

Ionic 2

Google Map

HW 7: home.html

```
<ion-slides #weatherslides [options]='slideoptions'>
  <ion-slide *ngIf="localWeather">
    <button ion-item> <ion-thumbnail item-left>... </ion-thumbnail> <h2>...</h2><h3>...</h3><p>...</p> </button>
    <ion-list no-lines>
      <ion-item height=60 ion-item *ngFor="let forecast of localForecast">
        <ion-avatar item-left>...</ion-avatar> <h3>...</h3> </ion-item></ion-list>
    </ion-slide>
  <ion-slide *ngFor="let weather of weatherList">
    <button ion-item> <ion-thumbnail item-left>... </ion-thumbnail> <h2>...</h2><h3>...</h3><p>...</p> </button>
    <ion-list no-lines [hidden]="!forecastList[weatherList.indexOf(weather)]">
      <ion-item height=60 *ngFor="let forecast of forecastList[weatherList.indexOf(weather)]">
        <ion-avatar item-left>...</ion-avatar> <h3>...</h3> </ion-item> </ion-list>
    <button ion-button block color="danger" (click)="removeWeather(weather)">... </button>
  </ion-slide>
</ion-slides>
```

Two ways for Google Map

- Option 1: Javascript API
 - Call google map api directly to google map site
 - Works on web browsers
- Option 2: Native Google Map SDK API
 - Use Cordova plugin for google map API
 - Which will call underlying SDK API provided by iOS or Android
 - Which will make google API calls
 - Need mobile platform
- Performance: SDK > JS
- Reliability: JS > SDK

Google Map JS API Quickstart

```
<script src="https://maps.googleapis.com/maps/api/js?
callback=initmap" async defer></script>
<script>
    function initmap() {
        new google.maps.Map(document.getElementById("map"),
            { zoom:10,                      // mapOptions
              center: {lat:37.2, lng: 127},
              mapTypeId: google.maps.MapTypeId.ROADMAP } );
    }
</script>
<div id="map" style="height: 354px; width:713px;"></div>
```

Understanding Google Map JS URL

- Requirement:
 - JS & Page is loaded before initmap
- Facts
 - window.load=initmap: calls initmap after page load
 - callback=initmap: calls initmap after JS load
- Case 1: JS → page → initmap
 - Option 1: <script> at top with window.load
 - Bad: <script> at top with callback
- Case 2: page → JS → initmap
 - Option 1: <script> at bottom
 - Option 2: <script...defer> at top
- Case 3: page & JS (simultaneously) → initmap
 - <script... **async defer**> at top
 - Very Fast !!

Versioning & API Key

- Which version to use?
 - Release version: stable, v=3 (now 3.25)
 - Experiment version: latest, v=3.exp (now 3.26)
 - Frozen version: previous release, v=3.24
- Standard: No version specified → experiment
- Premium: No version specified → release
- From version 3, API key is not required
 - Without key, considered standard
- Why API Key?
 - Services: usage statistics, free 25K/24h

Get Google Map JS API Key

- Go to <https://code.google.com/apis/console>
- Log in using google account
- Choose Google Map Javascript API
- Create a project
- Enable Google Map Javascript API for the project
- Go to Credentials
- Click “What credentials so I need” button
- Get the API Key

Type Definitions

- What is type definition?
 - Javascript libraries don't have type information
 - To use JS libraries with typescripts, we need type information for variables in JS libraries
- Install type definitions for Google Maps JS API
 - `npm install -g typings`
 - `typings search maps`
 - `typings install dt~google.maps --global --save`

Ionic 2

- home.html
 - `<div #map id="map"></div>`
- home.scss

```
#map {
  width: 100%;
  height: 100%;
}
```
- home.ts

```
ionViewDidEnter(){
  new google.maps.Map(document.getElementById( 'map'),
    { zoom:10,
      center: {lat:37.2, lng: 127},
      mapTypeId: google.maps.MapTypeId.ROADMAP } );
}
```

Ionic 2 with Map Component

- ionic g component map

- map.ts:

```
@Component({
  selector: 'map',
  template: '<div id="map"></div>'
})
export class Map {
  ...
  ngOnInit() {
    new google.maps.Map(document.getElementById( 'map'),
      { zoom:10, center: {lat:37.2, lng: 127},
        mapTypeId: google.maps.MapTypeId.ROADMAP } );
  }
}
```

- map.css

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