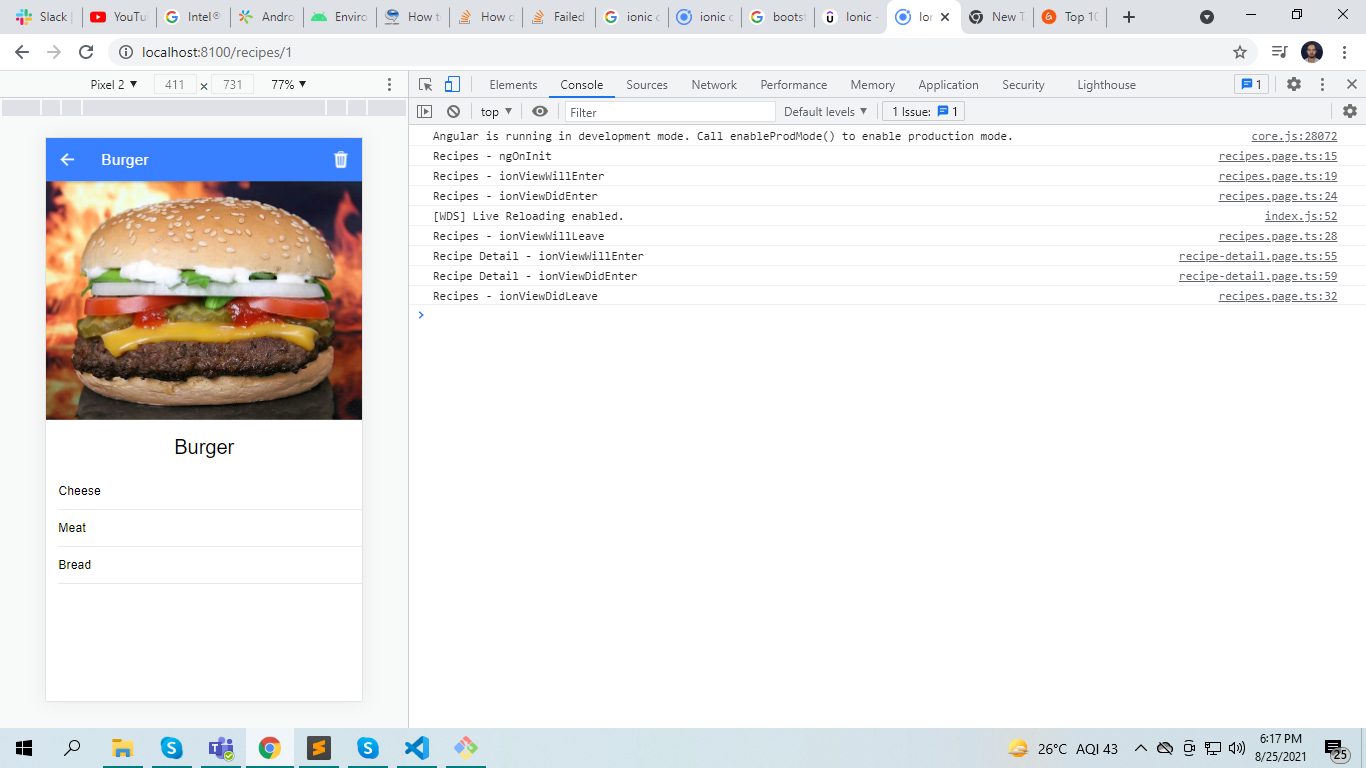
