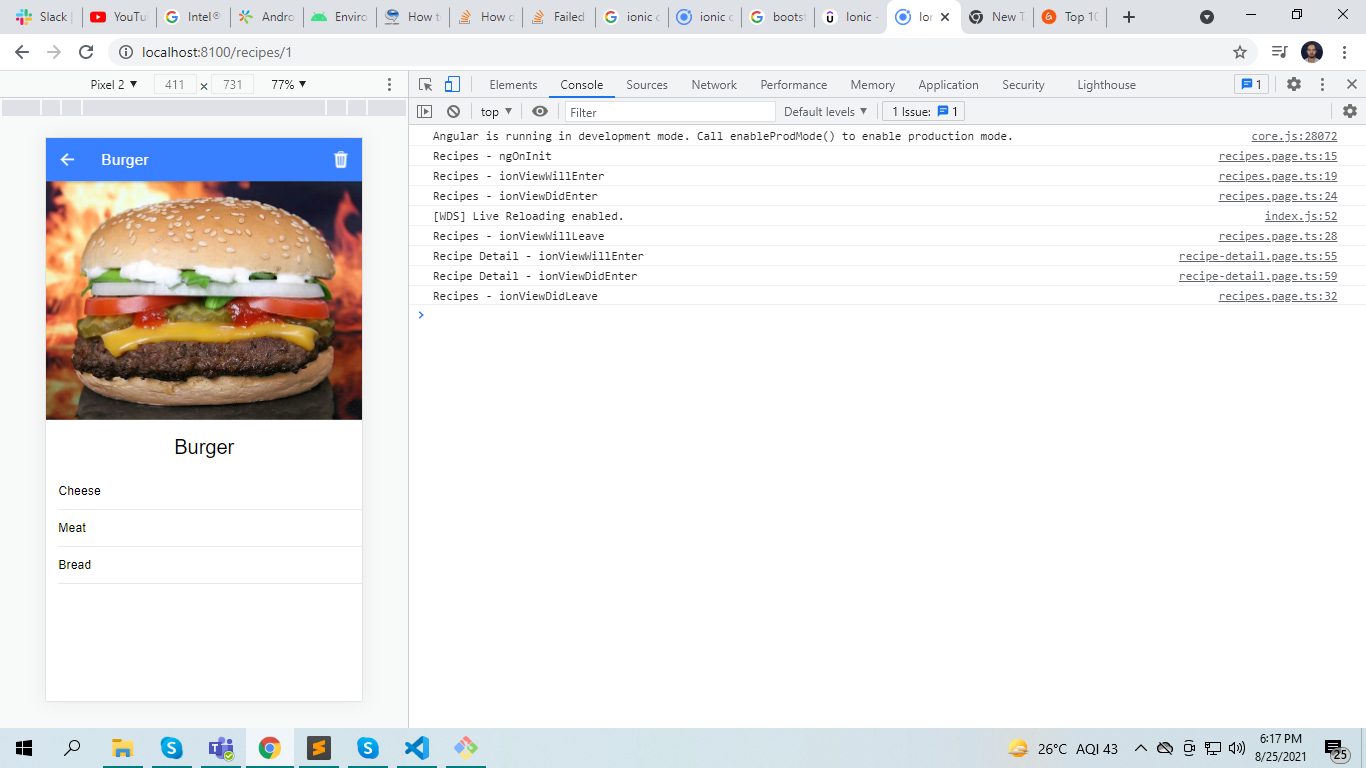
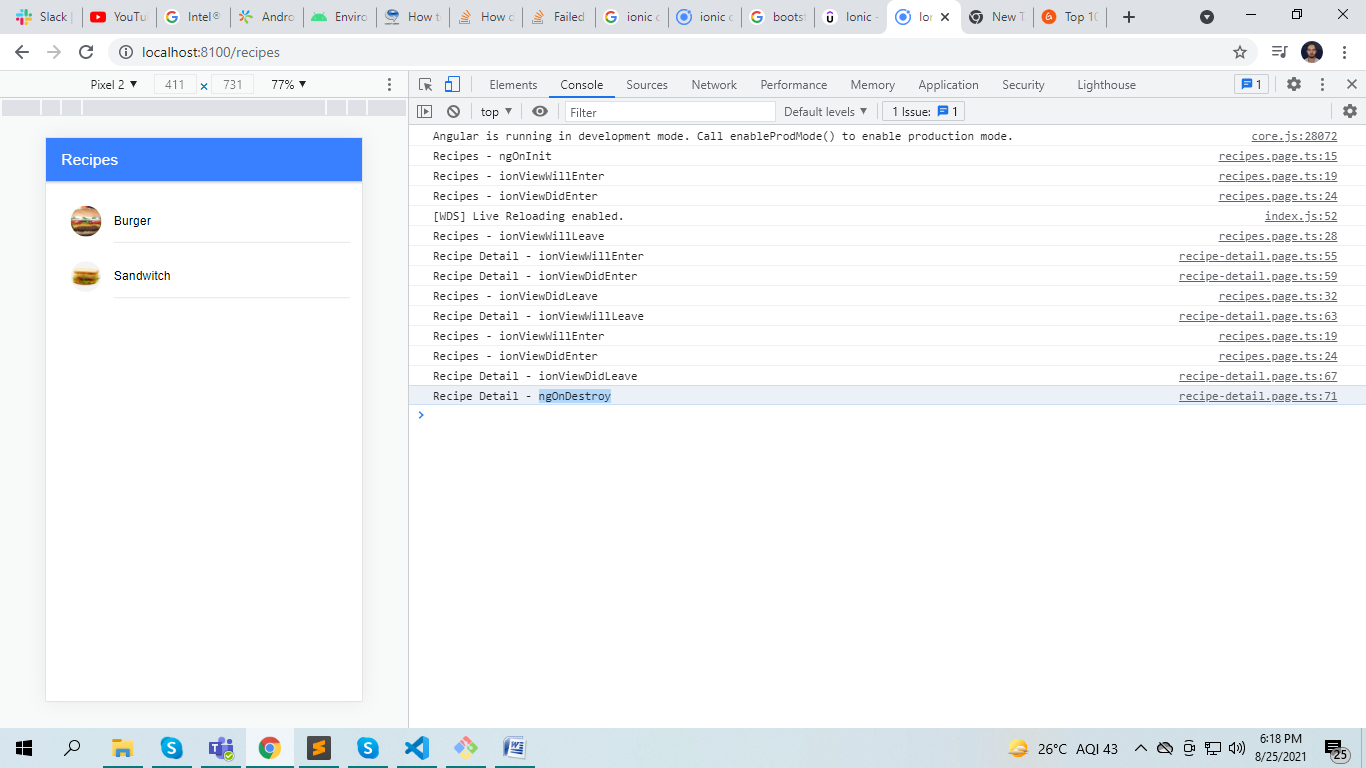
Ionic generate page recipes

