

# Cross Platform Mobile Programming

IF733

Building in Real Devices & Navigation: Tabs and Side Drawer

Andre Rusli

# Review

- Route parameters
- Page navigation
- Ionic controller component: Alert & Toast

```
deleteRecipe() {  
    this.recipesSvc.deleteRecipe(this.loadedRecipe.id);  
    this.router.navigate(['/recipes']);  
    this.presentToast();  
}
```

src/app/recipes/recipe-detail/recipe-detail.page.ts

```
async presentToast() {  
    const toast = await  
    this.toastController.create({  
        message: 'Recipe has been deleted.',  
        duration: 2000,  
        position: 'bottom'  
    });  
    toast.present();  
}
```

# Agenda

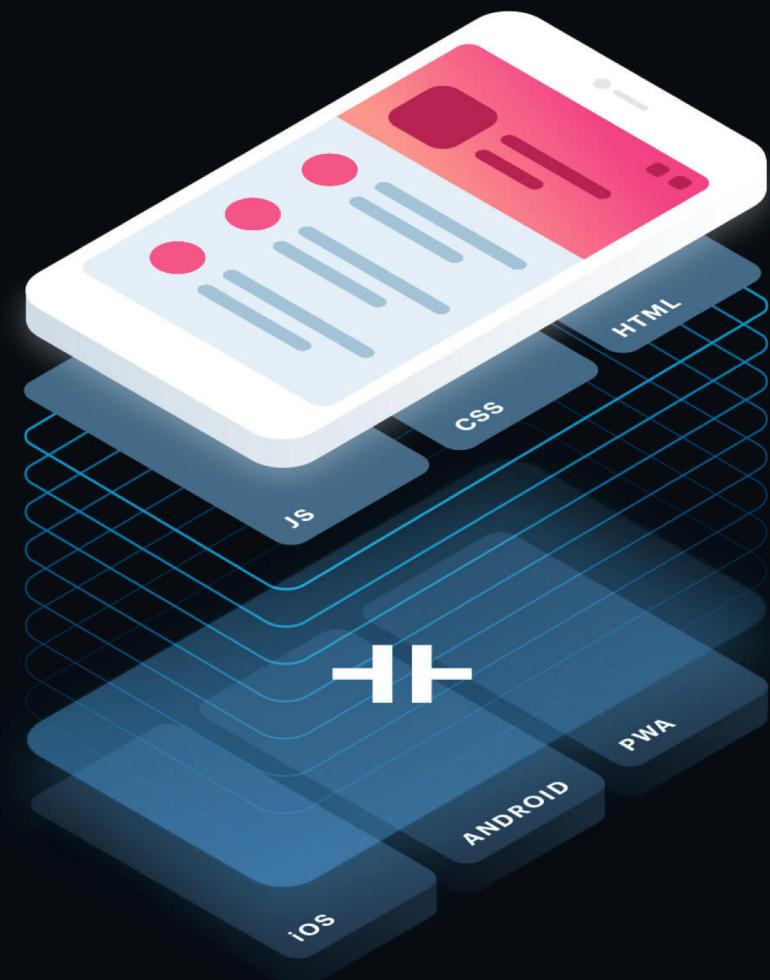
- Building in Native Devices
- Debugging
- Navigation: Tabs and Side Drawer

# Building in Native Devices

1/3

# The Native Bridge for Cross-Platform Web Apps

Invoke Native SDKs on iOS, Android, Electron, and the Web with one code base. Optimized for Ionic Framework apps, or use with any web app framework.

[Get Started](#)**SUPPORTS**

# Building for Android

## Android Development

First, the Java 8 JDK must be installed and [set to the default](#) if you have other versions of the JDK installed. Java 9 does *not* work at the moment.

Android development requires the Android SDK installed with [Android Studio](#). Technically, Android Studio isn't required as you can build and run apps using only the Android CLI tools, but it will make building and running your app much easier so we strongly recommend using it.

Android version support for Capacitor is more complex than iOS. Currently, we are targeting API level 21 or greater, meaning Android 5.0 (Lollipop) or above. [As of May 2019](#), this represents over 89% of the Android market.

Also, Capacitor requires an Android WebView with Chrome version 50 or later. On Android 5 and 6, the Capacitor uses the System WebView. On Android 7+, Google Chrome is used.

