Intro to Ionic

TA Seolha Lee

Today’s goals

- Overview of Ionic and Setting Up
- Ionic Components
  - Structural, Item, Inputs, Lists
- Services
- Routing
- Create a simple project
  - Showcase components
  - Showcase navigation

Setting Up Ionic

- Install Ionic

```
npm info ionic              // Check current versions of ionic
npm install -g @ionic/cli
ionic -v                    // Check ionic version
```

- Create Ionic project

```
ionic start [project_name]
```

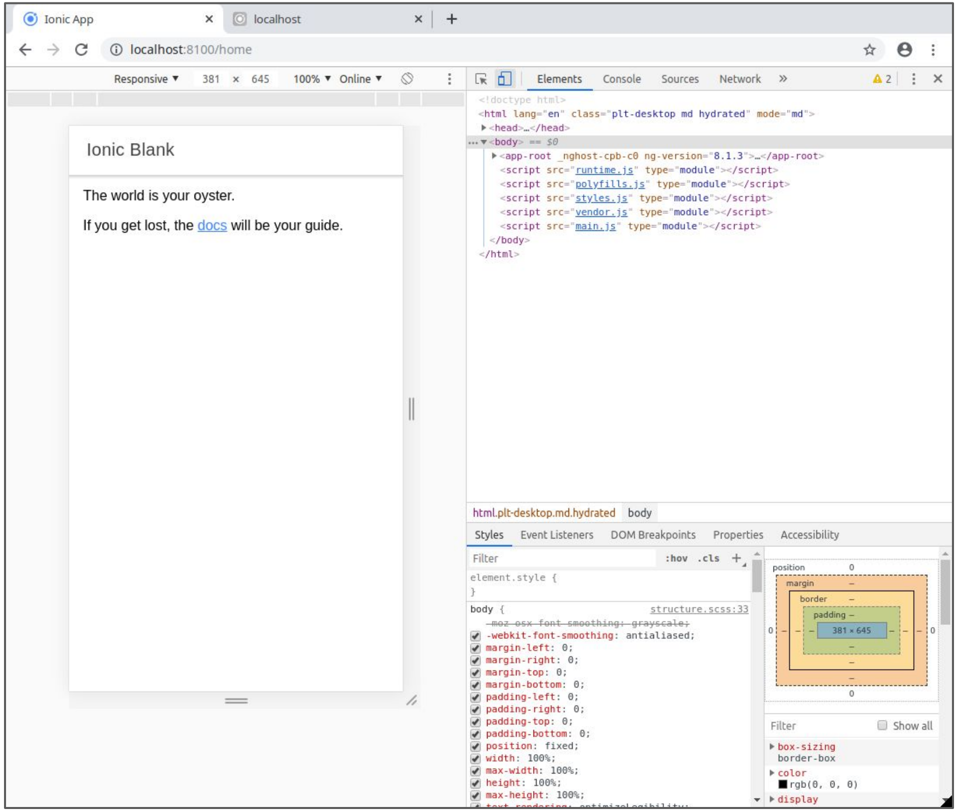
- Serve Ionic project

```
cd [project_name]           // Change directory to project
ionic serve                 // Default: localhost:8100
```

- Serve Ionic project in a lab

```
cd [project_name]           // Change directory to project
npm install @ionic/lab      // Install lab in the project (NO -g flag)
ionic serve -l              // Default for lab: localhost:8200
```

Ionic Project in Browser



Ionic Blank

The world is your oyster.