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

//this.navCtrl.navigateForward('/places/tabs/discover')

  }

}

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.

Modals in Ionic

Create component

Ionic generate component create-booking

Create-booking.ts

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

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

import { Place } from '../../places/place.model';

@Component({

  selector: 'app-create-booking',

  templateUrl: './create-booking.component.html',

  styleUrls: ['./create-booking.component.scss'],

})

export *class* CreateBookingComponent implements OnInit {

  @Input() selectedPlace: Place;

*constructor*(private *modalCtrl*: ModalController) {}

  ngOnInit() {}

  onCancel() {

    //In case mulitple modal, close particular modal with modal id as third argument

    //this.modalCtrl.dismiss(null, 'cancel','modal-ID');

    this.modalCtrl.dismiss(null, 'cancel');

  }

  onBookPlace() {

    this.modalCtrl.dismiss(

      {

        message: 'This is a dummy message!',

      },

      'confirm'

    );

  }

}

Create booking html

<ion-header>

  <ion-toolbar>

    <ion-title class="ion-text-center">{{ selectedPlace.title }}</ion-title>

    <ion-buttons slot="primary">

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

        <ion-icon  name="close"></ion-icon>

      </ion-button>

    </ion-buttons>

  </ion-toolbar>

</ion-header>

<ion-content class="ion-padding ion-text-center">

  <p> {{ selectedPlace.description }} </p>

  <ion-button (click)="onBookPlace()" color="primary">

    Book!

  </ion-button>

</ion-content>

Open Modal at Place detail page.ts

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

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

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

import { CreateBookingComponent } from 'src/app/bookings/create-booking/create-booking.component';

import { Place } from '../../place.model';

import { PlacesService } from '../../places.service';

@Component({

  selector: 'app-place-detail',

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

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

})

export *class* PlaceDetailPage implements OnInit {

*constructor*(

    private *route*: ActivatedRoute,

    private *navCtrl*: NavController,

    private *modalCtrl*: ModalController,

    private *placesService*: PlacesService

  ) {}

  place: Place;

  ngOnInit() {

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

      if (!*paramMap*.has('placeId')) {

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

        return;

      }

      this.place = this.placesService.getPlace(*paramMap*.get('placeId'));

    });

  }

  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

    //In case mulitple modal give modal id at the time of open

    this.modalCtrl

      .create({

        component: CreateBookingComponent,

        componentProps: {

          selectedPlace: this.place,

        },

        //id:'modal-ID'

      })

      .then((*modalEl*) *=>* {

*modalEl*.present();

        return *modalEl*.onDidDismiss();

      })

      .then((*resultData*) *=>* {

        console.log(*resultData*.data, *resultData*.role);

        if (*resultData*.role === 'confirm') {

          console.log('BOOKED!');

        }

      });

  }

}

Page detail HTML

<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>

<ion-content>

  <ion-button (click)="onBookPlace()" class="ion-margin" color="primary">

    Book

  </ion-button>

</ion-content>

Page detail module ts

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

import { CommonModule } from '@angular/common';

import { FormsModule } from '@angular/forms';

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

import { PlaceDetailPageRoutingModule } from './place-detail-routing.module';

import { PlaceDetailPage } from './place-detail.page';

import { CreateBookingComponent } from '../../../bookings/create-booking/create-booking.component';

@NgModule({

  imports: [

    CommonModule,

    FormsModule,

    IonicModule,

    PlaceDetailPageRoutingModule,

  ],

  declarations: [PlaceDetailPage, CreateBookingComponent],

  entryComponents: [CreateBookingComponent], //don’t forget to add component at enteryComponents