# Building for Android

Several useful commands

```
> ng build  
> ionic capacitor add android  
> ionic capacitor copy android  
> ionic capacitor run android  
> ionic capacitor run android -l
```

# Building for iOS

## iOS Development

For building iOS apps, Capacitor requires a Mac with Xcode 10 or above. Soon, you'll be able to use [Ionic Appflow](#) to build for iOS even if you're on Windows.

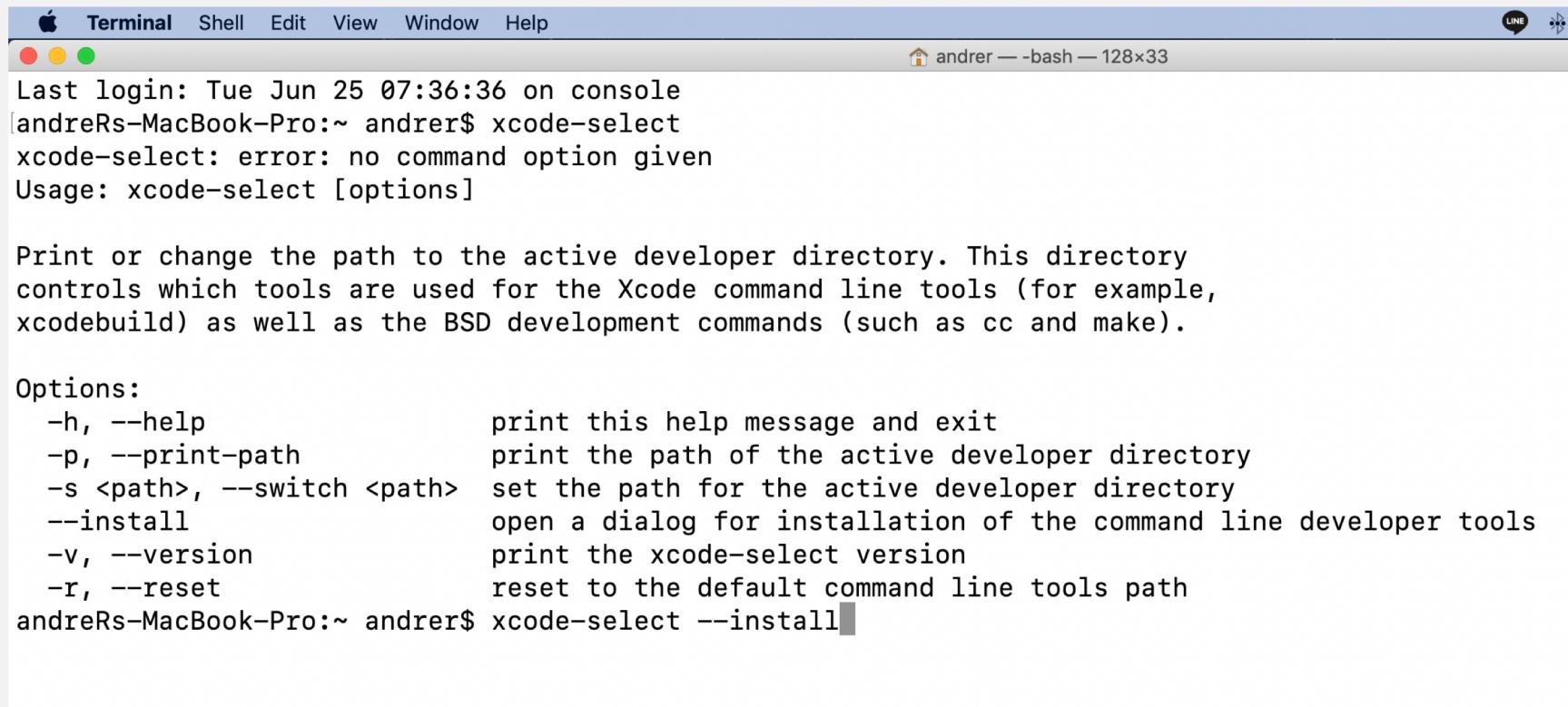
Additionally, you'll need to install [CocoaPods](#) (`sudo gem install cocoapods`), and install the Xcode Command Line tools (either from Xcode, or running `xcode-select --install`).