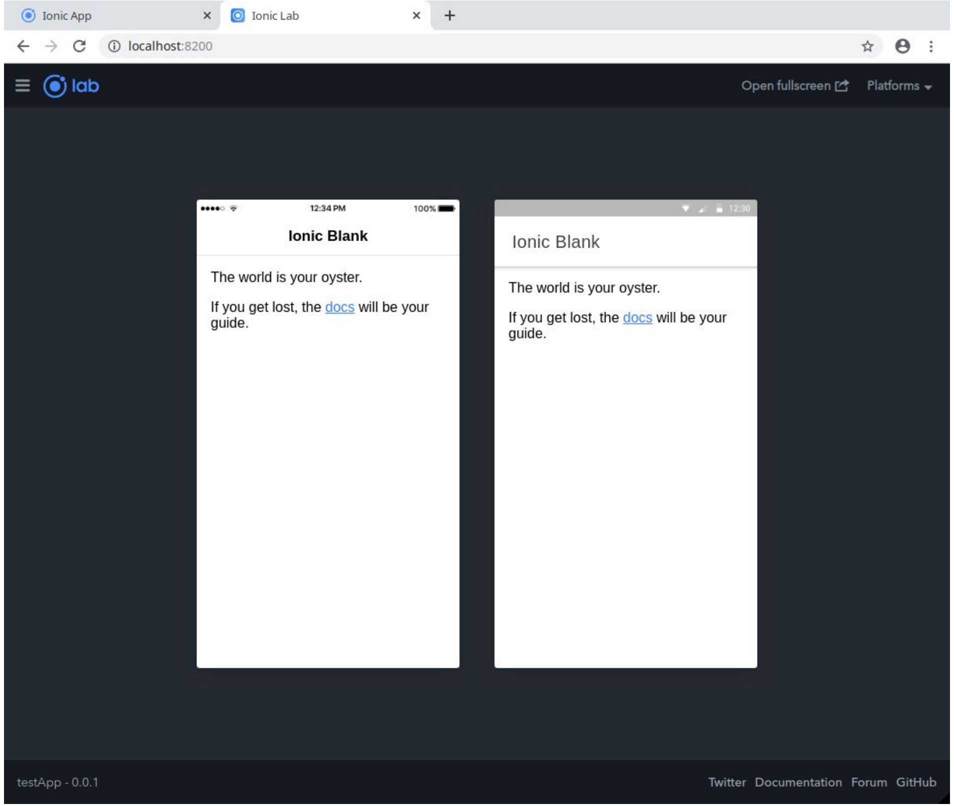
If you get lost, the docs will be your guide.

```
<!doctype html>
<html lang="en" class="plt-desktop md hydrated" mode="md">
 ▶<head>…</head>
…▼<body> == $0
   ▶<app-root _nghost-cpb-c0 ng-version="8.1.3">…</app-root>
    <script src="runtime.js" type="module"></script>
    <script src="polyfills.js" type="module"></script>
    <script src="styles.js" type="module"></script>
    <script src="vendor.js" type="module"></script>
    <script src="main.js" type="module"></script>
  </body>
</html>
```

```
element.style {
}
body {                                        structure.scss:33
  -moz-osx-font-smoothing: grayscale;
☑ -webkit-font-smoothing: antialiased;
☑ margin-left: 0;
☑ margin-right: 0;
☑ margin-top: 0;
☑ margin-bottom: 0;
☑ padding-left: 0;
☑ padding-right: 0;
☑ padding-top: 0;
☑ padding-bottom: 0;
☑ position: fixed;
☑ width: 100%;
☑ max-width: 100%;
☑ height: 100%;
☑ max-height: 100%;
☑ text-rendering: optimizeLegibility;
```

```
▶ box-sizing
  border-box
▶ color
  ■rgb(0, 0, 0)
▶ display
```

Ionic Lab in Browser



Ionic Blank

The world is your oyster.

