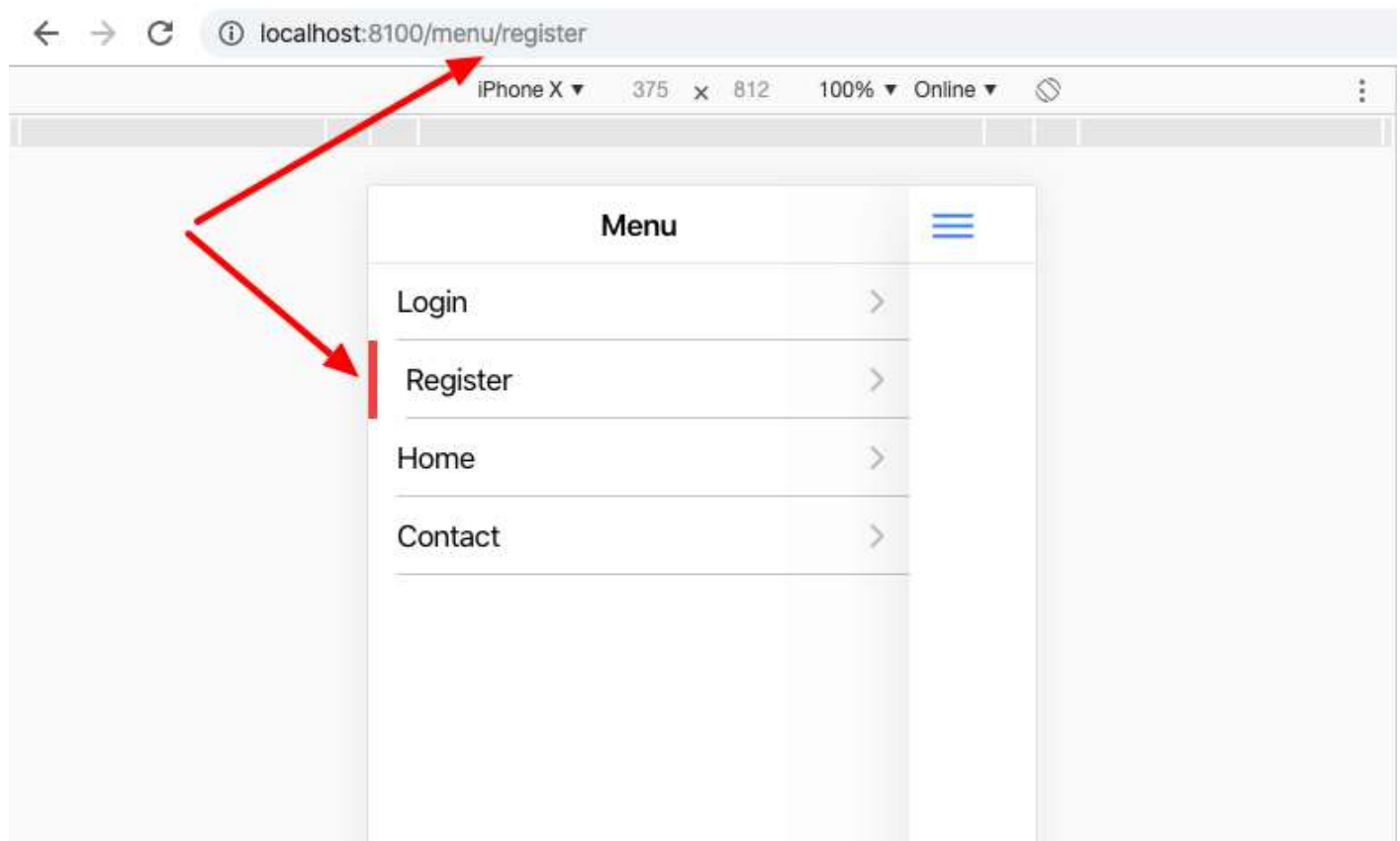


[Home](#) » [Ionic](#) » Add Dynamic Side Menu in Ionic 5 App with Active Class

Add Dynamic Side Menu in Ionic 5 App with Active Class

Last updated on **February 16, 2021** by Digamber

A comprehensive Ionic 5 step by step tutorial, Learn how to implement side menu dynamically in Ionic app with active class for selected page.



As we know, Ionic offers tons of UI components to create a Hybrid mobile application in lesser time. Most often, we fall into the situation where we need to use side menu navigation components to enhance the user experience.



The side menu looks intuitive and attractive and Implementing the side menu in Ionic is very easy. It is even considered best from the user experience's perspective.

The side menu component in the view slides in from the left, and slide position is changeable. The menu items are a direct sibling to the root content items.

We will build the menu items from an array of pages and show the pages with content area, along with that, we will dynamically set the active class for the selected url.

Table of Contents

1. Create a Blank App
2. Generate Pages
3. Set Up Routes for Side Menu
4. Creating Dynamic Side Menu in Ionic
5. Conclusion

Create a Blank App

First, Install the latest version of Ionic CLI on your device.

```
npm install -g ionic@latest
```



SPONSORED Migrate to the cloud securely and efficiently—on your own terms. [LEARN MORE](#)

```
ionic start ionic-side-menu-tabs blank --type=angular
```

Once the new Ionic app is created then get head over to the project directory.

```
cd ionic-side-menu-tabs
```

Start the side menu and tabs app in the browser.

```
ionic serve
```

Generate Pages

In this demo app, we need some pages so that we can show these pages in the menu items. We will also create the menu page and keep all the navigation-related configuration in this component.