Once you have CocoaPods installed, update your local repo by running `pod repo update`. You should run this command periodically to ensure you have the latest versions of CocoaPods dependencies.

As a rule, the latest version of Capacitor always supports the last two iOS versions. For example, iOS 11 and iOS 12. For support for older versions of iOS, use an older version of Capacitor (if available).

Capacitor uses the WKWebView.

# Building for iOS



A screenshot of a Mac OS X Terminal window. The window title is "Terminal". The menu bar includes "Terminal", "Shell", "Edit", "View", "Window", and "Help". The status bar shows "andrer — bash — 128x33". The terminal output is as follows:

```
Last login: Tue Jun 25 07:36:36 on console  
[andreRs-MacBook-Pro:~ andrer$ xcode-select  
xcode-select: error: no command option given  
Usage: xcode-select [options]  
  
Print or change the path to the active developer directory. This directory  
controls which tools are used for the Xcode command line tools (for example,  
xcodebuild) as well as the BSD development commands (such as cc and make).  
  
Options:  
-h, --help          print this help message and exit  
-p, --print-path   print the path of the active developer directory  
-s <path>, --switch <path> set the path for the active developer directory  
--install          open a dialog for installation of the command line developer tools  
-v, --version       print the xcode-select version  
-r, --reset         reset to the default command line tools path  
andreRs-MacBook-Pro:~ andrer$ xcode-select --install
```

# Building for iOS

Several useful commands

```
> ionic capacitor add ios  
> ng build  
> ionic capacitor copy ios  
> ionic capacitor run ios  
> ionic capacitor run ios -l
```

# Building for iOS

The screenshot shows the Xcode interface with the General tab selected in the top navigation bar. The main area displays configuration for an app target named "App".

**Identity**

- Display Name: w2\_2019
- Bundle Identifier: id.ac.umn.ionicRecipeApp
- Version: 1.0
- Build: 1

**Signing**

- Automatically manage signing  
Xcode will create and update profiles, app IDs, and certificates.
- Team: Andre Rusli (Personal Team)
- Provisioning Profile: Xcode Managed Profile
- Signing Certificate: iPhone Developer: andrerusli92@gmail.com (QE8...)

**Deployment Info**

- Deployment Target: 11.0
- Devices: Universal
- Main Interface: Main

# Debugging

## 2/3

# Error Messages & console.log()

```
Error: Invalid configuration of route main.ts:12
'recipes'. One of the following must be provided:
component, redirectTo, children or loadChildren
  at validateNode (router.js:607)
  at validateConfig (router.js:577)
  at
Router.push../node_modules/@angular/router/fesm5/
router.js.Router.resetConfig (router.js:4108)
  at new Router (router.js:3784)
  at setupRouter (router.js:5581)
  at _callFactory (core.js:21292)
  at _createProviderInstance (core.js:21238)
  at initNgModule (core.js:21168)
  at new NgModuleRef_ (core.js:21895)
  at createNgModuleRef (core.js:21884)
```

```
    ,
export class RecipesPage implements OnInit {
  recipes: Recipe[];

  constructor(private recipesService: RecipesService) { }

  ngOnInit() {
    this.recipes = this.recipesService.getAllRecipes();
    console.log(this.recipes);
  }

  getRecipe(recipeId: string){
    console.log(this.recipesService.getRecipe(recipeId));
  }
}
```

# Chrome DevTools

localhost:8100/recipes

iPhone 6/7/8 ▾ 375 x 667 96% ▾ No throttling ▾

## Recipes

- Gado-gado >
- Ketupat >
- Pizza Margerita >

Elements Console Sources Network Performance Memory Application > ① ② ③

Page > recipes.service.ts recipes.page.ts >

```
1 import { Component, OnInit } from '@angular/core';
2 import { Recipe } from './recipe.model';
3 import { RecipesService } from './recipes.service';
4
5 @Component({
6   selector: 'app-recipes',
7   templateUrl: './recipes.page.html',
8   styleUrls: ['./recipes.page.scss'],
9 })
10 export class RecipesPage implements OnInit {
11   recipes: Recipe[];
12
13   constructor(private recipesService: RecipesService) {
14     this.recipes = this.recipesService.getAllRecipes();
15     console.log(this.recipes);
16   }
17
18   ngOnInit() {
19     this.recipes = this.recipesService.getAllRecipes();
20     console.log(this.recipes);
21   }
22
23   getRecipe(recipeId: string) {
24     console.log(this.recipesService.getRecipe(recipeId));
25   }
}
```