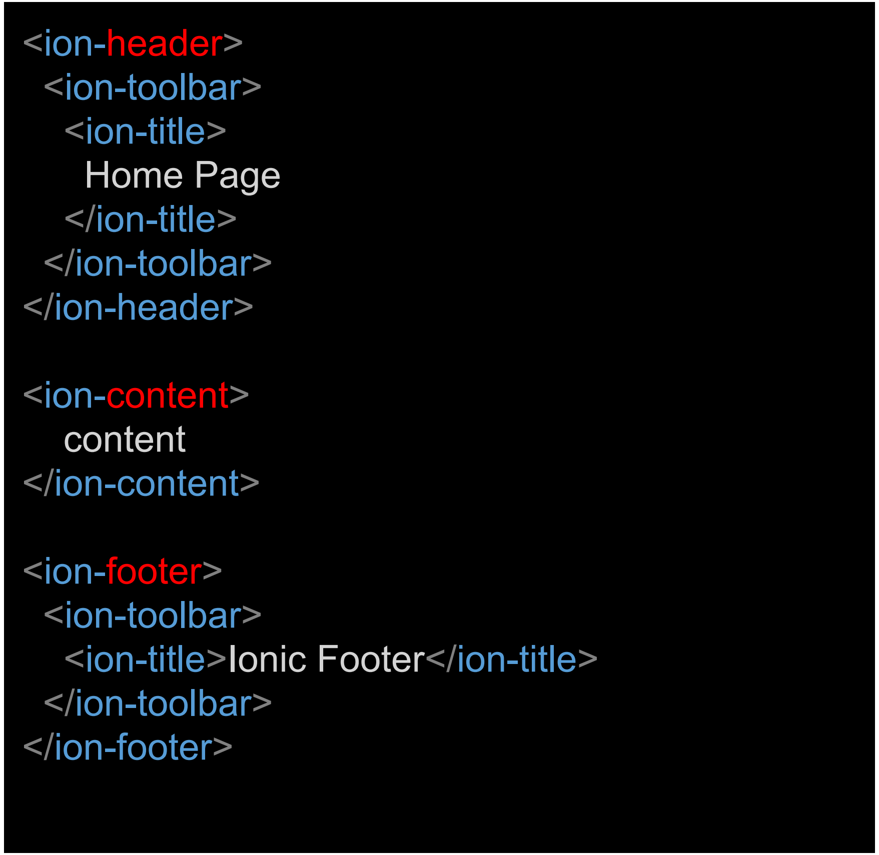
If you get lost, the docs will be your guide.

Ionic Blank

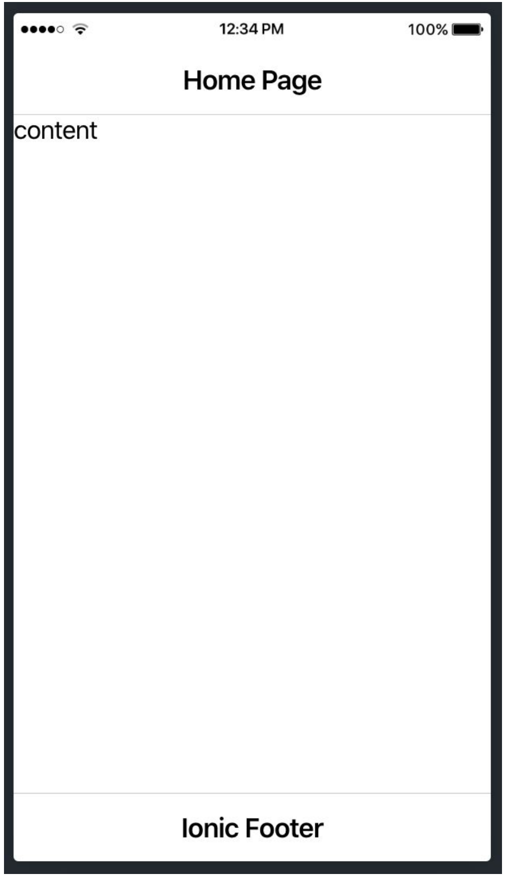
The world is your oyster.

If you get lost, the docs will be your guide.

Ionic Structural Components

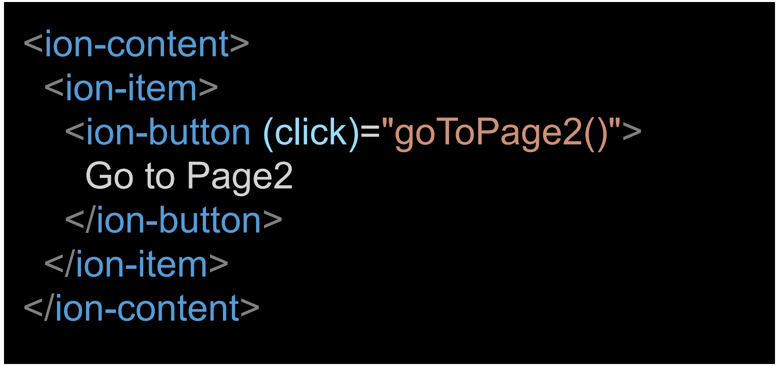


```
<ion-header>
  <ion-toolbar>
    <ion-title>
      Home Page
    </ion-title>
  </ion-toolbar>
</ion-header>

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

<ion-footer>
  <ion-toolbar>
    <ion-title>Ionic Footer</ion-title>
  </ion-toolbar>
</ion-footer>
```



