# **Google-Map-Auto-Complete-Search**

<https://github.com/ramsatt/Ionic-3---Google-Map-Auto-Complete-Search>

Docs on AGM(angular google map)

Google place auto complete

In terminal

1. Install Angular Google Map - npm install @agm/core --save
2. Install Google Maps - npm install @types/googlemaps --save-dev

3.in app.module.ts

...

import { AgmCoreModule } from '@agm/core';

…

….

imports: [

BrowserModule,

HttpClientModule,

IonicModule.forRoot(MyApp),

AgmCoreModule.forRoot({

apiKey: "AIzaSyC-uFyteAemje7\_NQ2zSDQItTstIVBrSgw",

libraries: ["places"]

})

],

…..

4 in my.html

...

<ion-content padding>

...

<ion-item>

<ion-label floating>Enter Address</ion-label>

<ion-input id="txtHome" type="text"></ion-input>

</ion-item>

<agm-map [latitude]="latitude" [longitude]="longitude" [scrollwheel]="false" [zoom]="zoom">

<agm-marker [latitude]="latitude" [longitude]="longitude"></agm-marker>

</agm-map>

</ion-content>

…

5. My.ts

import { Component, NgZone, ElementRef, OnInit, ViewChild } from '@angular/core';

import { NavController, ModalController } from 'ionic-angular';

import {FormControl} from "@angular/forms";

import { } from 'googlemaps';

import { MapsAPILoader } from '@agm/core';

@Component({

selector: 'page-home',

templateUrl: 'home.html'

})

export class HomePage {

public latitude: number;

public longitude: number;

public searchControl: FormControl;

public zoom: number;

count;

country;

city;

state;

txtHome;

@ViewChild("search")

public searchElementRef;

constructor(public navCtrl: NavController, private mapsAPILoader: MapsAPILoader,

private ngZone: NgZone) {

}

ionViewDidLoad() {

//set google maps defaults

this.zoom = 4;

this.latitude = 22.5726;

this.longitude = -88.3639;

//create search FormControl

this.searchControl = new FormControl();

//set current position

this.setCurrentPosition();

//load Places Autocomplete

this.mapsAPILoader.load().then(() => {

let nativeHomeInputBox = document.getElementById('txtHome').getElementsByTagName('input')[0];

var options = {

types: ['geocode'],

//types: ["address"]

//types: ['(cities)'],

//componentRestrictions: {country: "us"}

};

let autocomplete = new google.maps.places.Autocomplete(nativeHomeInputBox,options);

autocomplete.addListener("place\_changed", () => {

this.ngZone.run(() => {

//get the place result

let place: google.maps.places.PlaceResult = autocomplete.getPlace();

//verify result

if (place.geometry === undefined || place.geometry === null) {

return;

}

//set latitude, longitude and zoom

this.latitude = place.geometry.location.lat();

this.longitude = place.geometry.location.lng();

this.zoom = 12;

var address = place.formatted\_address;

var value = address.split(",");

this.google\_place = value[0];

console.log("google\_place",this.google\_place);

this.count=value.length;

this.country=value[this.count-1];

this.state=value[this.count-2];

this.city=value[this.count-3];

console.log(this.latitude,this.longitude);

console.log("getcity",this.city);

});

});

});

}

private setCurrentPosition() {

if ("geolocation" in navigator) {

navigator.geolocation.getCurrentPosition((position) => {

this.latitude = position.coords.latitude;

this.longitude = position.coords.longitude;

this.zoom = 12;

});

}

}

}

6.my.scss

…

agm-map {

height: 500px;

}

…

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/////////////////////\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*