

Ionic 2

HW6

Reading GPS

- Install Cordova GPS plugin
 - ionic plugin add cordova-plugin-geolocation
- import
 - import {Geolocation} from 'ionic-native';
- get GPS data
 - Geolocation.getCurrentPosition().then (
 (resp) => {
 console.log(resp.coords.latitude);
 console.log(resp.coords.longitude);
 });

Get local weather

- Show the weather of current location
 - Get GPS data
 - Make a weather query based on the location
 - Separate this local weather from added weathers
- Show forecast of current location when clicked
 - When clicked, make a forecast query using the city name

Local Storage

- Local storage allows to store (key, value) in the browser storage
- Persistent, i.e., stays after reload/app restart
- Storage is a provider: import & add in appmodule.ts, inject to constructor
- Limitation
 - 10MB

Local Storage

- Set up

```
import {Storage} from '@ionic/storage';
constructor( ..., public storage: Storage);
```
- Insert (key, value)
 - `storage.set( key, value );`
- Retrieve value for key
 - `storage.get( key ).then( value => ... )`
- Get an array of keys
 - `storage.keys().then( keys => ... )`
- Iterate through the keys
 - `storage.forEach().then( (value, key) => ... )`
- Clear the local storage
 - `storage.clear()`

How to store a JSON object?

- to store
`JSON.stringify( var )`
- when load
`JSON.parse( var )`

SQLite

- Access the SQLite DB provided by the mobile platform
- Need Cordova plug-in support
 - cordova plugin add cordova-sqlite-storage –save

- Code

```
import { SQLite } from 'ionic-native';
let db = new SQLite();
db.openDatabase( {name: 'mydb.db', location: 'default'} ).then(
    () => {
        db.executeSql( <SQL Statement>, {} ).then(
            () => {...},
            (err) => { ... } );
    },
    (err) => {...} );
```

- Check <https://www.thepolyglotdeveloper.com/2015/12/use-sqlite-in-ionic-2-instead-of-local-storage>

Persistent weather

- When App restarts/reloads, all added weathers are gone
- Store added weathers in a local storage
- Show them at start
- Idea
 - When the list updates, store weatherList object
 - At start, load stored object to weatherList

Slide to delete

```
<ion-item-sliding *ngFor="let item..." >  
  <button ion-item ...>  
    ...  
  </button>  
  <ion-item-options>  
    <button ion-button icon-only (click)='remove(item)'>  
      ...  
    </button>  
  </ion-item-options>  
</ion-item-sliding>
```

Slice to delete a weather

- Remove a weather from weatherList
 - get index of the weather
 - weatherList.indexOf(weather)
 - splice!
 - weatherList.splice(<index>, <length>)
 - remove from <index> to <index>+<length>-1

Pipe

- Used for frequent data manipulation before displaying to the user
 - “{{weather.dt*1000 | date: 'EEEE'}}” in forecast.html
- Why pipe?
 - Unix: cat myfile.txt | grep apple | wc -l
 - Ex: convert Minho Shin → Minho → MINHO
 - “Minho Shin” | firstName | toUppercase

Generate a pipe

- ionic g pipe myPipe
- Generated my-pipe.ts

```
import { Injectable, Pipe } from '@angular/core';
@Pipe( {name: 'mypipe' } )
@Injectable() export class MyPipe {
  transform(value, args) {
    value = value + "; // make sure it's a string
    return value.toUpperCase(); }}
```
- Register to app.module.ts: declarations section (also import)
- In the ts file using this pipe
 - import pipe
- Use in html
 - {{ name | mypipe }}

Pipe with arguments

- {{ name | case:"up" }}, {{name | case:"down"}}
- transform(value, option:string) {
 if(option == "up")
 ...
 else
 ...
}
- {{ name | change:"first":"down"]}}
 {{ name | change:"last":"up] }}
- transform(value, opt1: string, opt2: string)
- ...

Publish to Android Play Store

- Install Android Studio
- In SDK manager (android), install
 - Android SDK Platform for your targeted version of Android
 - Android SDK Platform-Tools
 - Android SDK build-tools **version 23.0.3** (not 24)
 - Android Support Repository (found under “Extras”)
- Set path
 - Mac: <http://ionicframework.com/docs/v2/resources/platform-setup/mac-setup.html>
 - Win: <http://ionicframework.com/docs/v2/resources/platform-setup/windows-setup.html>
- Create an APK
 - cordova build –release android
→ platforms/android/build/outputs/apk/android-release-unsigned.apk
- Generate private/public keys
 - keytool -genkey -v -keystore mykey.keystore -alias mykey -keyalg RSA -keysize 2048 -validity 10000
- Sign the APK
 - jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore mykey.keystore <apk_file> mykey
- Check <https://ionicframework.com/docs/guide/publishing.html>