Ionic Items

- <ion-item> is a basic component
- Essentially like <div>
- Creates a block, so its content become a “row”
- Contain other components (e.g.: a button):

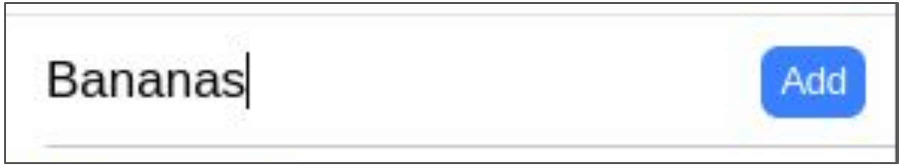


```
<ion-content>
 <ion-item>
  <ion-button (click)="goToPage2()">
   Go to Page2
  </ion-button>
 </ion-item>
</ion-content>
```

Ionic Inputs

- Ionic supports various inputs
  - Today we’ll only focus on text input
- Use [(ngModel)]= to bind to the variable

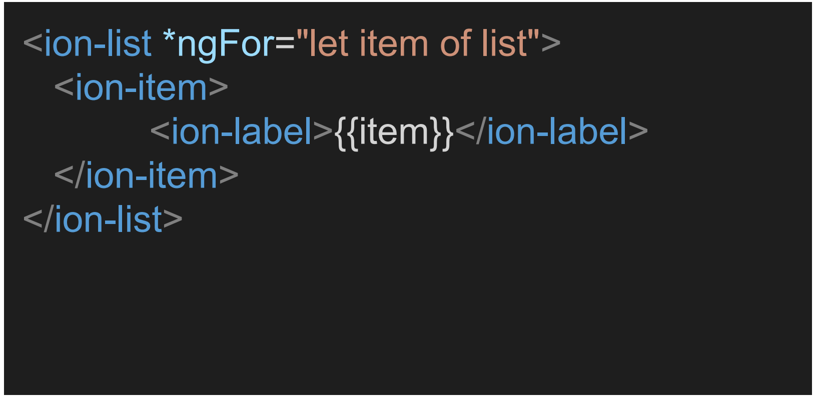
```
<ion-input type="text" placeholder="Item" [(ngModel)]="item"></ion-input>
<ion-button (click)="addItem()"> Add </ion-button>
```



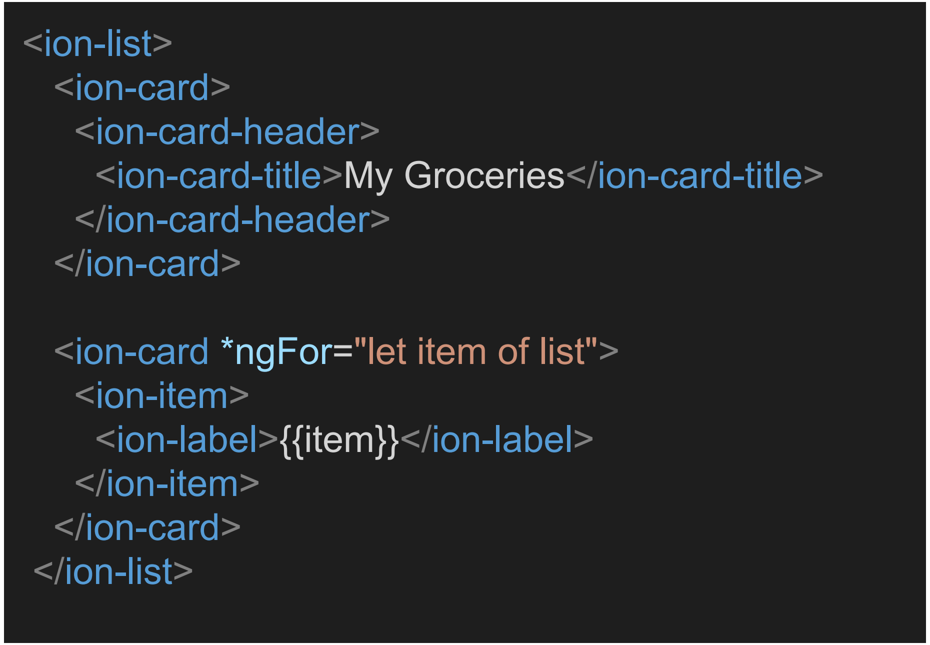
Bananas

Ionic Lists

- Display rows of information
- Items in lists are very customizable



```
<ion-list *ngFor="let item of list">
  <ion-item>
        <ion-label>{{item}}</ion-label>
  </ion-item>
</ion-list>
```