})

export *class* PlaceDetailPageModule {}

Ion grid

<ion-header>

  <ion-toolbar>

    <ion-buttons slot="start">

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

    </ion-buttons>

    <ion-title>New offer</ion-title>

  </ion-toolbar>

</ion-header>

<ion-content >

  <!-- <ion-grid class="ion-no-padding"> -->

  <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 3

      </ion-col>

      <ion-col>

        Row 1 Col 4

      </ion-col>

      <ion-col>

        Row 1 Col 5

      </ion-col>

      <ion-col>

        Row 1 Col 6

      </ion-col>

      <ion-col>

        Row 1 Col 7

      </ion-col>

      <ion-col>

        Row 1 Col 8

      </ion-col>

      <ion-col>

        Row 1 Col 9

      </ion-col>

      <ion-col>

        Row 1 Col 10

      </ion-col>

      <ion-col>

        Row 1 Col 11

      </ion-col>

      <ion-col>

        Row 1 Col 12

      </ion-col>

    </ion-row>

  </ion-grid>

</ion-content>

<ion-header>

  <ion-toolbar>

    <ion-buttons slot="start">

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

    </ion-buttons>

    <ion-title>New offer</ion-title>

  </ion-toolbar>

</ion-header>

<ion-content >

  <!-- <ion-grid class="ion-no-padding"> -->

  <ion-grid>

    <!-- class="ion-align-items-start" -->

    <ion-row class="ion-justify-content-center" >

      <ion-col size="4">

        Row 1 Col 1

      </ion-col>

      <ion-col size="4">

        <ion-card>

          <ion-card-header>

            <ion-card-title>Awesome Title</ion-card-title>

          </ion-card-header>

          <ion-card-content>

            <p>Super description</p>

          </ion-card-content>

        </ion-card>

      </ion-col>

    </ion-row>

  </ion-grid>

</ion-content>

Offset

<ion-content>

  <ion-grid>

    <ion-row>

      <ion-col size="4" offset="2">Row 1 col 1 </ion-col>

      <ion-col size="4">Row 1 col 1</ion-col>

    </ion-row>

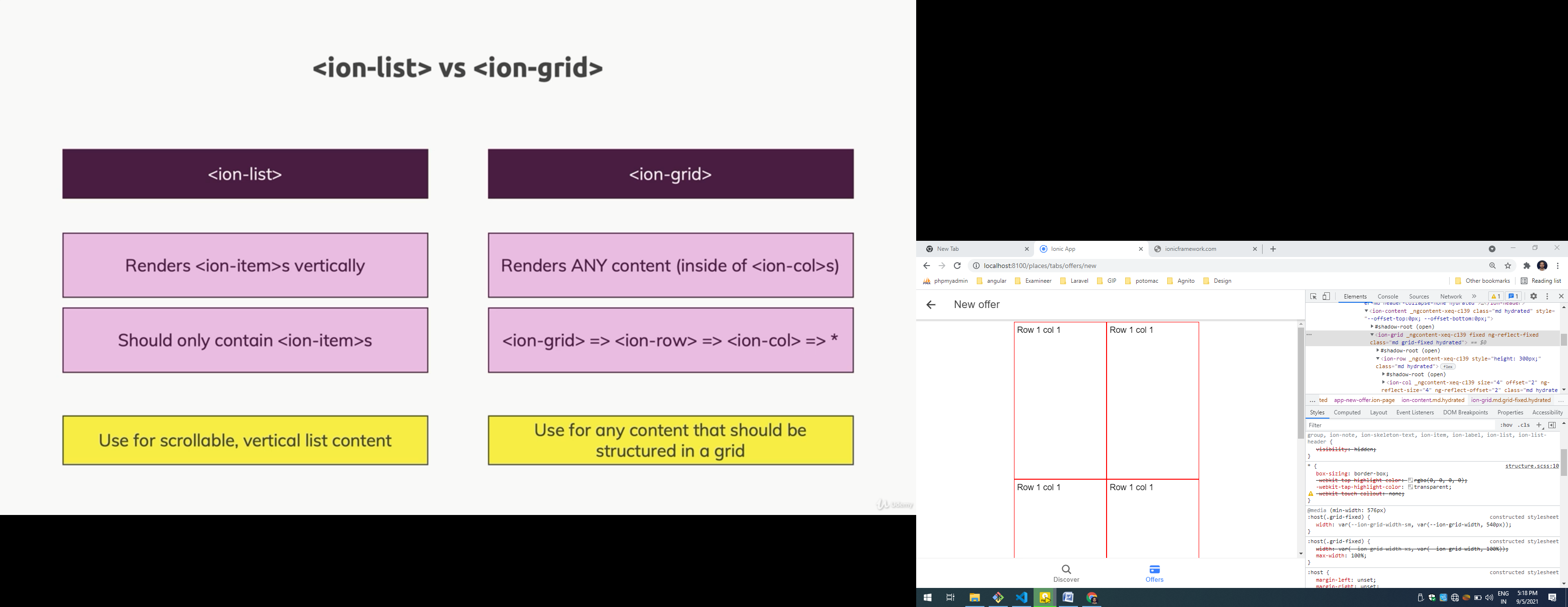
  </ion-grid>

</ion-content>

Ion grid with fixed width

<ion-grid fixed>

</ion-grid>



Ion-item

It is generally used with ion-list. But it can also used as standalone element. It provides different effect to ion-label when used with ion-input element. Floating effect of ion-lable cannot work without ion-item.

