

Cross Platform Mobile Programming

IF733

Ionic Components: Modal, Loading, Action Sheet, Grid

Andre Rusli

Review

/src/app/app.component.html

```
<ion-content>
  <ion-list>
    <ion-menu-toggle menu="m1">
      <ion-item lines="none" [routerLink]=["/places/tabs/discover/"]>
        <ion-icon name="business" slot="start"></ion-icon>
        <ion-label>Discover Places</ion-label>
      </ion-item>
    </ion-menu-toggle>
    <ion-menu-toggle menu="m1">
      <ion-item lines="none" [routerLink]=["/bookings/"]>
        <ion-icon name="checkbox-outline" slot="start"></ion-icon>
        <ion-label>Your Bookings</ion-label>
      </ion-item>
    </ion-menu-toggle>
    <ion-menu-toggle menu="m1">
      <ion-item lines="none" (click)="onLogout()" button>
        <ion-icon name="exit" slot="start"></ion-icon>
        <ion-label>Logout</ion-label>
      </ion-item>
    </ion-menu-toggle>
  </ion-list>
</ion-content>
```

Add links and switch
pages from Ionic
Menu/Side Drawer

Agenda

- Modal, Loading, & Action Sheet Controller
- More on Ionic Grid and Components
- Sliding Items and Segmented Button

Modal, Loading, and Action Sheet

1/3

Modal Controller

<https://ionicframework.com/docs/api/modal>

1.Create the component for the modal

>> **ionic generate component bookings/create-booking**

2.Declare the component in the `places/discover/place-detail/place-detail.module.ts` file on **declaration** and **entryComponents**