Threads Main service-worker.js #5611 (activated)

Watch Call Stack Not paused

Scope Not paused

Breakpoints recipes.page.ts:20 getRecipe(recipeId: string){ checked }

XHR/fetch Breakpoints

DOM Breakpoints

Global Listeners

Event Listener Breakpoints

Console What's New

Highlights from the Chrome 75 update

Meaningful preset values when autocompleting CSS functions

Properties like "filter" that take functions for values now autocomplete to previewable values in the Styles pane.

Clear site data from the Command Menu

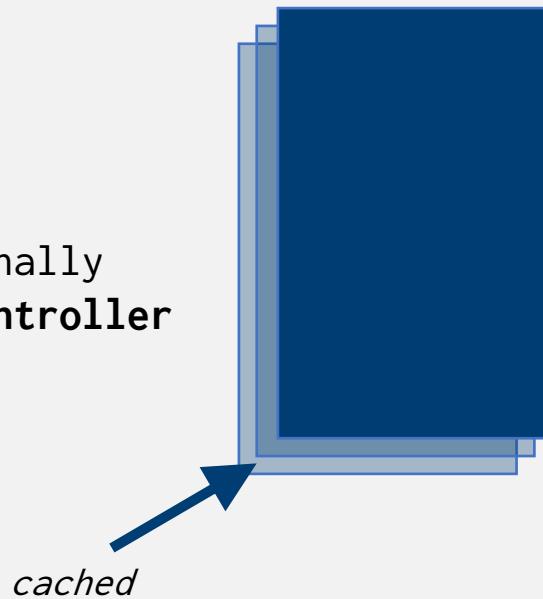
Press Command + Shift + P and run the "Clear Site Data" command to clear cookies, storage, and more.



Navigation: Tabs and Side Drawer  
3/3

# How Navigation Works

**Stack of pages**  
controlled internally  
by **Ionic StackController**



Angular Router

Ionic NavController

# Page Lifecycle

Angular Components

ngOnInit

ngOnDestroy

Ionic Pages

ionViewWillEnter

ionViewDidEnter

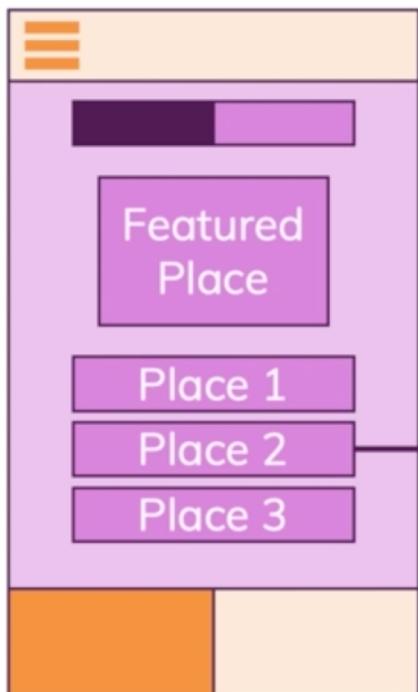
ionViewWillLeave

ionViewDidEnter

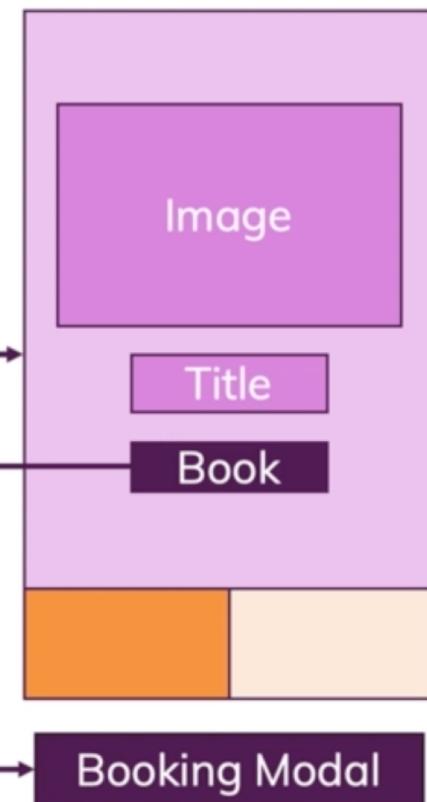


Let's create an app for the next few lectures.