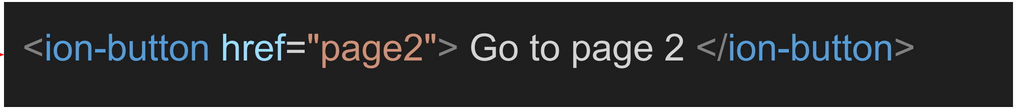
```
<ion-list>
  <ion-card>
    <ion-card-header>
      <ion-card-title>My Groceries</ion-card-title>
    </ion-card-header>
  </ion-card>

  <ion-card *ngFor="let item of list">
    <ion-item>
      <ion-label>{{item}}</ion-label>
    </ion-item>
  </ion-card>
 </ion-list>
```

Ionic Services

- Works very similarly with Angular Services
- Generate service
  - `ionic generate service [service_name]`
- Must include in the “`providers`” in the app.module.ts
- Injecting services work the same way as Angular

Ionic Routing

- Ionic has largely two types of views: component and pages
  - Page is a component that acts as **an entire view** (with some nested components)
  - Components are “smaller”
- To navigate around multiple pages, Ionic uses the app-routing.module.ts
  - Using Ionic CLI to generate pages fills out the app-routing.module.ts for you
  - Generate page: `ionic generate page [page_name]`
- Two different ways to navigate
  - In view

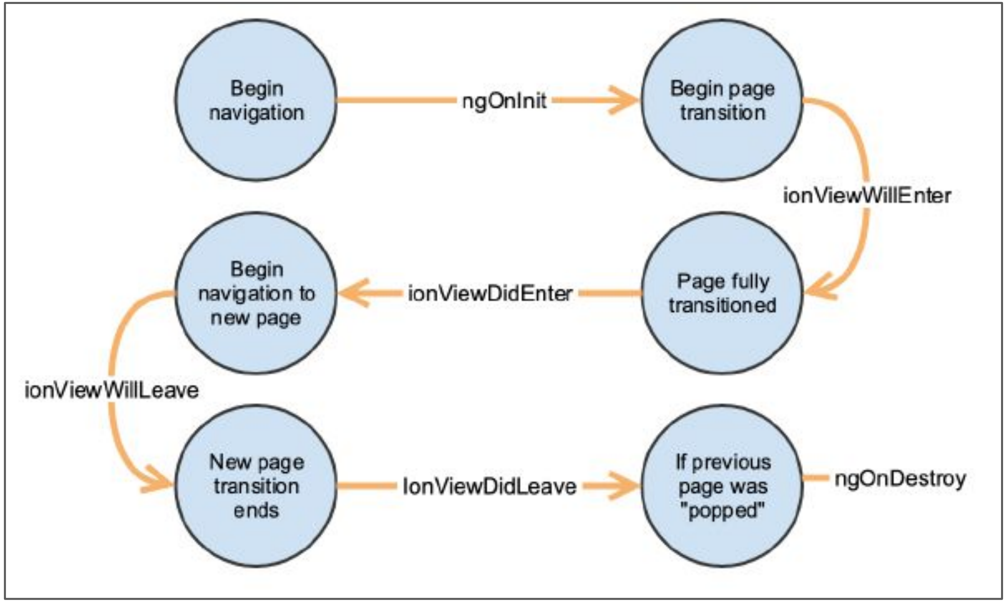


```
<ion-button href="page2"> Go to page 2 </ion-button>
```