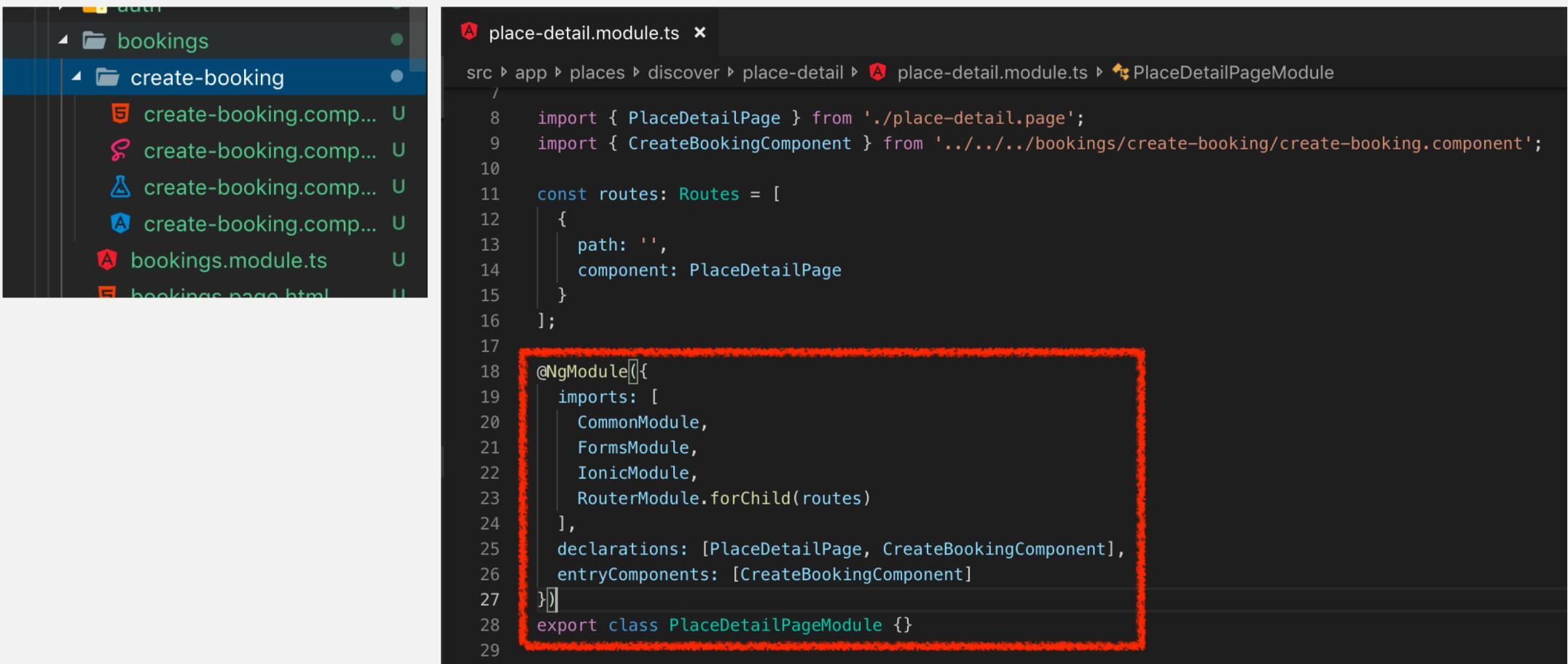
3.Create the button to show the modal

4.Present the modal using **ModalController**

1.Create the component for the modal

>> **ionic generate component bookings/create-booking**

2.Declare the component in the places/discover/place-detail/place-detail.module.ts file on **declaration** and **entryComponents**



The screenshot shows a code editor with a file tree on the left and a code editor window on the right.

File Tree:

- src
- app
- places
- discover
- place-detail
- place-detail.module.ts
- PlaceDetailPageModule
- create-booking
- create-booking.component.ts
- create-booking.component.html
- create-booking.component.css
- create-booking.component.spec.ts
- create-booking.module.ts
- create-booking.page.html

Code Editor (place-detail.module.ts):

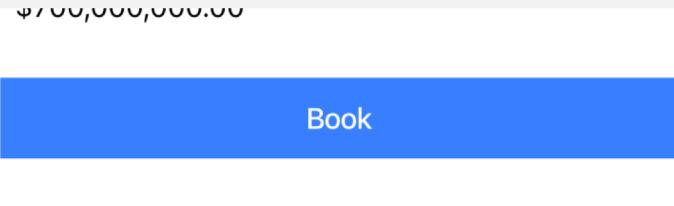
```
src > app > places > discover > place-detail > place-detail.module.ts > PlaceDetailPageModule
/
8  import { PlaceDetailPage } from './place-detail.page';
9  import { CreateBookingComponent } from '../../../../../bookings/create-booking/create-booking.component';
10
11 const routes: Routes = [
12   {
13     path: '',
14     component: PlaceDetailPage
15   }
16 ];
17
18 @NgModule([
19   imports: [
20     CommonModule,
21     FormsModule,
22     IonicModule,
23     RouterModule.forChild(routes)
24   ],
25   declarations: [PlaceDetailPage, CreateBookingComponent],
26   entryComponents: [CreateBookingComponent]
27 ])
28 export class PlaceDetailPageModule {}
```

A red rectangular box highlights the section of the code where the `CreateBookingComponent` is declared in the `entryComponents` array.

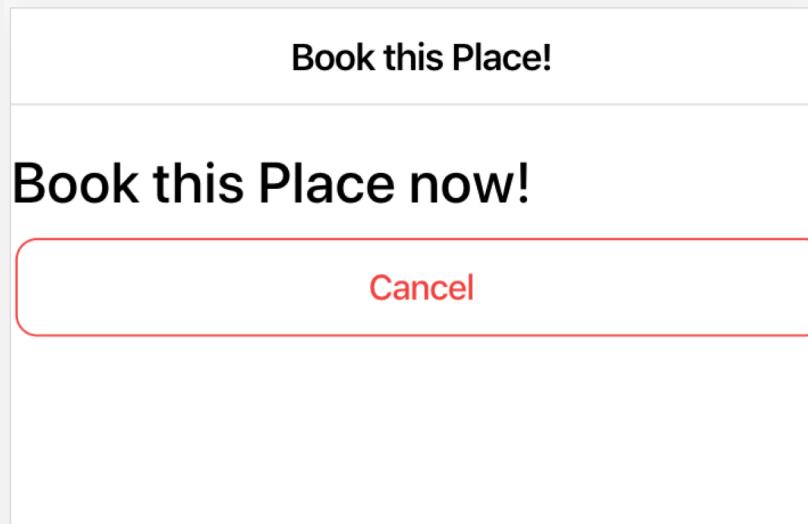
3. Create the button to show the modal
4. Present the modal using **ModalController**

```
<ion-button expand="full" (click)="bookPlace()">Book</ion-button>
```