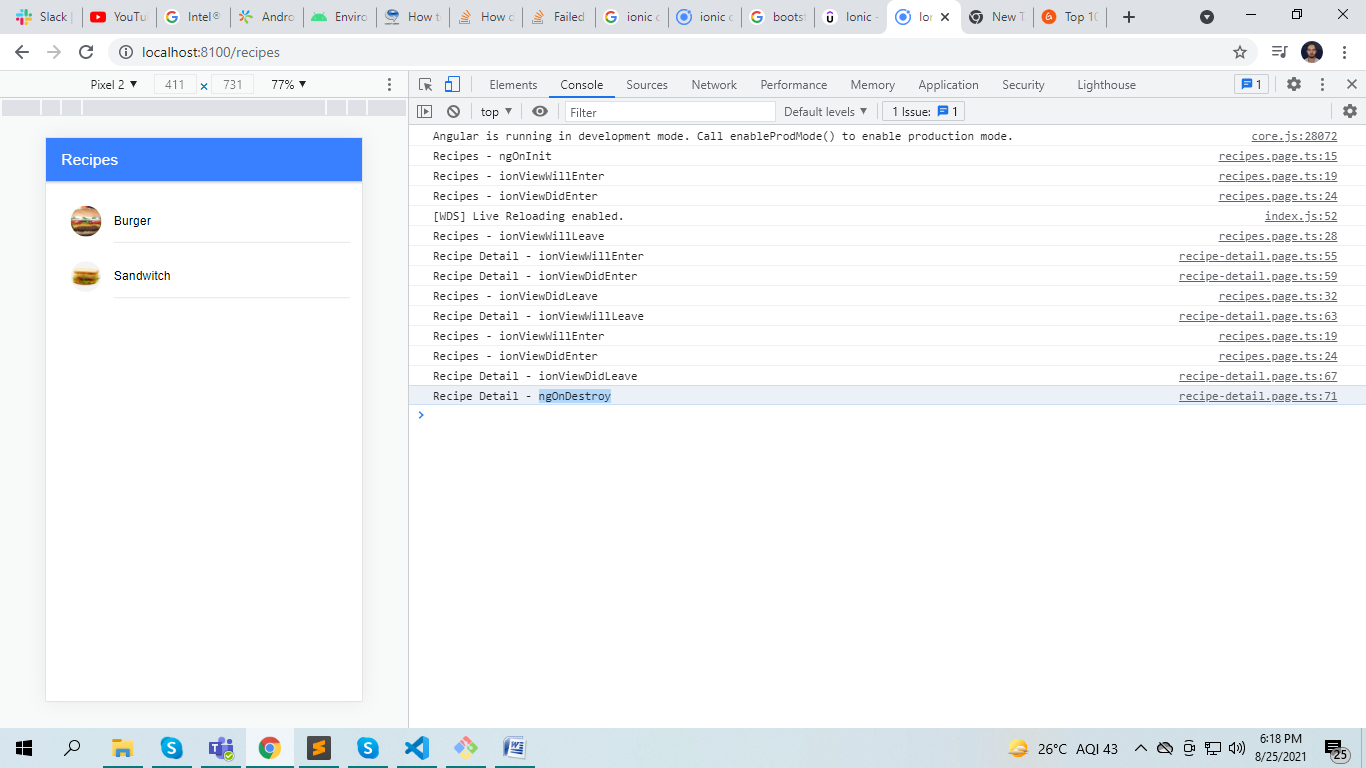
Ionic generate page recipes

