# **Geolocation Tracker with Google Map and Track Drawing**

https://devdactic.com/ionic-location-tracker-map-track/

|  |  |
| --- | --- |
| 1  2  3  4  5 | ionic start devdacticLocationTracker blank  cd devdacticLocationTracker  ionic cordova plugin add cordova-plugin-geolocation --variable GEOLOCATION\_USAGE\_DESCRIPTION="To track your walks"  ionic cordova plugin add cordova-sqlite-storage  6. npm install --save @ionic-native/geolocation **src/index.html right before the cordova import:** |

<script src="<http://maps.google.com/maps/api/js?key=YOUR_API_KEY_HERE>"></script>

**app/app.module.ts**:

import { Geolocation } from '@ionic-native/geolocation';

import { IonicStorageModule } from '@ionic/storage';

imports: [

..

IonicStorageModule.forRoot(),

..

],

providers: [

..

Geolocation

]

### Additional config for iOS plist(config.xml):

|  |  |
| --- | --- |
|  | <edit-config file="\*-Info.plist" mode="merge" target="NSLocationWhenInUseUsageDescription">  <string>This App wants to track your location</string>  </edit-config> |

|  |
| --- |
| <ion-header>  <ion-navbar color="primary">  <ion-title>  GPS Tracking  </ion-title>  </ion-navbar>  </ion-header>    <ion-content padding>    <button ion-button full icon-left (click)="startTracking()" \*ngIf="!isTracking">  <ion-icon name="locate"></ion-icon>  Start Tracking  </button>  <button ion-button full color="danger" icon-left (click)="stopTracking()" \*ngIf="isTracking">  <ion-icon name="hand"></ion-icon>  Stop Tracking  </button>    <div #map id="map"></div>    <ion-list>  <ion-list-header>Previous Tracks</ion-list-header>  <ion-item \*ngFor="let route of previousTracks">  {{ route.finished | date }}, {{ route.path.length }} Waypoints  <button ion-button clear item-end (click)="showHistoryRoute(route.path)">View Route</button>  </ion-item>  </ion-list>  </ion-content> |

**pages/home/home.scss**:

|  |  |
| --- | --- |
|  | page-home {  #map {  width: 100%;  height: 300px;  }  } |

**pages/home/home.ts**:

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

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

import { Geolocation } from '@ionic-native/geolocation';

import { Subscription } from 'rxjs/Subscription';

import { filter } from 'rxjs/operators';

import { Storage } from '@ionic/storage';

declare var google;

@Component({

selector: 'page-home',

templateUrl: 'home.html'

})

export class HomePage {

@ViewChild('map') mapElement: ElementRef;

map: any;

currentMapTrack = null;

isTracking = false;

trackedRoute = [];

previousTracks = [];

positionSubscription: Subscription;

constructor(public navCtrl: NavController, private plt: Platform, private geolocation: Geolocation, private storage: Storage) { }

ionViewDidLoad() {

this.plt.ready().then(() => {

this.loadHistoricRoutes();

let mapOptions = {

zoom: 13,

mapTypeId: google.maps.MapTypeId.ROADMAP,

mapTypeControl: false,

streetViewControl: false,

fullscreenControl: false

}

this.map = new google.maps.Map(this.mapElement.nativeElement, mapOptions);

this.geolocation.getCurrentPosition().then(pos => {

let latLng = new google.maps.LatLng(pos.coords.latitude, pos.coords.longitude);

this.map.setCenter(latLng);

this.map.setZoom(16);

}).catch((error) => {

console.log('Error getting location', error);

});

});

}

loadHistoricRoutes() {

this.storage.get('routes').then(data => {

if (data) {

this.previousTracks = data;

}

});

}

|  |  |
| --- | --- |
|  | startTracking() {  this.isTracking = true;  this.trackedRoute = [];    this.positionSubscription = this.geolocation.watchPosition()  .pipe(  filter((p) => p.coords !== undefined) //Filter Out Errors  )  .subscribe(data => {  setTimeout(() => {  this.trackedRoute.push({ lat: data.coords.latitude, lng: data.coords.longitude });  this.redrawPath(this.trackedRoute);  }, 0);  });    }    redrawPath(path) {  if (this.currentMapTrack) {  this.currentMapTrack.setMap(null);  }    if (path.length > 1) {  this.currentMapTrack = new google.maps.Polyline({  path: path,  geodesic: true,  strokeColor: '#ff00ff',  strokeOpacity: 1.0,  strokeWeight: 3  });  this.currentMapTrack.setMap(this.map);  }  } |

|  |  |
| --- | --- |
|  | stopTracking() {  let newRoute = { finished: new Date().getTime(), path: this.trackedRoute };  this.previousTracks.push(newRoute);  this.storage.set('routes', this.previousTracks);    this.isTracking = false;  this.positionSubscription.unsubscribe();  this.currentMapTrack.setMap(null);  }    showHistoryRoute(route) {  this.redrawPath(route);  } |

}