

Ionic 2

Modals



- Out on Sept 28, 2016
- Update to Angular 2 Final
- Ahead-of-Time (AoT) compiling: Faster
- New build process: Rollup, slimmer, no-Gulp
- New app structure
 - src: every source code
 - src/app: root app component
 - src/pages, src/providers, src/asset, src/theme
- Some API changes, but NO MORE!
- Announcement: <http://blog.ionic.io/announcing-the-ionic-2-release-candidate/>
- Change log: <https://github.com/driftyco/ionic/blob/master/CHANGELOG.md>
- Steps to upgrade to RC0: see change log
- **Install ionic RC0**
 - update nodejs higher than 6
 - npm uninstall -g ionic
 - npm install -g ionic

Migrate existing project to RC0

- `ionic start myapp --v2`
- copy `app/pages/*` to `src/pages/`
- copy `app/providers/*` to `src/providers/`
- copy `app/pipes/*` to `src/pipes/`
- modify `templateUrl` relative to the TS file
- Put to `src/app/app.module.ts` in sections
 - declarations: `pages, pipes, components`
 - `entryComponents: pages`
 - `providers: providers`
- Remove `pipes/providers` array from `@Component`
- `<button>` → `<button ion-button>`

Tips

- ionic serve options
 - b: no browser launched
 - c: console output to command console
 - d: no live reload
 - l: lab
- Running chrome without decorations
 - WIN: `chrome.exe --app=http://localhost:8100`
 - MAC: `/Applications/Google Chrome.app/Contents/MacOS/Google Chrome --app=http://localhost:8100`

HW5

Adding Pull-to-refresh feature

```
<ion-content padding>
```

```
  <ion-refresher (ionRefresh)="refreshHandler($event)">
```

```
    <ion-refresher-content>
```

```
    </ion-refresher-content>
```

```
  </ion-refresher>
```

```
  ...
```

```
</ion-content>
```

```
refreshHandler(refresher) {
```

```
  ....
```

```
  refresher.complete();  // stop spinning wheel, push back the list
```

```
}
```

Adding Modal

- add trigger button in HomePage
- add a page myModal
 - ionic g page myModal
- add MyModal class to app.module.ts in
 - declarations & entryComponents
- in home.ts
 - import/inject ModalController as modalCtrl
 - on click
 - let mymodal = this.modalCtrl.create(MyModal);
 - mymodal.present();
- in mymodal.html/ts
 - add cancel button with icon “close-circle”
 - import/inject ViewController as view
 - on dismiss, call this.view.dismiss();

Adding Content to Modal

- typed input
 - <ion-label stacked>City</ion-label>
 - <ion-input type="text"></ion-text>
- Selection
 - <ion-select>
 - <ion-option value="kr">Korea</ion-option>
 - <ion-option value="us">United States</ion-option>
 - </ion-select>
- Button
 - <button ion-button block (click)="...">...</button>

Passing Data (Input) Back

- Modal can send data as a parameter
 - `this.view.dismiss(<param>)`
- Parent page can receive data by `onDismiss`
`let mymodal = this.modalCtrl.create(MyModal);`
`mymodal.onDismiss( data => {...} );`
`mymodal.present();`

Openweathermap.org API

- openweathermap.org
- sign up for app id
- [http://api.openweathermap.org/data/2.5/
weather?appid=<appid>&q=Seoul,KR](http://api.openweathermap.org/data/2.5/weather?appid=<appid>&q=Seoul,KR)

Make your own provider

- ionic g provider myProvider
- add it to app.module.ts
- import
 - import { Injectable } from '@angular/core';
 - import { Http, Response } from '@angular/http';
 - import {Observable} from 'rxjs/Observable';
 - import 'rxjs/add/operator/map';
- getWeatherByCity()

Weather List

- ```
<ion-list>
 <button ion-item *ngFor='let weather of weatherList' >
 <ion-thumbnail item-left>
 <img [src] = 'http://openweathermap.org/img/w/
 {{weather.weather[0].icon}}.png' height=50>
 </ion-thumbnail>
 <h2>{{weather.main.tmp}}</h2>
 <h3>{{weather.name}}</h3>
 <p> Low: {{weather.main.temp_min}} High:
 {{weather.main.temp_max}}</p>
 </button>
 </ion-list>
```