<ion-header>

  <ion-toolbar>

    <ion-title>Authentication</ion-title>

  </ion-toolbar>

</ion-header>

<ion-content class="ion-padding">

  <ion-item>

    <ion-label position="floating">E-Mail</ion-label>

    <ion-input type="text" placeholder="Enter your email"></ion-input>

  </ion-item>

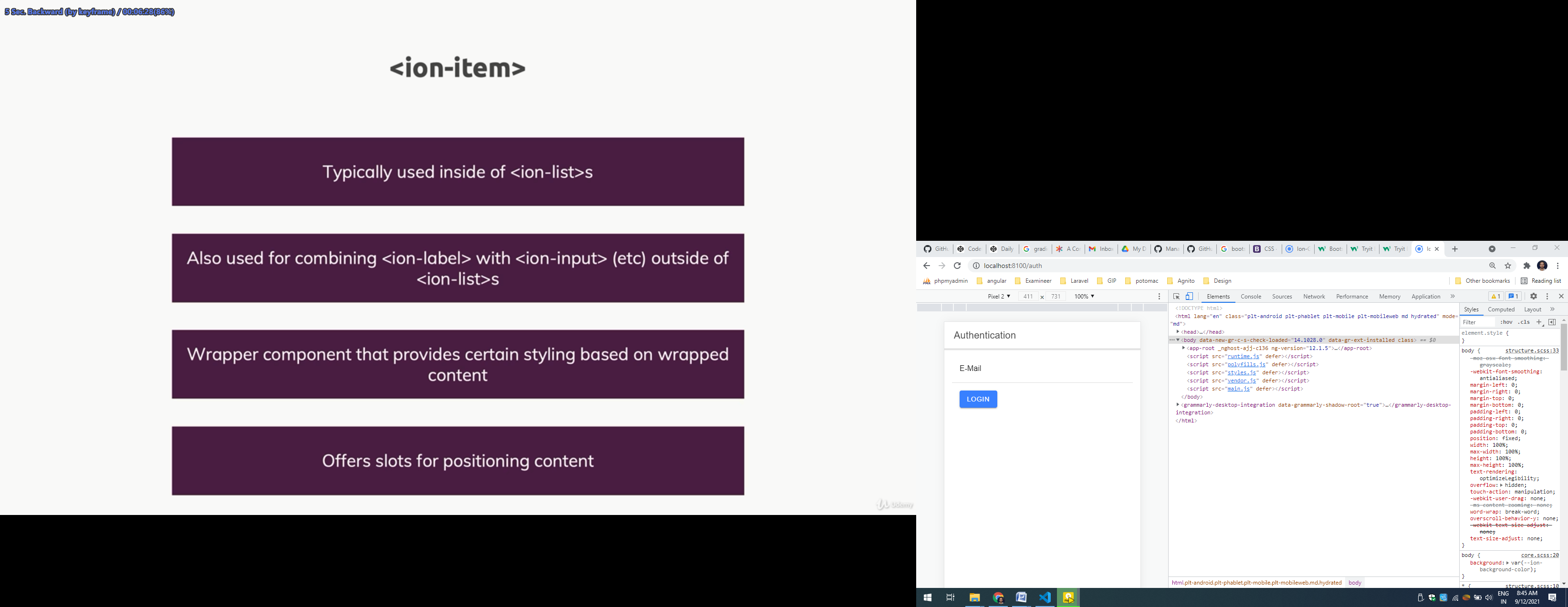
  <ion-button (click)="onLogin()" color="primary" class="ion-margin">