## Find Places



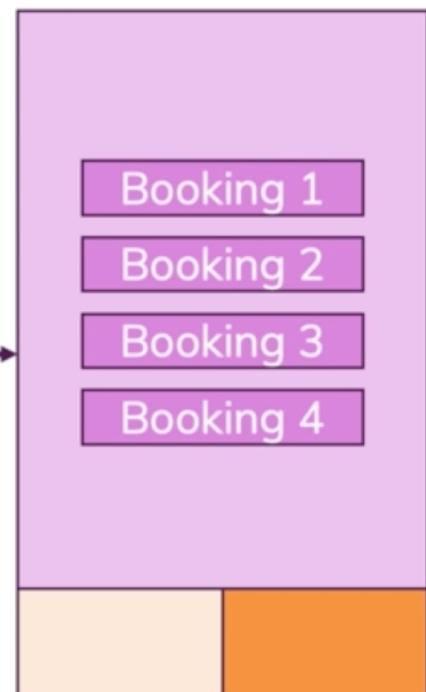
## Place Detail



## Offers



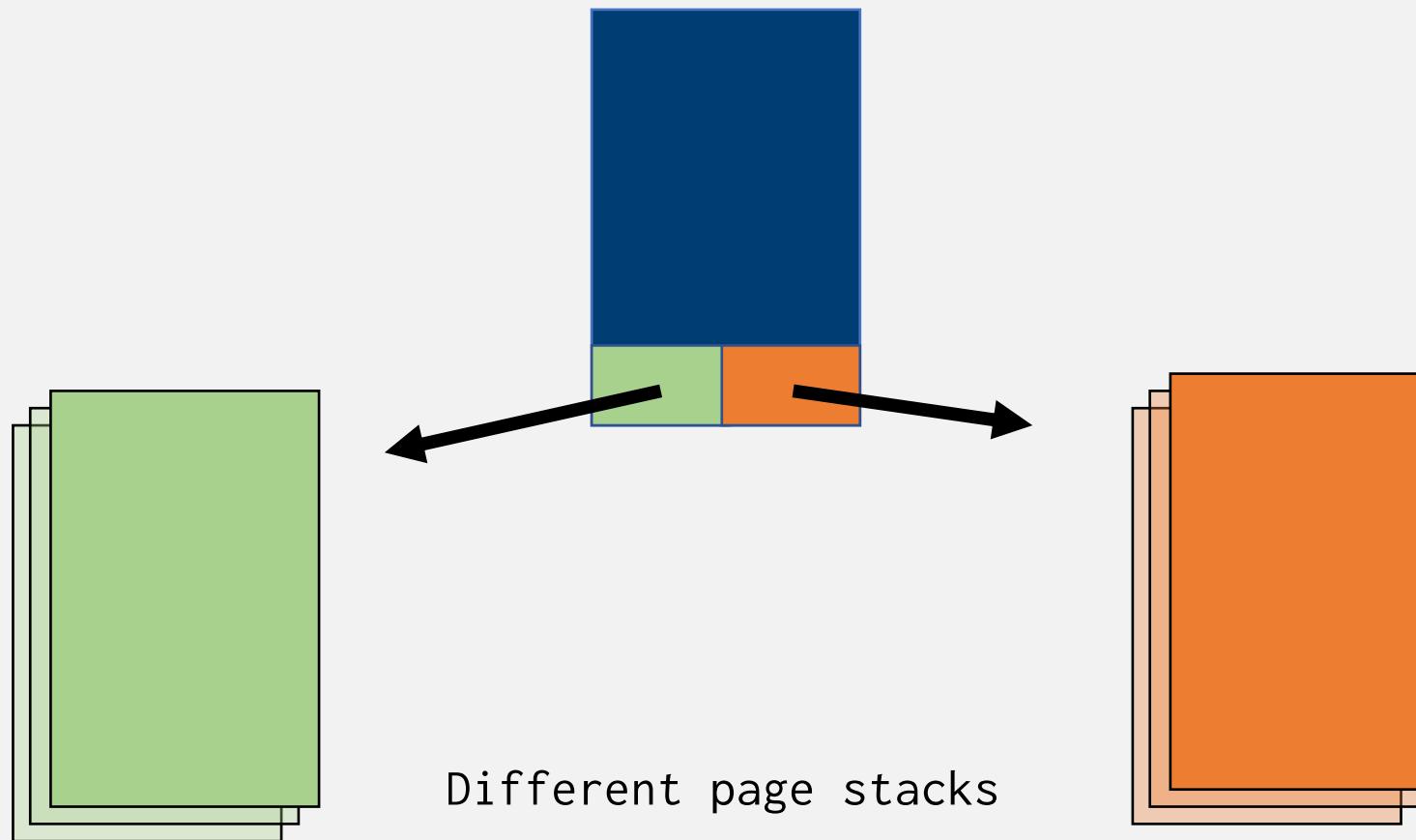
## Offer Details



Booking Modal



# Navigation: Tabs



# Navigation: Creating Tabs

src/app/places/places.page.html

```
<ion-tabs>
  <ion-tab-bar>
    <ion-tab-button tab="discover">
      <ion-label>Discover</ion-label>
      <ion-icon name="search"></ion-icon>
    </ion-tab-button>
    <ion-tab-button tab="offers">
      <ion-label>Offers</ion-label>
      <ion-icon name="card"></ion-icon>
    </ion-tab-button>
  </ion-tab-bar>
</ion-tabs>
```



Discover



Offers

# Navigation: Creating Tabs

## -Setting up the AppRoutingModule -

src/app/app-routing.module.ts

```
const routes: Routes = [
  { path: '', redirectTo: 'places', pathMatch: 'full' },
  { path: 'auth', loadChildren: './auth/auth.module#AuthPageModule' },
  { path: 'bookings', loadChildren: './bookings/bookings.module#BookingsPageModule' },
  { path: 'places', loadChildren: './places/places.module#PlacesPageModule' },
];

@NgModule({
  imports: [
    RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules })
  ],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

# Navigation: Creating Tabs

-Setting up the `PlacesPageModule` -

`src/app/places/places.module.ts`

```
import { PlacesPage } from './places.page';
import { PlacesRoutingModule } from './places-routing.module';

@NgModule({
  imports: [
    CommonModule,
    IonicModule,
    PlacesRoutingModule,
  ],
  declarations: [PlacesPage]
})
export class PlacesPageModule {}
```

## src/app/places/places-routing.module.ts

```
const routes: Routes = [
  {
    {
      path: '',