www.ooo.ooo.ooo.ooo



Book



Book this Place!

Book this Place now!

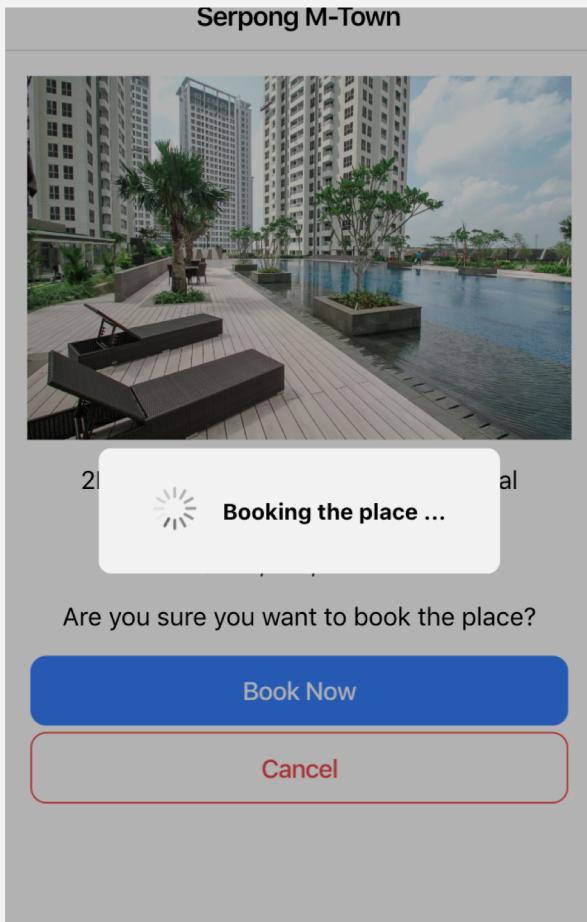
Cancel

```
export class PlaceDetailPage implements OnInit {
  place: Place;
  constructor(
    private router: Router,
    private navCtrl: NavController,
    private route: ActivatedRoute,
    private placesSvc: PlacesService,
    private modalCtrl: ModalController) { }

  bookPlace() {
    this.modalCtrl.create({ component: CreateBookingComponent })
      .then(modalElement => {
        modalElement.present();
      });
  }
}
```

Loading Controller

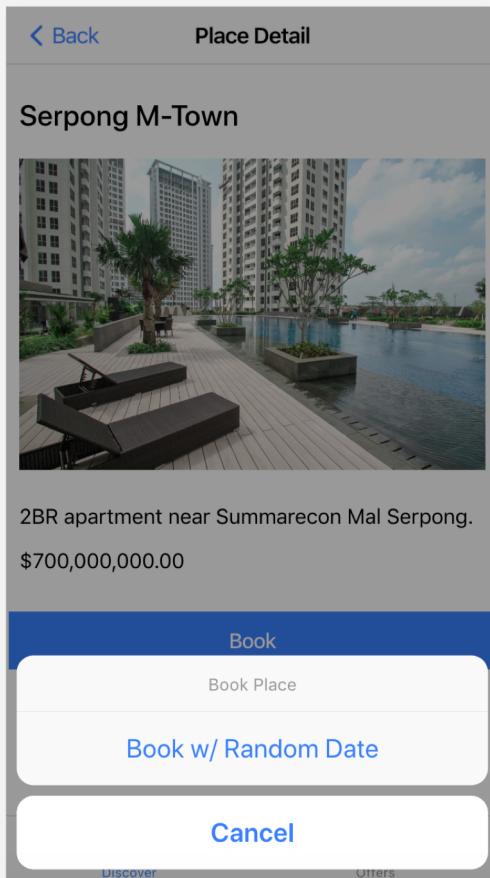
<https://ionicframework.com/docs/api/loading-controller>



```
constructor(private modalCtrl: ModalController,  
private loadingCtrl: LoadingController) { }  
  
bookThisPlace() {  
  // this.isLoading = true;  
  this.loadingCtrl.create({  
    keyboardClose: true,  
    message: 'Booking the place ...'  
})  
.then(loadingEl => {  
  loadingEl.present();  
  setTimeout(() => {  
    loadingEl.dismiss();  
    this.modalCtrl.dismiss({ message: 'booked!' },  
    'confirm');  
  }, 2000);  
}  
}
```