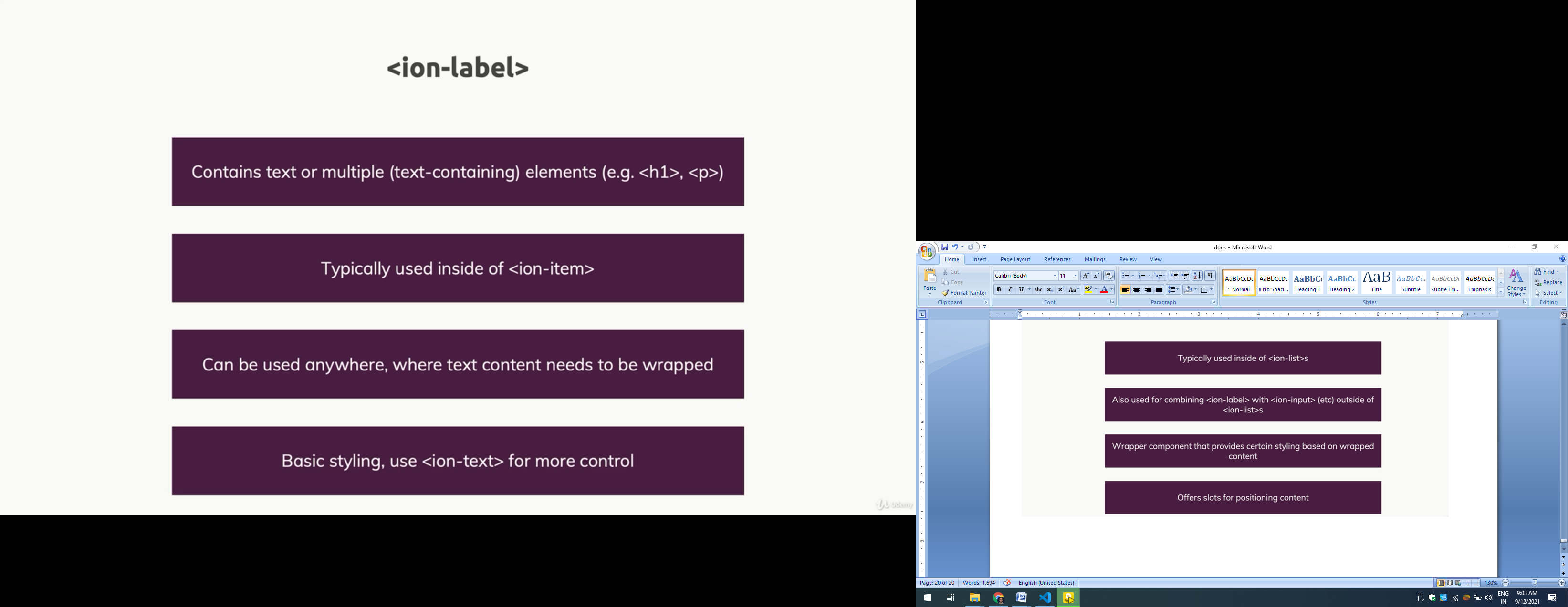
    Login

  </ion-button>

</ion-content>







If we have multiple text inside the ion-item use ion-lable

Ion-text

<p>

   <ion-text color="danger">

     {{ loadedPlaces[0].description }}

   </ion-text>

 </p>

Swipe able List Item

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-sliding \*ngFor="let offer of offers" #slidingItem>

            <ion-item-options side="start">

              <ion-item-option

                color="primary"

                (click)="onDelete(offer.id,slidingItem)"

                >Delete</ion-item-option

              >

            </ion-item-options>

            <ion-item-options side="end">

              <ion-item-option

                color="primary"

                (click)="onEdit(offer.id,slidingItem)"

              >

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

              </ion-item-option>

            </ion-item-options>

            <ion-item [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-item-sliding>

        </ion-list>

      </ion-col>

    </ion-row>

  </ion-grid>

</ion-content>

Offers.page.ts

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

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

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

import { Place } from '../place.model';

import { PlacesService } from '../places.service';

@Component({

  selector: 'app-offers',

  templateUrl: './offers.page.html',

  styleUrls: ['./offers.page.scss'],

})

export class OffersPage implements OnInit {

  offers: Place[] = [];

  constructor(private placesService: PlacesService, private router: Router) {}

  ngOnInit() {

    this.offers = this.placesService.places;

  }

  onDelete(offerId: string, slide: IonItemSliding) {

    console.log(`Offer with id ${offerId}`);

    slide.close();

  }

  onEdit(offerId: string, slide: IonItemSliding) {

    console.log(`Offer with id ${offerId}`);

    //this.router.navigateByUrl(`/places/tabs/offers/${offerId}`);

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

    slide.close();

  }

}

Side drawer

Main attribute is deprecated

<ion-app>

  <ion-menu side="start" menuId="mainMenu" contentId="main">

    <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-menu side="start" menuId="authMenu" contentId="main">

    <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>Register</ion-label>

          </ion-item>

        </ion-menu-toggle>

      </ion-list>

    </ion-content>

  </ion-menu>

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

</ion-app>

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

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

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

import { AuthService } from './auth/auth.service';

@Component({

  selector: 'app-root',

  templateUrl: 'app.component.html',

  styleUrls: ['app.component.scss'],

})

export class AppComponent implements OnInit {

  constructor(

    private authService: AuthService,

    private router: Router,

    private menu: MenuController

  ) {}

  ngOnInit() {

    this.menu.enable(true, 'mainMenu');

  }

  onLogout() {

    this.authService.onLoggout();

    this.router.navigateByUrl('/auth');

  }

}



<ion-header>

  <ion-toolbar>

    <ion-buttons slot="start">

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

    </ion-buttons>

    <ion-title>Discover Places</ion-title>

  </ion-toolbar>

</ion-header>

<ion-content class="ion-padding">

  <ion-grid fixed>

    <ion-row>

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

        <ion-card>

          <ion-card-header>

            <ion-card-title> {{loadedPlaces[0].title}} </ion-card-title>

            <ion-card-subtitle>

              {{ loadedPlaces[0].price | currency }}/ Night

            </ion-card-subtitle>

          </ion-card-header>

          <ion-img [src]="loadedPlaces[0].imageUrl"></ion-img>

          <ion-card-content>

            <p>{{ loadedPlaces[0].description }}</p>

            <ion-button

              [routerLink]="['/','places','tabs','discover',loadedPlaces[0].id]"

              fill="clear"

              color="primary"

            >

              More

            </ion-button>

          </ion-card-content>

        </ion-card>

      </ion-col>

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

        <ion-virtual-scroll

          [items]="listedLoadedPlaces"

          approxItemHeight="70px"