Run the below command to generate the pages.

```
ng generate page menu  
ng generate page login  
ng generate page contact  
ng generate page register
```

Now, all the pages are generated. In the next step, we will configure routers.



some changes in AppRoutingModuleModule.

Open the `app/app-routing.module.ts` file and replace the existing code with the given below code.

```
import { NgModule } from '@angular/core';
import { PreloadAllModules, RouterModule, Routes } from '@angular/router';

const routes: Routes = [
  {
    path: '',
    loadChildren: './menu/menu.module#MenuPageModule'
  }
];

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

export class AppRoutingModule { }
```

Now, this page will send the routing request to the **MenuPageModule**.

Next, open the `menu.module.ts` page and replace with the following code.

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';
import { CommonModule } from '@angular/common';
import { FormsModule } from '@angular/forms';
import { IonicModule } from '@ionic/angular';

import { MenuPage } from './menu.page';

const routes: Routes = [
```



```

children: [
  {
    path: 'home',
    loadChildren: '../home/home.module#HomePageModule'
  },
  {
    path: 'login',
    loadChildren: '../login/login.module#LoginPageModule'
  },
  {
    path: 'register',
    loadChildren: '../register/register.module#RegisterPageModule'
  },
  {
    path: 'contact',
    loadChildren: '../contact/contact.module#ContactPageModule'
  }
]
},
{
  path: '',
  redirectTo: '/menu/home'
}
]

@NgModule({
  imports: [
    CommonModule,
    FormsModule,
    IonicModule,
    RouterModule.forChild(routes)
  ],
  declarations: [MenuPage]
})

export class MenuPageModule { }

```

Let's understand what is going on in here; we defined the main path and named it 'menu'.



...the menu items entered in the browser, then it will be connected to /menu/home page.

Creating Dynamic Side Menu in Ionic

Now, we will start implementing the side menu, open the [menu.page.ts](#) file, add the following code in it.

```
import { Component, OnInit } from '@angular/core';
import { Router, RouterEvent } from '@angular/router';

@Component({
  selector: 'app-menu',
  templateUrl: './menu.page.html',
  styleUrls: ['./menu.page.scss'],
})
export class MenuPage implements OnInit {

  activePath = '';

  pages = [
    {
      name: 'Login',
      path: '/menu/login'
    },
    {
      name: 'Register',
      path: '/menu/register'
    },
    {
      name: 'Home',
      path: '/menu/home'
    },
    {
      name: 'Contact',
      path: '/menu/contact'
    }
  ]

  constructor(private router: Router) {
```



```

    }

    ngOnInit() {
    }
  }
}

```

To show the dynamic pages in the Ionic side menu, create the array of the pages and define the Page name and path inside of it.

We defined the **activePath** variable to **show the active class for the selected route in Ionic 5**. Then, we access the events method via **`Router`** service and subscribe to the events and bind the selected route's url with the activePath variable.

Open [menu.page.html](#) and add the following code:

```

<ion-app>
  <ion-menu contentId="main-content" side="start">
    <ion-header>
      <ion-toolbar>
        <ion-title>Menu</ion-title>
      </ion-toolbar>
    </ion-header>

    <ion-content>
      <ion-list>
        <ion-menu-toggle auto-hide="false" *ngFor="let page of pages">
          <ion-item [routerLink]="page.path" routerDirection="root" [class.active-m
            {{page.name}}
          </ion-item>
        </ion-menu-toggle>
      </ion-list>
    </ion-content>
  </ion-menu>

  <ion-router-outlet id="main-content" main></ion-router-outlet>

</ion-app>

```



Now, define the **ion-menu** component to show the hamburger icon on which we click and open the side menu items list.

Inside the **ion-content**, we define the **ion-list** and **ion-menu-toggle** and bind the **ngFor** loop for showing the ionic pages dynamically in the side menu component.

The **[routerLink]** directive is the dynamic path for the associated menu item.

We also supply **routerDirection** and pass the root value to it. It is used to apply the useful screen transition animation to the navigation.

Also note that, you supply the **contentId** to the **ion-menu** directive and also pass the same id to the **ion-router-outlet**.

To show the side menu's hamburger icon on every page, you must define **ion-menu-button** in the **ion-header** directive, as given below.

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

    <!-- Side menu button -->
    <ion-buttons slot="start">
      <ion-menu-button autoHide="false"></ion-menu-button>
    </ion-buttons>

  </ion-toolbar>
</ion-header>
```

Conclusion

Finally, we have completed Ionic side menu tutorial, in this tutorial, we looked at how to implement dynamic side menu with Ionic pages, and we also learned to apply the active class for the selected route.



SPONSORED Migrate to the cloud securely and efficiently—on your own terms. [LEARN MORE](#)



www.positronx.io



Subscribe on YouTube

Follow me on Twitter

@ImDigamberSingh