      redirectTo: '/places/tabs/discover',
      pathMatch: 'full'
    },
    {
      path: 'tabs',
      component: PlacesPage,
      children: [
        {
          path: 'discover',
          children: [
            {
              path: '',
              loadChildren: './discover/discover.module#DiscoverPageModule'
            },
            ...
            {
              path: 'offers',
              children: [ // the order matters, hardcoded path must go before the dynamic path or else
                {
                  path: '',
                  loadChildren: './offers/offers.module#OffersPageModule'
                },
                {
                  path: 'new',
                  loadChildren: './offers/new-offer/new-offer.module#NewOfferPageModule'
                },
                ...
              ]
            }
          ]
        }
      ]
    }
  ]
]
```

src/app/places/places-routing.module.ts

```
...
...
@NgModule({
  imports: [RouterModule.forChild(routes)],
  exports: [RouterModule],
})
export class PlacesRoutingModule {}
```

# Navigation: Back Button (1)

Using the <ion-back-button>

```
<ion-header>
  <ion-toolbar>
    <ion-buttons>
      <ion-back-button defaultHref="/places/tabs/discover"></ion-back-button>
    </ion-buttons>
    <ion-title>Place Detail</ion-title>
  </ion-toolbar>
</ion-header>
```

# Navigation: Back Button (2)

Using Angular Router

```
<ion-toolbar>
  <ion-buttons>
    <ion-button color="primary" margin (click)="goBack()">
      <ion-icon name="arrow-back"></ion-icon>
      Back
    </ion-button>
  </ion-buttons>
  <ion-title>Place Detail</ion-title>
</ion-toolbar>
</ion-header>
```