        >

          <ion-item

            \*virtualItem="let place"

            [routerLink]="['/','places','tabs','discover',place.id]"

            detail

          >

            <ion-thumbnail slot="start">

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

            </ion-thumbnail>

            <ion-label>

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

              <p>{{ place.description }}</p>

            </ion-label>

          </ion-item>

        </ion-virtual-scroll>

      </ion-col>

    </ion-row>

  </ion-grid>

</ion-content>

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

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

import { Place } from '../place.model';

import { PlacesService } from '../places.service';

@Component({

  selector: 'app-discover',

  templateUrl: './discover.page.html',

  styleUrls: ['./discover.page.scss'],

})

export class DiscoverPage implements OnInit {

  loadedPlaces: Place[];

  listedLoadedPlaces: Place[];

  constructor(

    private placeService: PlacesService,

    private menuCtrl: MenuController

  ) {}

  ngOnInit() {

    this.loadedPlaces = this.placeService.places;

    this.listedLoadedPlaces = this.loadedPlaces.slice(1);

  }

  onOpenMenu() {

    //this.menuCtrl.toggle();

    //in case multiple menu; menuId is used

    //this.menuCtrl.open('m1');

  }

}

Adding Images

Ion-image load image lazily.

Ion-thumbnail(squre) and ion-avatar(rounded) is image wrapper

Segmented Buttons

Discover.page.html

<ion-header>

  <ion-toolbar>

    <ion-buttons slot="start">

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

    </ion-buttons>

    <ion-title>Discover Places</ion-title>

  </ion-toolbar>

</ion-header>

<ion-content class="ion-padding">

<!—-Ion segment -->

  <ion-segment value="all" (ionChange)="onFilterUpdate($event)">

    <ion-segment-button value="all">

      <ion-label>All Places</ion-label>

    </ion-segment-button>

    <ion-segment-button value="bookable">

      <ion-label>Bookable Places</ion-label>

    </ion-segment-button>

  </ion-segment>

  <ion-grid>

    <ion-row>

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

        <ion-card>

          <ion-card-header>

            <ion-card-title> {{loadedPlaces[0].title}} </ion-card-title>

            <ion-card-subtitle>

              {{ loadedPlaces[0].price | currency }}/ Night

            </ion-card-subtitle>

          </ion-card-header>

          <ion-img [src]="loadedPlaces[0].imageUrl"></ion-img>

          <ion-card-content>

            <p>{{ loadedPlaces[0].description }}</p>

            <ion-button

              [routerLink]="['/','places','tabs','discover',loadedPlaces[0].id]"

              fill="clear"

              color="primary"

            >

              More

            </ion-button>

          </ion-card-content>

        </ion-card>

      </ion-col>

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

        <ion-virtual-scroll

          [items]="listedLoadedPlaces"

          approxItemHeight="70px"

        >

          <ion-item

            \*virtualItem="let place"

            [routerLink]="['/','places','tabs','discover',place.id]"

            detail

          >

            <ion-thumbnail slot="start">

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

            </ion-thumbnail>

            <ion-label>

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

              <p>{{ place.description }}</p>

            </ion-label>

          </ion-item>

        </ion-virtual-scroll>

      </ion-col>

    </ion-row>

  </ion-grid>

</ion-content>

Discover.page.ts

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

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

import { SegmentChangeEventDetail } from '@ionic/core';

import { Place } from '../place.model';

import { PlacesService } from '../places.service';

@Component({

  selector: 'app-discover',

  templateUrl: './discover.page.html',

  styleUrls: ['./discover.page.scss'],

})

export class DiscoverPage implements OnInit {

  loadedPlaces: Place[];

  listedLoadedPlaces: Place[];

  constructor(

    private placeService: PlacesService,

    private menuCtrl: MenuController

  ) {}

  ngOnInit() {

    this.loadedPlaces = this.placeService.places;

    this.listedLoadedPlaces = this.loadedPlaces.slice(1);

  }

  onOpenMenu() {

    //this.menuCtrl.toggle();

    //in case multiple menu; menuId is used

    //this.menuCtrl.open('m1');

  }

  onFilterUpdate(event: CustomEvent<SegmentChangeEventDetail>) {

    console.log(event.detail);

  }

}

Adding Spinner

Auth.page.html

<ion-header>

  <ion-toolbar>

    <ion-title>Authentication</ion-title>

  </ion-toolbar>

</ion-header>

<ion-content class="ion-padding">

  <ion-list>

    <ion-item>

      <ion-label position="floating">E-Mail</ion-label>

      <ion-input type="text" placeholder="Enter your email"></ion-input>

    </ion-item>

  </ion-list>

  <div class="ion-text-center ion-margin" \*ngIf="isLoading">

    <ion-spinner color="primary"></ion-spinner>

  </div>

  <ion-button

    \*ngIf="!isLoading"

    (click)="onLogin()"

    color="primary"

    class="ion-margin"

  >

    Login

  </ion-button>

</ion-content>

Auth.page.ts

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

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

import { AuthService } from './auth.service';