Action Sheet Controller

<https://ionicframework.com/docs/api/action-sheet#usage>



ion-action-sheet

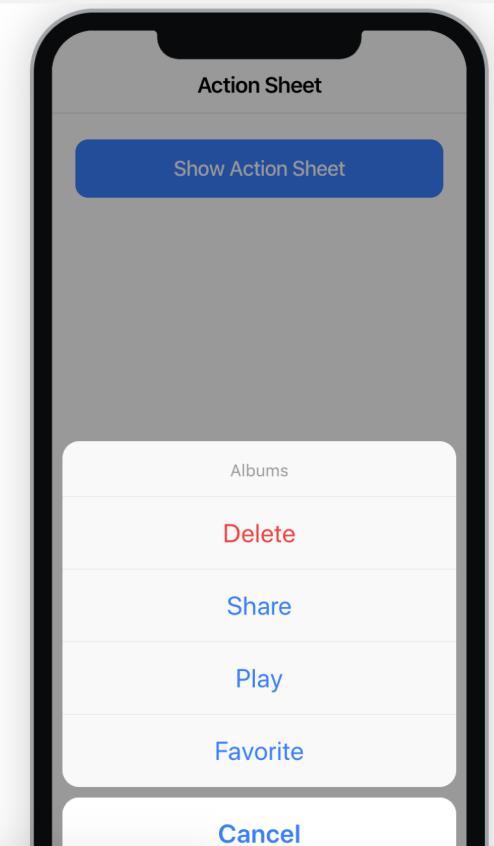
CONTENTS

- Usage
- Properties
- Events
- Methods
- CSS Custom Properties

An Action Sheet is a dialog that displays a set of options. It appears on top of the app's content, and must be manually dismissed by the user before they can resume interaction with the app. Destructive options are made obvious in `ios` mode. There are multiple ways to dismiss the action sheet, including tapping the backdrop or hitting the escape key on desktop.

Creating

An action sheet can be created by the `Action Sheet Controller` from an array of `buttons`, with each button including properties for its `text`, and optionally a `handler` and `role`.



Action Sheet Controller

```
async bookPlace() {
  const actionSheet = await this.actionSheetCtrl.create({
    header: 'Book Place',
    buttons: [
      {
        text: 'Book w/ Random Date',
        handler: () => {
          this.modalCtrl.create({ component: CreateBookingComponent,
            componentProps: { selectedPlace: this.place } })
            .then(modalElement => {
              modalElement.present();
              return modalElement.onDidDismiss();
            })
            .then(resultData => {
              console.log(resultData);
            });
        }
      },
      {
        text: 'Cancel',
        role: 'cancel',
        handler: () => {
          console.log('Cancel clicked');
        }
      }
    ]);
  await actionSheet.present();
}
```

More on Ionic Grids and
Some Other Components!

2/3

`<ion-grid>`

`<ion-row>`

`<ion-col>`

Content here...

`<ion-col>`

Content here...

`<ion-col>`

Content here...