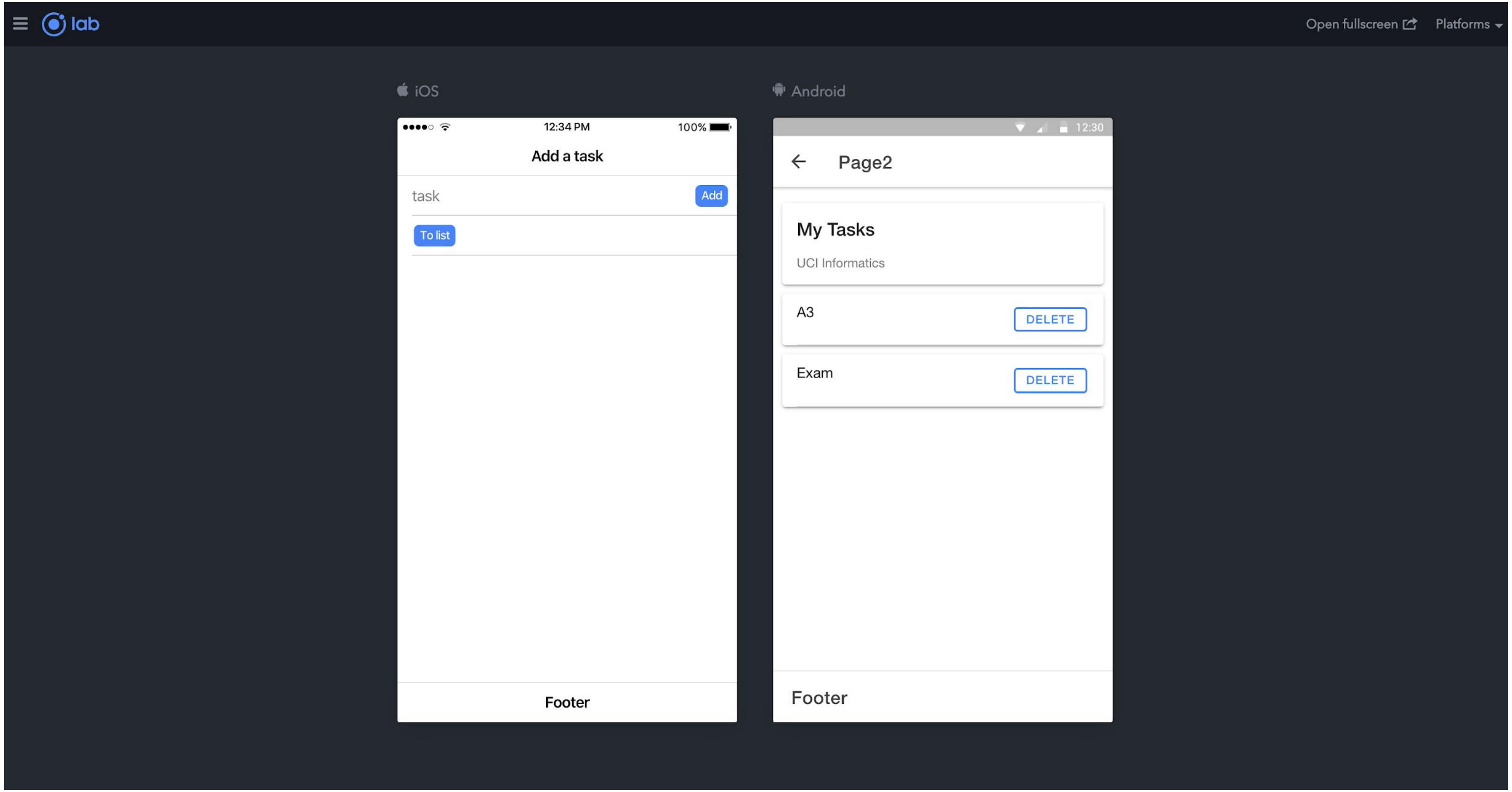
  - In controler
    - More intuitive with an example, so will be explained in the live demo

Ionic Page Lifecycle

- Various events are fired between when a page is generated to its closure
- Mostly used are
  - ngOnInit
    - Fired once when a page is loaded
  - ngOnDestroy
    - Fired right before the view is destroyed.



Ionic Live Demo



Steps for live demo

- Add header, content area, footer in home page
- Generate new page
  - ionic generate page list
- Add navigation between pages
  - home.page.html: add button, set (click)=”goToListPage()”
  - home.page.ts: import NavController, inject, add goToListPage() {this.navCtrl.navigateForward('/page2');}
  - list.page.html: add ion-buttons with slot=start, add ion-back-button with click()=”goBack()”
  - list.page.ts: import NavController, inject, add goBack() {this.navCtrl.back();}
- Add list:String[] in list.page.ts
- Add card-based list in list.page.html
- Generate groceries service: ionic generate service groceries
  - Import and add to app.module.ts providers
- Add input and inject service to home