It create complete module

Ionic Events





Recipe.page.ts

import { Component, OnDestroy, OnInit } from '@angular/core';

import { Recipe } from './recipe.modal';

import { RecipeService } from './recipe.service';

@Component({

  selector: 'app-recipes',

  templateUrl: './recipes.page.html',

  styleUrls: ['./recipes.page.scss'],

})

export *class* RecipesPage implements OnInit, OnDestroy {

  recipes: Recipe[] = [];

*constructor*(public *recipeService*: RecipeService) {}

  ngOnInit() {

    console.log('Recipes - ngOnInit');

  }

  ionViewWillEnter() {

    console.log('Recipes - ionViewWillEnter');

  }

  ionViewDidEnter() {

    this.recipes = this.recipeService.getAllRecipes();

    console.log('Recipes - ionViewDidEnter');

  }

  ionViewWillLeave() {

    console.log('Recipes - ionViewWillLeave');

  }

  ionViewDidLeave() {

    console.log('Recipes - ionViewDidLeave');

  }

  ngOnDestroy() {

    console.log('Recipes - ngOnDestroy');

  }

}

Recipe-detail.ts

import { Component, OnDestroy, OnInit } from '@angular/core';

import { ActivatedRoute, Router } from '@angular/router';

import { AlertController } from '@ionic/angular';

import { Recipe } from '../recipe.modal';

import { RecipeService } from '../recipe.service';

@Component({

  selector: 'app-recipe-detail',

  templateUrl: './recipe-detail.page.html',

  styleUrls: ['./recipe-detail.page.scss'],

})

export *class* RecipeDetailPage implements OnInit, OnDestroy {

*constructor*(

    private *recipeService*: RecipeService,

    private *route*: ActivatedRoute,

    private *router*: Router,

    private *alertCtrl*: AlertController

  ) {}

  loadedRecipe: Recipe = null;

  ngOnInit() {

    this.route.paramMap.subscribe((*param*) *=>* {

      if (!*param*.has('recipeId')) {

        this.router.navigate(['/recipes']);

        return;

      }

*const* recipeId = +*param*.get('recipeId');

      this.loadedRecipe = this.recipeService.getRecipe(recipeId);

    });

  }

  async deleteRecipe() {

*const* alert = await this.alertCtrl.create({

      header: 'Are you sure?',

      message: 'Once you delete it will not be revert back',

      buttons: [

        {

          text: 'Cancel',

          role: 'cancel',

        },

        {

          text: 'Yes',

          handler: () *=>* {

            this.recipeService.deleteRecipe(this.loadedRecipe.id);

            this.router.navigate(['/recipes']);

          },

        },

      ],

    });

    await alert.present();

  }

  ionViewWillEnter() {

    console.log('Recipe Detail - ionViewWillEnter');

  }

  ionViewDidEnter() {

    console.log('Recipe Detail - ionViewDidEnter');

  }

  ionViewWillLeave() {

    console.log('Recipe Detail - ionViewWillLeave');

  }

  ionViewDidLeave() {

    console.log('Recipe Detail - ionViewDidLeave');

  }

  ngOnDestroy() {

    console.log('Recipe Detail - ngOnDestroy');

  }

}

Note:

ngOnDestory of first screen is never call because Ionic is work on stack of pages.

ngOnint of other screen is never called. Only first screen ngOnint is called.

ionViewWillEnter animation start

ionViewWillLeave animation end

routing.page.html

<ion-header>

  <ion-toolbar color="primary">

    <ion-title>Recipes</ion-title>

  </ion-toolbar>

</ion-header>

<ion-content class="ion-padding">

  <ion-list

    \*ngFor="let recipe of recipes"

    [routerLink]="['/recipes',recipe.id]"