`<ion-row>`

`<ion-row>`

```
<ion-content>
  <ion-grid>
    <ion-row>
      <ion-col>
        Row 1 Col 1
      </ion-col>
      <ion-col>
        Row 1 Col 2
      </ion-col>
      ...
      <ion-col>
        Row 1 Col 12
      </ion-col>
    </ion-row>
  </ion-grid>
</ion-content>
```

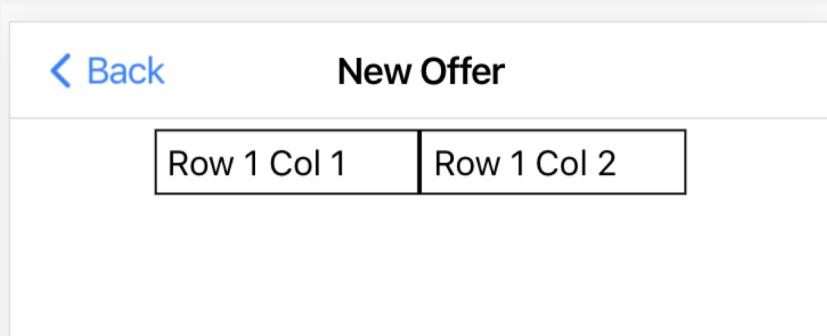
< Back **New Offer**

Row 1 Col 1	Row 1 Col 2	Row 1 Col 3	Row 1 Col 4	Row 1 Col 5	Row 1 Col 6	Row 1 Col 7	Row 1 Col 8	Row 1 Col 9	Row 1 Col 10	Row 1 Col 11	Row 1 Col 12
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	-----------------	-----------------	-----------------

< Back **New Offer**

Row 1 Col 1	Row 1 Col 2	Row 1 Col 3	Row 1 Col 4	Row 1 Col 5	Row 1 Col 6	Row 1 Col 7	Row 1 Col 8	Row 1 Col 9	Row 1 Col 10	Row 1 Col 11	Row 1 Col 12
----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	-----------------	-----------------	-----------------

```
<ion-grid>
  <ion-row>
    <ion-col size='4' offset='2'>
      Row 1 Col 1
    </ion-col>
    <ion-col size='4'>
      Row 1 Col 2
    </ion-col>
  </ion-row>
</ion-grid>
```



Name	Value	Width Prefix	Offset Prefix	Push Prefix	Pull Prefix	Description
xs	0	size-	offset-	push-	pull-	Set columns when (min-width: 0)
sm	576px	size-sm-	offset-sm-	push-sm-	pull-sm-	Set columns when (min-width: 576px)
md	768px	size-md-	offset-md-	push-md-	pull-md-	Set columns when (min-width: 768px)
lg	992px	size-lg-	offset-lg-	push-lg-	pull-lg-	Set columns when (min-width: 992px)
xl	1200px	size-xl-	offset-xl-	push-xl-	pull-xl-	Set columns when (min-width: 1200px)

```
<ion-row>
  <ion-col size="12" size-sm="8" offset-sm="2" text-center>
    <ion-list>
      <ion-item
        *ngFor="let p of loadedPlaces.slice(1)"
        [routerLink]="['/', 'places', 'tabs', 'discover', p.id]">
        <ion-thumbnail slot="start">
          <ion-img [src]="p.imgUrl"></ion-img>
        </ion-thumbnail>
        <ion-label>
          <h2>{{ p.title }}</h2>
          <p>{{ p.description }}</p>
        </ion-label>
      </ion-item>
    </ion-list>
  </ion-col>
</ion-row>
```

 Discover Places

Serpong M-Town

\$700,000,000.00



2BR apartment near Summarecon Mal Serpong.