@Component({

  selector: 'app-auth',

  templateUrl: './auth.page.html',

  styleUrls: ['./auth.page.scss'],

})

export class AuthPage implements OnInit {

  isLoading = false;

  constructor(private authService: AuthService, private router: Router) {}

  ngOnInit() {}

  onLogin() {

    this.isLoading = true;

    this.authService.onLogin();

    setTimeout(() => {

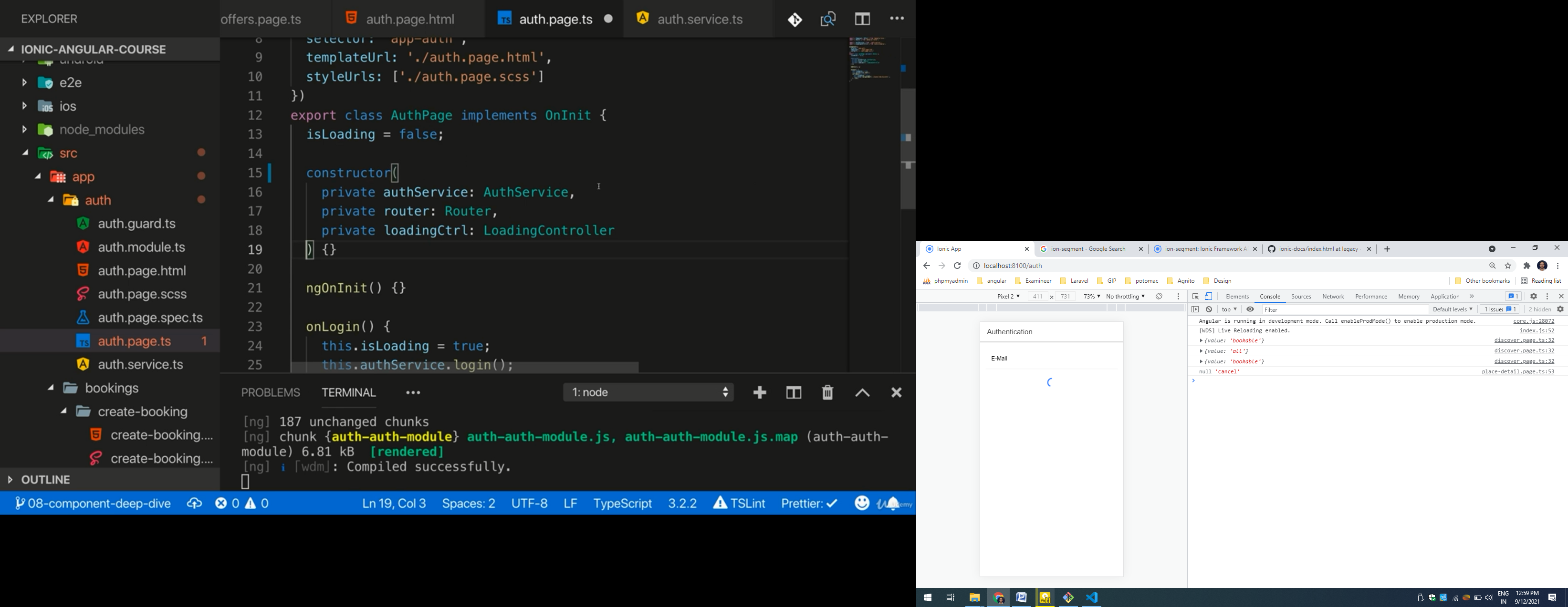
      this.isLoading = false;

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

    }, 1500);

  }

}



Loading Controller

Auth.page.html

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

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

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

import { AuthService } from './auth.service';

@Component({

  selector: 'app-auth',

  templateUrl: './auth.page.html',

  styleUrls: ['./auth.page.scss'],

})

export class AuthPage implements OnInit {

  constructor(

    private authService: AuthService,

    private router: Router,

    private loadingCtrl: LoadingController

  ) {}

  ngOnInit() {}

  async onLogin() {

    this.authService.onLogin();

    const loadCtrl = await this.loadingCtrl.create({

      keyboardClose: true,

      message: 'Logging in...',

    });

    await loadCtrl.present();

    setTimeout(() => {

      loadCtrl.dismiss();

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

    }, 1500);

  }

}

Auth.page.html

<ion-header>

  <ion-toolbar>

    <ion-title>Authentication</ion-title>

  </ion-toolbar>

</ion-header>

<ion-content class="ion-padding">

  <ion-list>

    <ion-item>

      <ion-label position="floating">E-Mail</ion-label>

      <ion-input type="text" placeholder="Enter your email"></ion-input>

    </ion-item>

  </ion-list>

  <ion-button (click)="onLogin()" color="primary" class="ion-margin">

    Login

  </ion-button>

</ion-content>

ActionSheet Controller

Set of options that drive from the bottom of UI

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

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

import {

  ModalController,

  NavController,

  ActionSheetController,

} from '@ionic/angular';

import { CreateBookingComponent } from 'src/app/bookings/create-booking/create-booking.component';

import { Place } from '../../place.model';

import { PlacesService } from '../../places.service';

@Component({

  selector: 'app-place-detail',

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

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

})

export class PlaceDetailPage implements OnInit {

  constructor(

    private route: ActivatedRoute,

    private navCtrl: NavController,

    private modalCtrl: ModalController,

    private placesService: PlacesService,

    private actionSheetCtrl: ActionSheetController

  ) {}

  place: Place;

  ngOnInit() {

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

      if (!paramMap.has('placeId')) {

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

        return;