```
export class PlaceDetailPage implements OnInit {

  constructor(private router: Router) { }

  ngOnInit() {
  }

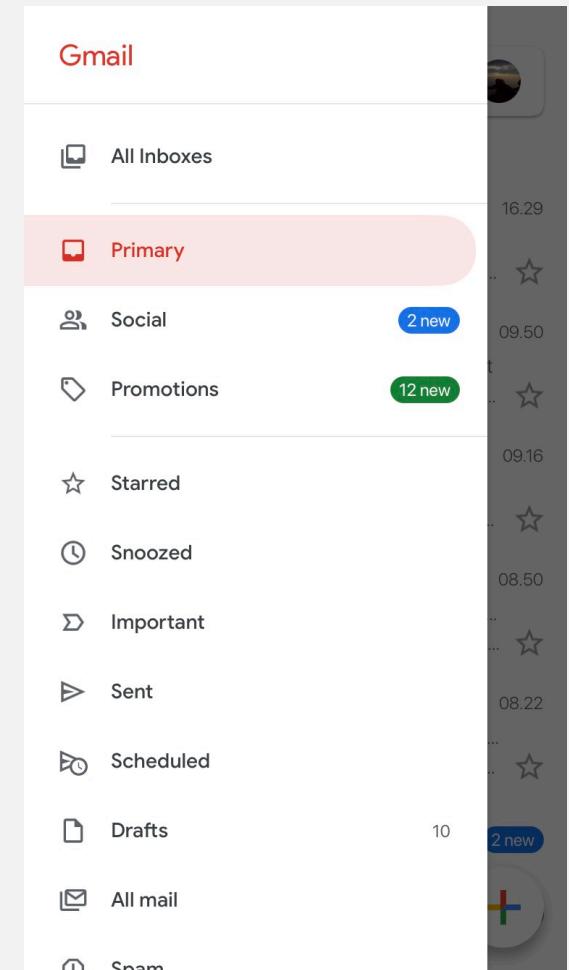
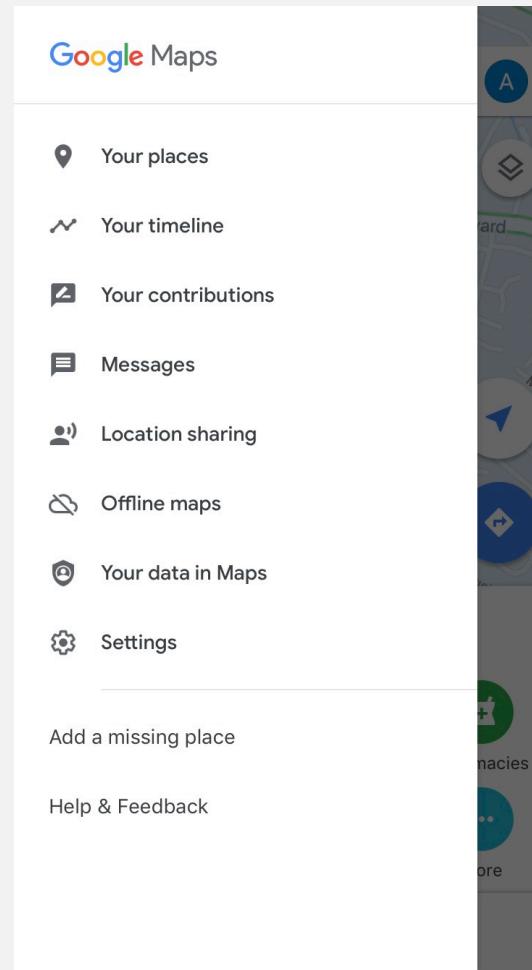
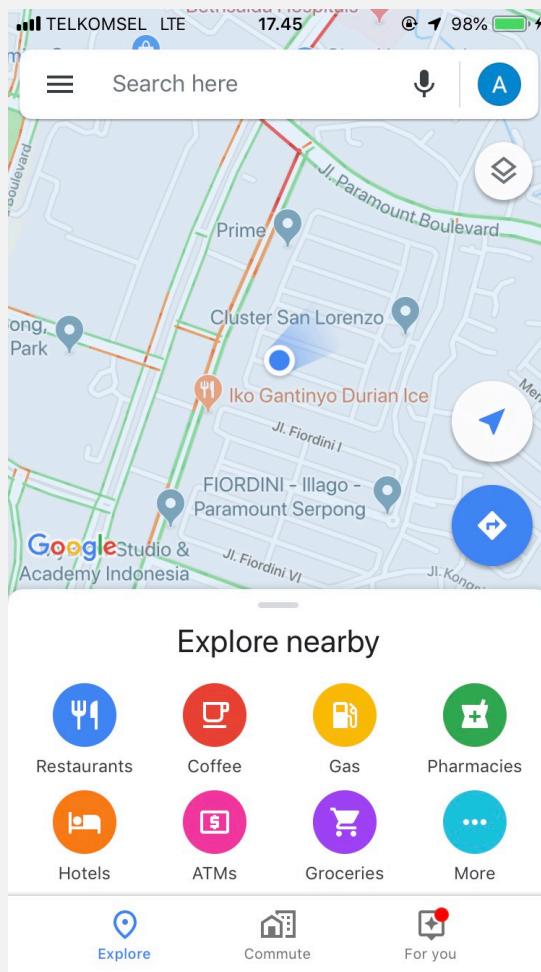
  goBack() {
    this.router.navigateByUrl('/places/tabs/discover');
  }
}
```

# Navigation: Back Button (3)

Using Ionic NavController