[More](#)

 Discover Places

Serpong M-Town

\$700,000,000.00

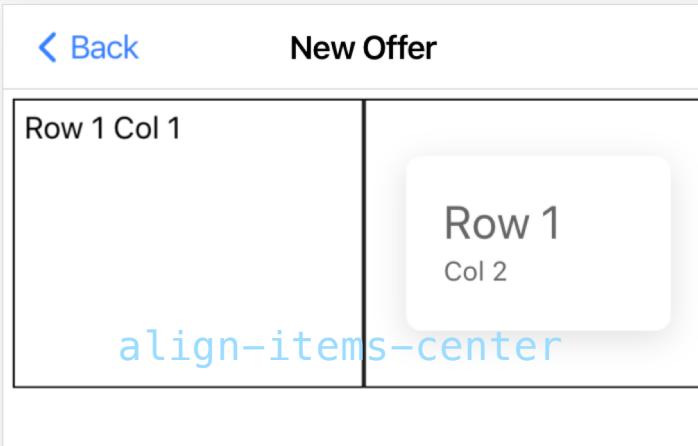


2BR apartment near Summarecon Mal Serpong.

Grid Alignments

~ Vertical and Horizontal ~

```
<ion-row>
  <ion-col>
    Row 1 Col 1
  </ion-col>
  <ion-col>
    <ion-card>
      <ion-card-content>
        <h1>Row 1</h1>
        <p>Col 2</p>
      </ion-card-content>
    </ion-card>
  </ion-col>
</ion-row>
```



```
<ion-row align-items-center>
  <ion-col>
    Row 1 Col 1
  </ion-col>
  <ion-col>
    <ion-card>
      <ion-card-content>
        <h1>Row 1</h1>
        <p>Col 2</p>
      </ion-card-content>
    </ion-card>
  </ion-col>
</ion-row>
```



Grid Alignments

~ Vertical and Horizontal ~

```
<ion-grid>
  <ion-row align-items-center justify-content-end>
    <ion-col size="4">
      Row 1 Col 1
    </ion-col>
    <ion-col size="4">
      <ion-card>
        <ion-card-content>
          <h1>Row 1</h1>
          <p>Col 2</p>
        </ion-card-content>
      </ion-card>
    </ion-col>
  </ion-row>
</ion-grid>
```



<ion-list> OR <ion-grid>

<ion-list>

Renders <ion-item>s vertically

Should only contain <ion-item>s

Use for scrollable, vertical list content

<ion-grid>

Renders ANY content (inside of <ion-col>s)

<ion-grid> => <ion-row> => <ion-col> => *

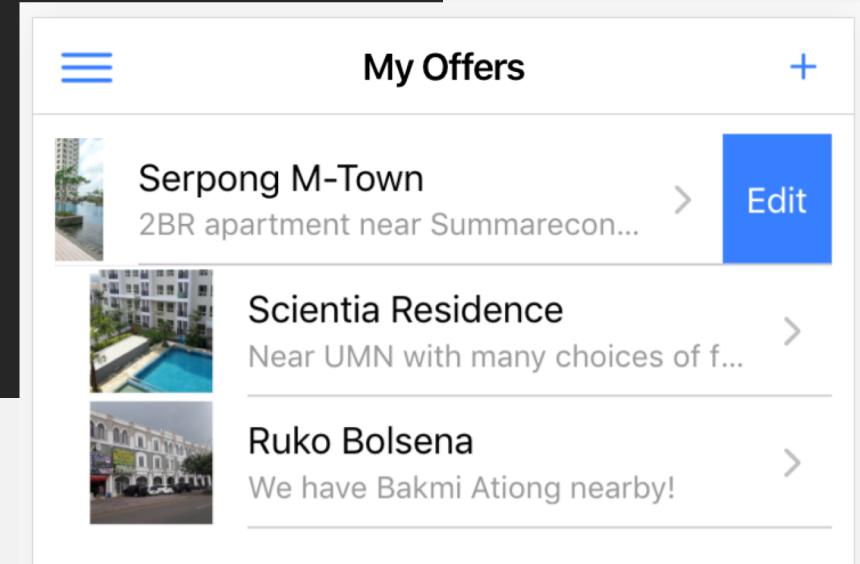
Use for any content that should be structured in a grid

