/\*

\* spurtcommerce

\* version 3.0.1

\* http://api.spurtcommerce.com

\*

\* Copyright (c) 2019 piccosoft ltd

\* Author piccosoft ltd <support@piccosoft.com>

\* Licensed under the MIT license.

\*/

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

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

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

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

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

import { SwiperConfigInterface, SwiperDirective } from 'ngx-swiper-wrapper';

import { ProductZoomComponent } from './product-zoom/product-zoom.component';

import { ListsSandbox } from '../../../../core/lists/lists.sandbox';

import { ConfigService } from '../../../../core/service/config.service';

import { Subscription } from 'rxjs';

import { MatDialog } from '@angular/material';

@Component({

selector: 'app-product',

templateUrl: './product.component.html',

styleUrls: ['./product.component.scss']

})

export class ProductComponent implements OnInit, OnDestroy, AfterViewInit {

// decorator

@ViewChild('zoomViewer') zoomViewer;

@ViewChild(SwiperDirective) directiveRef: SwiperDirective;

// configuration

public config: SwiperConfigInterface = {};

public configuration: SwiperConfigInterface = {};

public product: any;

// images

public image: any;

public zoomImage: any;

public imagePath: string;

public productImageId: number;

// route params

private sub: any;

private id: any;

// available options

public templateHidden: number;

// subcription

private subscriptions: Array<Subscription> = [];

isActive = [];

constructor(

private activatedRoute: ActivatedRoute,

public formBuilder: FormBuilder,

public dialog: MatDialog,

public productDetail: ListsSandbox,

private configService: ConfigService,

private changeDetectRef: ChangeDetectorRef,

private router: Router

) {

this.router.events.subscribe(evt => {

if (evt instanceof NavigationEnd) {

this.router.navigated = true;

window.scrollTo(0, 0);

}

});

}

/\*\* Initially initialize getProductdetail,getRelatedProducts when subscribed

\* subscribe productDetails$ and set initially default values for required options \*\*/

ngOnInit() {

this.imagePath = this.configService.getImageUrl();

// subscribe route params and trigger selected product detail, related products

this.sub = this.activatedRoute.params.subscribe(params => {

this.id = params['id'];

this.getProductdetail();

this.getBannerList();

});

}

// fetch banner list from sandbox

getBannerList() {

const params: any = {};

params.limit = 100;

params.offset = 0;

this.productDetail.getBannerList(params);

}

/\*\*

\* calls productDetail getProductDetails with param.Then subscribe productDetail productDetails$

\*

\* Then store the image path and image name in the array.

\* \*\*/

getProductdetail() {

const params: any = {};

params.id = this.id;

this.productDetail.getProductDetails(params);

this.subscriptions.push(

this.productDetail.productDetails$.subscribe(detail => {

if (detail && detail.productImage.length > 0) {

const imageLength = detail.productImage.length;

this.isActive = [];

this.isActive[detail.productImage[0].productId] = true;

for (let i = 0; i < imageLength; i++) {if (detail.productImage[i].defaultImage === 1) {

this.productImageId = detail.productImage[0].productId;

this.image =

this.imagePath + '?path=' +

detail.productImage[0].containerName + '&name=' +

detail.productImage[0].image +

'&width=390&height=390';

this.image = detail.productImage[0].containerName + detail.productImage[0].image;

this.zoomImage =

this.imagePath + '?path=' +

detail.productImage[i].containerName + '&name=' +

detail.productImage[i].image +

'&width=660&height=660';

this.zoomImage =

'/assets/images/'+detail.productImage[0].containerName + detail.productImage[0].image;;

this.changeDetectRef.detectChanges();

setTimeout(() => {

this.config.observer = true;

this.changeDetectRef.detectChanges();

});

// if (detail.productImage[i].defaultImage === 1) {

// this.productImageId = detail.productImage[0].productId;

// this.image =

// this.imagePath + '?path=' +

// detail.productImage[i].containerName + '&name=' +

// detail.productImage[i].image +

// '&width=390&height=390';

// this.zoomImage =

// this.imagePath + '?path=' +

// detail.productImage[i].containerName + '&name=' +

// detail.productImage[i].image +

// '&width=660&height=660';

// this.changeDetectRef.detectChanges();

// setTimeout(() => {

// this.config.observer = true;

// this.changeDetectRef.detectChanges();

// });

}

}

}

})

);

}

ngAfterViewInit() {

this.config = {

observer: false,

direction: 'vertical',

autoHeight: true,

autoplay: true,

spaceBetween: 10,

keyboard: true,

navigation: true,

grabCursor: true,

pagination: false,

loop: false,

preloadImages: false,

lazy: true,

watchSlidesVisibility: true,

breakpoints: {

480: {

slidesPerView: 2

},

600: {

slidesPerView: 3

}

}

};

this.configuration = {

observer: false,

slidesPerView: 4,

spaceBetween: 10,

keyboard: true,

navigation: true,

grabCursor: true,

pagination: false,

loop: false,

preloadImages: false,

lazy: true,

autoplay: false,

watchSlidesVisibility: true,

breakpoints: {

480: {

slidesPerView: 2

},

600: {

slidesPerView: 3

}

}

};

}

// view the selected image

public selectImage(image, i) {

this.productImageId = image.productId;

this.isActive = [];

this.isActive[image.productId] = true;

/\* this.image =

this.imagePath + '?path=' +

image.containerName + '&name=' +

image.image +

'&width=390&height=390';\*/

this.image = image.containerName + image.image;

this.changeDetectRef.detectChanges();

/\*this.zoomImage =

this.imagePath + '?path=' +

image.containerName + '&name=' +

image.image +

'&width=660&height=660';\*/

this.zoomImage = '/assets/images/'+image.containerName + image.image;

console.log("this.image", this.image);

console.log("this.zoomImage", this.zoomImage);

}

// zoom the image when mouse got moved over the image

public onMouseMove(e) {

if (window.innerWidth >= 1280) {

let image, offsetX, offsetY, x, y, zoomer;

image = e.currentTarget;

offsetX = e.offsetX;

offsetY = e.offsetY;

x = (offsetX / image.offsetWidth) \* 100;

y = (offsetY / image.offsetHeight) \* 100;

if (this.zoomImage) {

zoomer = this.zoomViewer.nativeElement.children[0];

}

if (zoomer) {

zoomer.style.backgroundPosition = x + '% ' + y + '%';

zoomer.style.display = 'block';

zoomer.style.height = image.height + 'px';

zoomer.style.width = image.width + 'px';

}

}

}

// event when mouse clicked to zoom the image

public onMouseLeave(event) {

this.zoomViewer.nativeElement.children[0].style.display = 'none';

}

// open pop to view the zoom image

public openZoomViewer() {

this.dialog.open(ProductZoomComponent, {

data: this.zoomImage,

panelClass: 'zoom-dialog'

});

}

// unsubscribe subscribed events while destroy the page

ngOnDestroy() {

this.sub.unsubscribe();

this.subscriptions.forEach(each => {

each.unsubscribe();

});

}

}