      }

      this.place = this.placesService.getPlace(paramMap.get('placeId'));

    });

  }

  // 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

  //   //In case mulitple modal give modal id at the time of open

  //   this.modalCtrl

  //     .create({

  //       component: CreateBookingComponent,

  //       componentProps: {

  //         selectedPlace: this.place,

  //       },

  //       //id:'modal-ID'

  //     })

  //     .then((modalEl) => {

  //       modalEl.present();

  //       return modalEl.onDidDismiss();

  //     })

  //     .then((resultData) => {

  //       console.log(resultData.data, resultData.role);

  //       if (resultData.role === 'confirm') {

  //         console.log('BOOKED!');

  //       }

  //     });

  // }

  async onBookPlace() {

    const actionSheet = await this.actionSheetCtrl.create({

      header: 'Choose an Action',

      cssClass: 'my-custom-class',

      buttons: [

        {

          text: 'Delete',

          role: 'destructive', //red color in IOS

          icon: 'trash',

          handler: () => {

            console.log('Delete clicked');

          },

        },

        {

          text: 'Share',

          icon: 'share',

          handler: () => {

            console.log('Share clicked');

          },

        },

        {

          text: 'Play (open modal)',

          icon: 'caret-forward-circle',

          handler: () => {

            console.log('Play clicked');

          },

        },

        {

          text: 'Favorite',

          icon: 'heart',

          handler: () => {

            console.log('Favorite clicked');

          },

        },

        {

          text: 'Cancel',

          icon: 'close',

          role: 'cancel', //to close action sheet

          handler: () => {

            console.log('Cancel clicked');

          },

        },

      ],

    });

    await actionSheet.present();

    const { role } = await actionSheet.onDidDismiss();

    console.log('onDidDismiss resolved with role', role);

  }

}

Place-detail.page.ts

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

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

import {

  ModalController,

  NavController,

  ActionSheetController,

} from '@ionic/angular';

import { CreateBookingComponent } from 'src/app/bookings/create-booking/create-booking.component';

import { Place } from '../../place.model';

import { PlacesService } from '../../places.service';

@Component({

  selector: 'app-place-detail',

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

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

})

export class PlaceDetailPage implements OnInit {

  constructor(

    private route: ActivatedRoute,

    private navCtrl: NavController,

    private modalCtrl: ModalController,

    private placesService: PlacesService,

    private actionSheetCtrl: ActionSheetController

  ) {}

  place: Place;

  ngOnInit() {

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

      if (!paramMap.has('placeId')) {

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

        return;

      }

      this.place = this.placesService.getPlace(paramMap.get('placeId'));

    });

  }

  // 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

  //   //In case mulitple modal give modal id at the time of open

  //   this.modalCtrl

  //     .create({

  //       component: CreateBookingComponent,

  //       componentProps: {

  //         selectedPlace: this.place,

  //       },

  //       //id:'modal-ID'

  //     })

  //     .then((modalEl) => {

  //       modalEl.present();

  //       return modalEl.onDidDismiss();

  //     })

  //     .then((resultData) => {

  //       console.log(resultData.data, resultData.role);

  //       if (resultData.role === 'confirm') {

  //         console.log('BOOKED!');

  //       }

  //     });

  // }

  async onBookPlace() {

    const actionSheet = await this.actionSheetCtrl.create({

      header: 'Choose an Action',

      buttons: [

        {

          text: 'Select Date',

          role: 'select',

          handler: () => {

            this.openBookingModal('select');

          },

        },

        {

          text: 'Random Date',

          role: 'random',

          handler: () => {

            this.openBookingModal('random');

          },

        },

        {

          text: 'Cancel',

          role: 'cancel',

          handler: () => {

            console.log('Cancel clicked');

          },

        },

      ],

    });

    await actionSheet.present();

    const { role } = await actionSheet.onDidDismiss();

    console.log('onDidDismiss resolved with role', role);

  }

  async openBookingModal(mode: 'select' | 'random') {

    console.log('modal' + mode);

    let modalCtrl = await this.modalCtrl.create({

      component: CreateBookingComponent,

      componentProps: {

        selectedPlace: this.place,

      },

      //id:'modal-ID'

    });

    await modalCtrl.present();

    const { role, data } = await modalCtrl.onDidDismiss();

    console.log(`${role}`);

    console.log(JSON.stringify(data));

  }

}

CreateBookingComponent.ts

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

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

import { Place } from '../../places/place.model';

@Component({

  selector: 'app-create-booking',

  templateUrl: './create-booking.component.html',

  styleUrls: ['./create-booking.component.scss'],

})

export class CreateBookingComponent implements OnInit {

  @Input() selectedPlace: Place; //passed by modal

  constructor(private modalCtrl: ModalController) {}

  ngOnInit() {}

  onCancel() {

    //In case mulitple modal, close particular modal with modal id as third argument

    //this.modalCtrl.dismiss(null, 'cancel','modal-ID');

    this.modalCtrl.dismiss(null, 'cancel');

  }

  onBookPlace() {

    this.modalCtrl.dismiss(

      {

        message: 'This is a dummy message!',

      },

      'confirm'

    );

  }

}