Sliding Items and Segmented Button

3/3

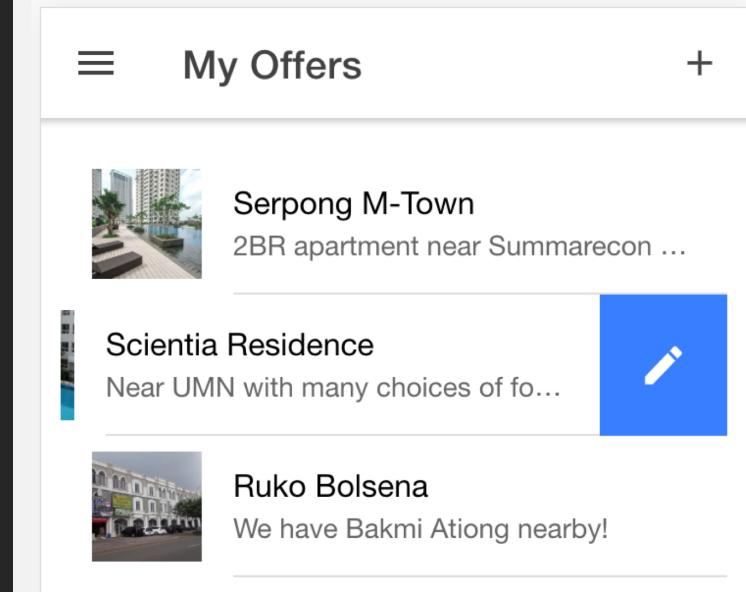
Sliding Items

```
<ion-list>
  <ion-item-sliding *ngFor="let p of loadedPlaces">
    <ion-item-options side="end">
      <ion-item-option (click)="editOffer()">Edit</ion-item-option>
    </ion-item-options>
    <ion-item [routerLink]="['/', 'places', 'tabs', 'offers', p.id]">
      <ion-thumbnail slot="start">
        <ion-img [src]="p imgUrl"></ion-img>
      </ion-thumbnail>
      <ion-label>
        <h2>{{ p.title }}</h2>
        <p>{{ p.description }}</p>
      </ion-label>
    </ion-item>
  </ion-item-sliding>
</ion-list>
```