  >

    <ion-item>

      <ion-avatar slot="start">

        <img [src]="recipe.image" />

      </ion-avatar>

      <ion-label> {{recipe.name}} </ion-label>

    </ion-item>

  </ion-list>

</ion-content>

Routing-detail.page.html

<ion-header>

  <ion-toolbar color="primary">

    <ion-buttons slot="start">

      <ion-back-button defaultHref="/recipes"></ion-back-button>

    </ion-buttons>

    <ion-title>{{ loadedRecipe?.name }}</ion-title>

    <ion-buttons slot="primary">

      <ion-button (click)="deleteRecipe()">

        <ion-icon slot="icon-only" name="trash"></ion-icon>

      </ion-button>

    </ion-buttons>

  </ion-toolbar>

</ion-header>

<ion-content>

  <ion-grid class="ion-no-padding">

    <ion-row>

      <ion-col class="ion-no-padding">

        <ion-img [src]="loadedRecipe?.image"></ion-img>

      </ion-col>

    </ion-row>

    <ion-row>

      <ion-col class="ion-text-center">

        <h1>{{ loadedRecipe?.name }}</h1>

      </ion-col>

    </ion-row>

    <ion-row>

      <ion-col>

        <ion-list>

          <ion-item \*ngFor="let ig of loadedRecipe?.ingredients">

            <ion-label>{{ ig }}</ion-label>

          </ion-item>

        </ion-list>

      </ion-col>

    </ion-row>

  </ion-grid>

</ion-content>

**Navigation in Ionic**

Ionic Tabs

App-routing.module.ts

import { NgModule } from '@angular/core';

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

const routes: Routes = [

  {

    path: '',

    redirectTo: 'places',

    pathMatch: 'full',

  },

  {

    path: 'auth',

    loadChildren: () =>

      import('./auth/auth.module').then((m) => m.AuthPageModule),

  },

  {

    path: 'places',

    loadChildren: () =>

      import('./places/places.module').then((m) => m.PlacesPageModule),

  },

  {

    path: 'bookings',

    loadChildren: () =>

      import('./bookings/bookings.module').then((m) => m.BookingsPageModule),

  },

];

@NgModule({

  imports: [

    RouterModule.forRoot(routes, { preloadingStrategy: PreloadAllModules }),

  ],

  exports: [RouterModule],

})

export class AppRoutingModule {}

app.component.html

<ion-app>

  <ion-router-outlet></ion-router-outlet>

</ion-app>

Places-routing.module.ts

import { NgModule } from '@angular/core';

import { Routes, RouterModule } from '@angular/router';

import { PlacesPage } from './places.page';

const routes: Routes = [

  {

    path: 'tabs',

    component: PlacesPage, //define tabs code here

    children: [

      {

        path: 'discover',

        children: [

          {

            path: '',

            loadChildren: () =>

              import('./discover/discover.module').then(

                (m) => m.DiscoverPageModule

              ),

          },

          {

            path: ':placeId',

            loadChildren: () =>

              import('./discover/place-detail/place-detail.module').then(

                (m) => m.PlaceDetailPageModule

              ),

          },

        ],

      },

      {

        path: 'offers',

        children: [

          {

            path: '',

            loadChildren: () =>

              import('./offers/offers.module').then((m) => m.OffersPageModule),

          },

          {

            path: 'new',

            loadChildren: () =>

              import('./offers/new-offer/new-offer.module').then(

                (m) => m.NewOfferPageModule

              ),

          },

          {

            path: 'edit/:placeId',

            loadChildren: () =>

              import('./offers/edit-offer/edit-offer.module').then(

                (m) => m.EditOfferPageModule

              ),

          },

          {

            path: ':placeId',

            loadChildren: () =>

              import('./offers/offer-bookings/offer-bookings.module').then(

                (m) => m.OfferBookingsPageModule

              ),

          },

        ],

      },

      {

        path: '',

        redirectTo: '/places/tabs/discover',

        pathMatch: 'full',

      },

    ],

  },

  {

    path: '',

    redirectTo: '/places/tabs/discover',

    pathMatch: 'full',

  },

];

@NgModule({

  imports: [RouterModule.forChild(routes)],

  exports: [RouterModule],

})

export class PlacesPageRoutingModule {}

Note: Put all the dynamic route parameter below:

path: ':placeId',

places.page.html

<ion-tabs>

  <ion-tab-bar slot="bottom">

    <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>

Routing animation

routerDirection = “backward/forward” (put in navigation button and component)

by default ionic handle all the animation

Back buttons

<ion-header>

  <ion-toolbar>

    <ion-buttons slot="start">

      <ion-back-button defaultHref="/places/tabs/discover"></ion-back-button>

    </ion-buttons>

    <ion-title>Place Detail</ion-title>

  </ion-toolbar>

</ion-header>

defaultHref is used in case when page refresh, Ionic do not determine where we have to go.

