebar shows a table of contents with links to "Before you begin", "Create API keys", "Restrict API keys", and "Add the API key to your request".

The API key is a unique identifier that authenticates requests associated with your project for usage and billing purposes. You must have at least one API key associated with your project.

To create an API key:

1. Go to the **Google Maps Platform > Credentials** page.
2. On the **Credentials** page, click **Create credentials > API key**. The **API key created** dialog displays your newly created API key.
3. Click **Close**. The new API key is listed on the **Credentials** page under **API keys**. (Remember to **restrict the API** key before using it in production.)

Restrict API keys

Google strongly recommends that you restrict your API keys by limiting their usage to those only APIs needed for your

Then go to the "Credentials" option in this web page and create a new API Key:

Google Cloud console showing the Google Maps Platform Credentials page. The page displays a warning to restrict API keys, a reminder to configure the OAuth consent screen, and sections for API Keys, OAuth 2.0 Client IDs, and Service Accounts. The 'API Keys' section shows a single key named 'Clave de API 1' created on June 19, 2023, with 'APPLY RECOMMENDED RESTRICTIONS' button. The 'Service Accounts' section shows the 'App Engine default service account'.

Select this option "CREATE CREDENTIALS" -> "API KEY"

Google Cloud console showing the Google Maps Platform Credentials page. A dropdown menu is open, showing options: API key, OAuth client ID, Service account, and Help me choose. The 'API key' option is selected.

Google Maps JavaScript/TypeScript API

<https://developers.google.com/maps/documentation/javascript/overview>

Here's an Angular application that integrates the provided TypeScript code to show a simple Google Maps:

1.Create a new Angular blank application.

```
ng new googlemaps_simplemap
```

2.Install the required dependencies:

```
npm install @types/googlemaps
```

3.Navigate inside the project folder

```
cd project-folder
```

and create a new Angular component. Let's call it MapComponent:

```
ng generate component Map
```

4.Open the generated map.component.ts file and replace the content with the following code:

```
import { Component, OnInit } from '@angular/core';

declare const google: any;

@Component({
  selector: 'app-map',
  template: `
    <div id="map" style="width: 100%; height: 400px;"></div>
  `,
  styles: [`
    #map {
      height: 100%;
    }
  `]
})
export class MapComponent implements OnInit {
  ngOnInit() {
    this.initMap();
  }

  async initMap(): Promise<void> {
    const map = new google.maps.Map(document.getElementById("map"), {
      center: { lat: -34.397, lng: 150.644 },
      zoom: 8,
    });
  }
}
```

5.Open the app.module.ts file and import the MapComponent:

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';

import { AppComponent } from './app.component';
import { MapComponent } from './map/map.component';

@NgModule({
  declarations: [
    AppComponent,
    MapComponent
  ],
  imports: [
    BrowserModule
  ],
  providers: [],
  bootstrap: [AppComponent]
})
export class AppModule { }
```

6. Open the app.component.html file and add the <app-map> </app-map> tag:

```
<h1>My Angular App</h1>
<app-map></app-map>
```

7. Open the tsconfig.app.json file in the root directory of your Angular project. Add "node_modules/@types" to the "types" array. Your tsconfig.app.json file should look like this:

```
{
  "extends": "./tsconfig.json",
  "compilerOptions": {
    "outDir": "./out-tsc/app",
    "types": [
      "node",
      "googlemaps"
    ]
  },
  "files": [
    "src/main.ts",
    "src/polyfills.ts"
  ],
  "include": [
    "src/**/*.d.ts"
  ]
}
```

8. In the root directory of your Angular project, open the angular.json file. Inside the "build" target, ensure that the "options" property has a "polyfills" property with the value set to "src/polyfills.ts". It

should look like this:

```
"options": {
  "polyfills": "src/polyfills.ts",
  ...
},
```

9. Now, create a new file called polyfills.ts in the src folder of your project. Add the following content to the polyfills.ts file:

```
import 'zone.js/dist/zone'; // Included by default in Angular CLI projects
```

10. Confirm that you have added the Google Maps API script correctly in the index.html file. It should be placed just before the closing tag, like this:

```
<script src="https://maps.googleapis.com/maps/api/js?key=YOUR_API_KEY&libraries=places"></script>
```

VERY IMPORTANT! Do not forget to set the "Google Maps API KEY" in the above line.

11. Start the application with the command:

```
ng serve -o
```

This is the application output