```
<ion-toolbar>
  <ion-buttons>
    <ion-button color="primary" margin (click)="goBack()">
      <ion-icon name="arrow-back"></ion-icon>
      Back
    </ion-button>
  </ion-buttons>
  <ion-title>Place Detail</ion-title>
</ion-toolbar>
</ion-header>
goBack() {
  // this.router.navigateByUrl('/places/tabs/discover');
  this.navCtrl.navigateBack('/places/tabs/discover');
  // or you can also use pop() below
  // this.navCtrl.pop();
}
```

# Navigation: SideDrawer



/src/app/app.  
component.ts

```
<ion-app>
  <ion-menu side="start" menuId="m1">
    <ion-header>
      <ion-toolbar>
        <ion-title>Ionic Places App</ion-title>
      </ion-toolbar>
    </ion-header>
    <ion-content>
      <ion-list>
        <ion-item lines="none">
          <ion-icon name="business" slot="start"></ion-icon>
          <ion-label>Discover Places</ion-label>
        </ion-item>
        <ion-item lines="none">
          <ion-icon name="checkbox-outline" slot="start"></ion-icon>
          <ion-label>Your Bookings</ion-label>
        </ion-item>
        <ion-item lines="none">
          <ion-icon name="exit" slot="start"></ion-icon>
          <ion-label>Logout</ion-label>
        </ion-item>
      </ion-list>
    </ion-content>
  </ion-menu>
  <ion-router-outlet main></ion-router-outlet>
</ion-app>
```

```
/src/app/places/discover/discover.page.html
```

```
<ion-header>
  <ion-toolbar>
    <ion-title>Discover Places</ion-title>
    <ion-buttons slot="start">
      <ion-menu-button menu="m1"></ion-menu-button>
      <!-- OR -->
      <ion-button (click)="onOpenMenu()"><ion-icon name="menu"></ion-icon></ion-button>
    </ion-buttons>
  </ion-toolbar>
</ion-header>
```



```
/src/app/places/discover/discover.page.ts
```

```
...
import { MenuController } from '@ionic/angular';

export class DiscoverPage implements OnInit {
  ...
  constructor(private menuCtrl: MenuController) { }
  ...
  onOpenMenu(){
    this.menuCtrl.toggle('m1');
  }
}
```

 Discover Places

**Serpong M-Town**  
\$700,000,000.00



2BR apartment near Summarecon Mal Serpong.

[More](#)

 **Scientia Residence**  
Near UMN with many choices of f... >

 [Discover](#)

 [Offers](#)

 Ionic Places App

 [Discover Places](#)

 [Your Bookings](#)

 [Logout](#)



2



Questions?

Comments?

# NEXT

- Modal, Loading, & ActionSheet Controllers
- More on Ionic Grid
- Sliding Items & Virtual Scrolling

Thank you