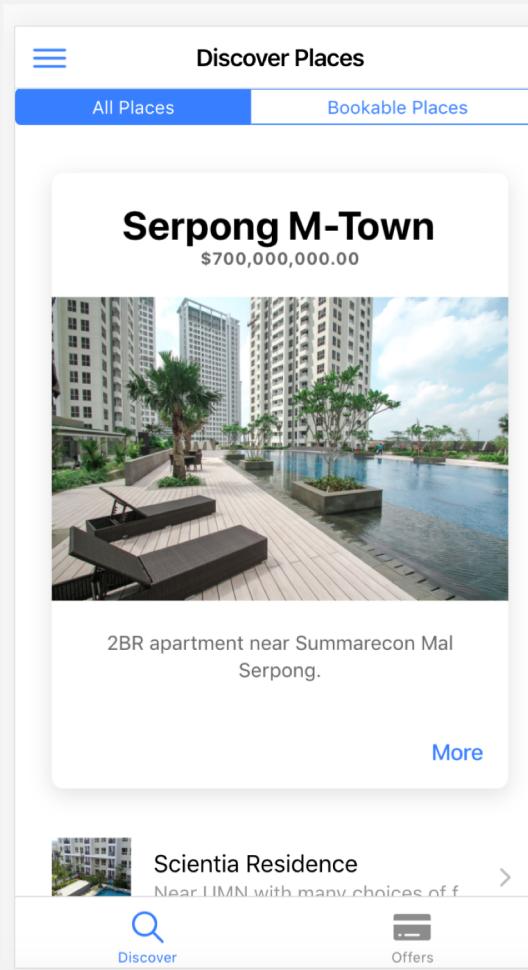
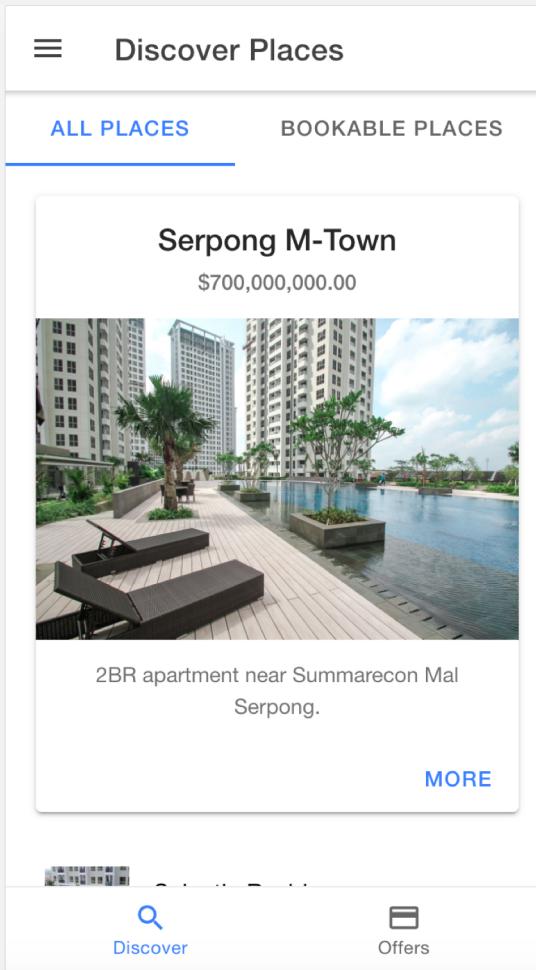
Sliding Items

```
<ion-item-sliding *ngFor="let p of loadedPlaces" #slidingItem>
  <ion-item-options side="end">
    <ion-item-option (click)="editOffer(p.id, slidingItem)">
      <ion-icon name="create" slot="icon-only"></ion-icon>
    </ion-item-option>
  </ion-item-options>
  <ion-item [routerLink]="/places/tabs/offers/p.id">
    <ion-thumbnail slot="start">
      <ion-img [src]=>p.imgUrl</ion-img>
    </ion-thumbnail>
    <ion-label>
      <h2>{{ p.title }}</h2>
      <p>{{ p.description }}</p>
    </ion-label>
  </ion-item>
</ion-item-sliding>
```



```
editOffer(id: string, slidingItem: IonItemSliding) {
  slidingItem.close();
  this.router.navigate(['/places/tabs/offers/edit', id]);
}
```

Segmented Button



Angular is running in the [core.js:16829](#) development mode. Call enableProdMode() to enable the production mode.

⚠ ▶ Native: tried calling [common.js:290](#) StatusBar.styleDefault, but Cordova is not available. Make sure to include cordova.js or run in a device/simulator

⚠ ▶ Native: tried calling [common.js:290](#) SplashScreen.hide, but Cordova is not available. Make sure to include cordova.js or run in a device/simulator

▶ `{value: "all"}` [discover.page.ts:26](#)

▶ `{value: "bookable"}` [discover.page.ts:26](#)

>

Segmented Button

```
<ion-content>
  <ion-segment (ionChange)="onFilterUpdate($event)">
    <ion-segment-button value="all" checked>All Places</ion-segment-button>
    <ion-segment-button value="bookable">Bookable Places</ion-segment-button>
  </ion-segment>
  <ion-grid>
    <ion-row>
      <ion-col size="12" size-sm="8" offset-sm="2" text-center>
        <ion-card>
          <ion-card-header>
            <ion-card-title>{{ loadedPlaces[0].title }}</ion-card-title>
            <ion-card-subtitle>
              {{ loadedPlaces[0].price | currency }}
            ...
          </ion-card>
        </ion-col>
      </ion-row>
    </ion-grid>
  </ion-content>
```

```
onFilterUpdate(evt: CustomEvent<SegmentChangeEventDetail>) {
  console.log(evt.detail);
}
```

Questions?

Comments?

NEXT

- asd

Thank you