Navigate programmatically

import { Component, OnInit } from '@angular/core';

import { Router } from '@angular/router';

import { NavController } from '@ionic/angular';

@Component({

  selector: 'app-place-detail',

  templateUrl: './place-detail.page.html',

  styleUrls: ['./place-detail.page.scss'],

})

export class PlaceDetailPage implements OnInit {

  constructor(private router: Router, private navCtrl: NavController) {}

  ngOnInit() {}

  onBookPlace() {

    //this.router.navigateByUrl('/places/tabs/discover');

    //this.router.navigate(['/', 'places', 'tabs', 'discover']);

    this.navCtrl.navigateBack('/places/tabs/discover'); //give animation effect

    //this.navCtrl.pop(); // it will not work when page is refresh

  }

}

Add button in toolbar

Slot = “primary” means add buttons at the end the of toolbar

<ion-header>

  <ion-toolbar>

    <ion-title>My offer</ion-title>

    <ion-buttons slot="primary">

      <ion-button [routerLink]="['/','places','tabs','offers','new']">

        <ion-icon slot="icon-only" name="add"></ion-icon>

      </ion-button>

    </ion-buttons>

  </ion-toolbar>

</ion-header>

Side drawer

App.component.html

<ion-app>

  <ion-menu side="start">

    <ion-header>

      <ion-toolbar>

        <ion-title>PairPNB</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>

Offers.page.html

<ion-header>

  <ion-toolbar>

    <ion-buttons slot="start">

      <ion-menu-button></ion-menu-button>

    </ion-buttons>

    <ion-title>My offer</ion-title>

    <ion-buttons slot="primary">

      <ion-button [routerLink]="['/','places','tabs','offers','new']">

        <ion-icon slot="icon-only" name="add"></ion-icon>

      </ion-button>

    </ion-buttons>

  </ion-toolbar>

</ion-header>

<ion-content>

  <ion-grid fixed>

    <ion-row>

      <ion-col size="12" size-sm="8" offset-sm="2">

        <ion-list>

          <ion-item

            \*ngFor="let offer of offers"

            [routerLink]="['/','places','tabs','offers',offer.id]"

          >

            <ion-thumbnail slot="start">

              <img [src]="offer.imageUrl" />

            </ion-thumbnail>

            <ion-label>

              <h2>{{offer.title}}</h2>

            </ion-label>

          </ion-item>

        </ion-list>

      </ion-col>

    </ion-row>

  </ion-grid>

</ion-content>

If in case multiple menu is used then menuId is used

<ion-app>

  <ion-menu side="start">

    <ion-header>

      <ion-toolbar>

        <ion-title>PairPNB</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-menu side="start" menuId="m1">

    <ion-header>

      <ion-toolbar>

        <ion-title>PairPNB1</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>

 <ion-buttons slot="start">

      <ion-menu-button menu="m1"></ion-menu-button>

</ion-buttons>

Programmatically open menu

 onOpenMenu() {

    //this.menuCtrl.toggle();

    //in case multiple menu; menuId is used

    //this.menuCtrl.open('m1');

  }

Auto Close Sidedrawer on menu item click ion-menu-toggle wrapper is used

<ion-menu-toggle> menu item goes here </ion-menu-toggle>

<ion-menu-toggle menu="m1"> menu item goes here </ion-menu-toggle> //to close particular menu

App.component.ts

<ion-app>

  <ion-menu side="start">

    <ion-header>

      <ion-toolbar>

        <ion-title>PairPNB</ion-title>

      </ion-toolbar>

    </ion-header>

    <ion-content>

      <ion-list>

        <ion-menu-toggle>

          <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>

          <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>

          <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>

  </ion-menu>

  <ion-router-outlet main></ion-router-outlet>

</ion-app>

Fake authentication

canLoad : It does not allow to load the entire module.

canActivate : it just protect route in application.