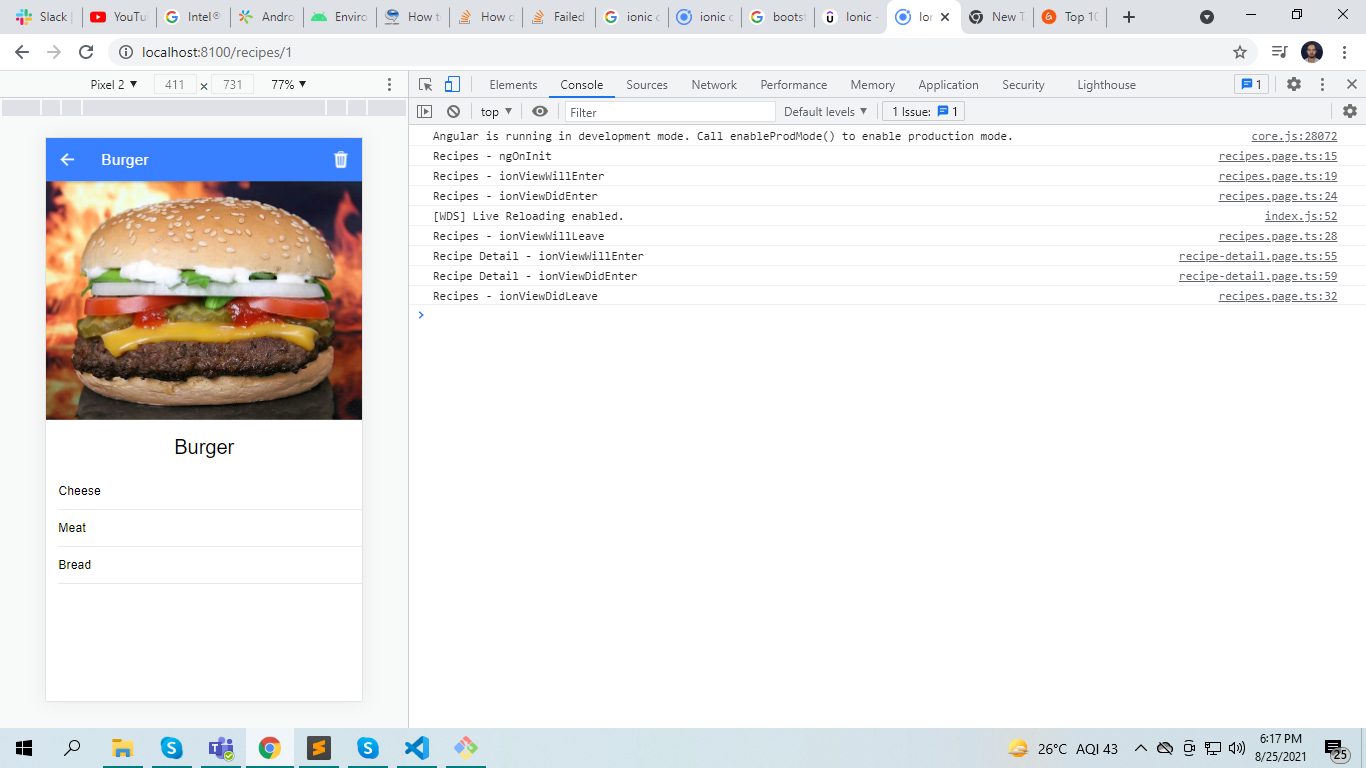
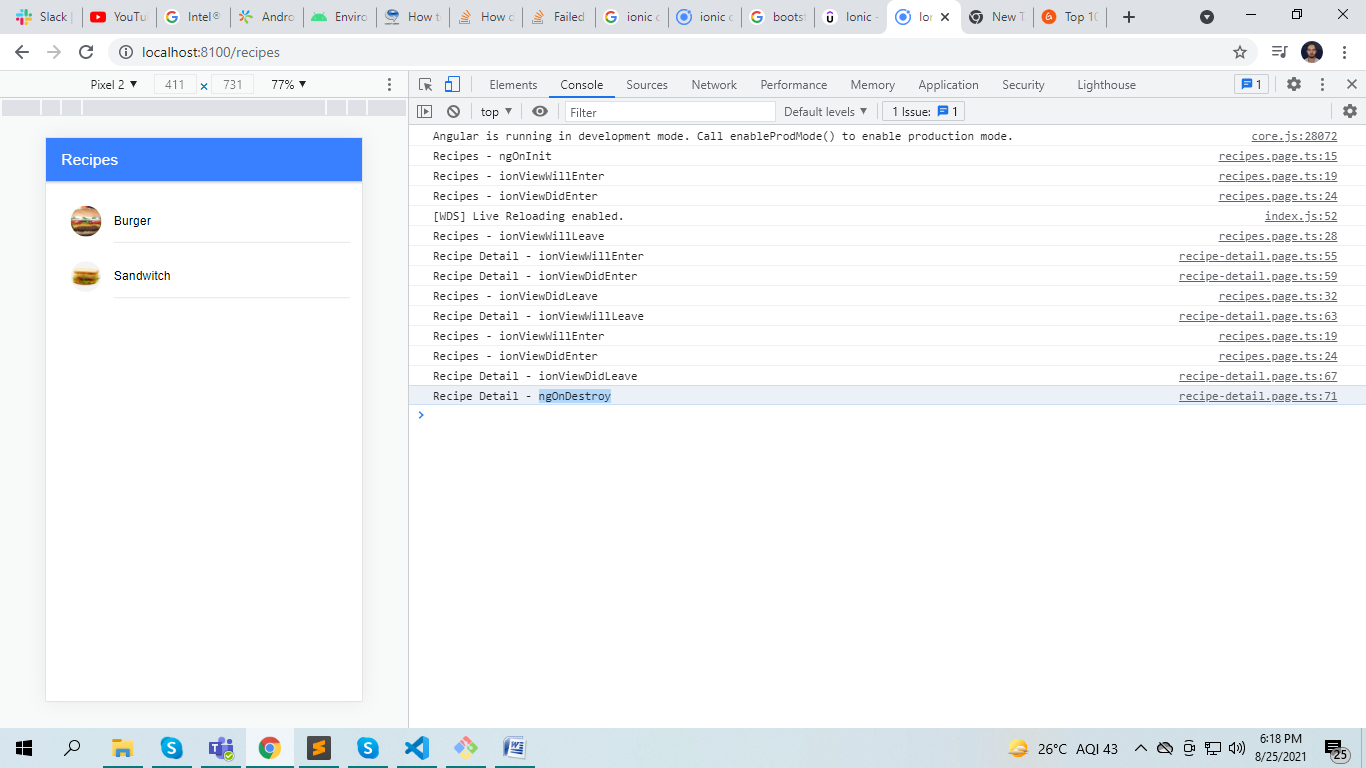
Ionic generate page recipes

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>