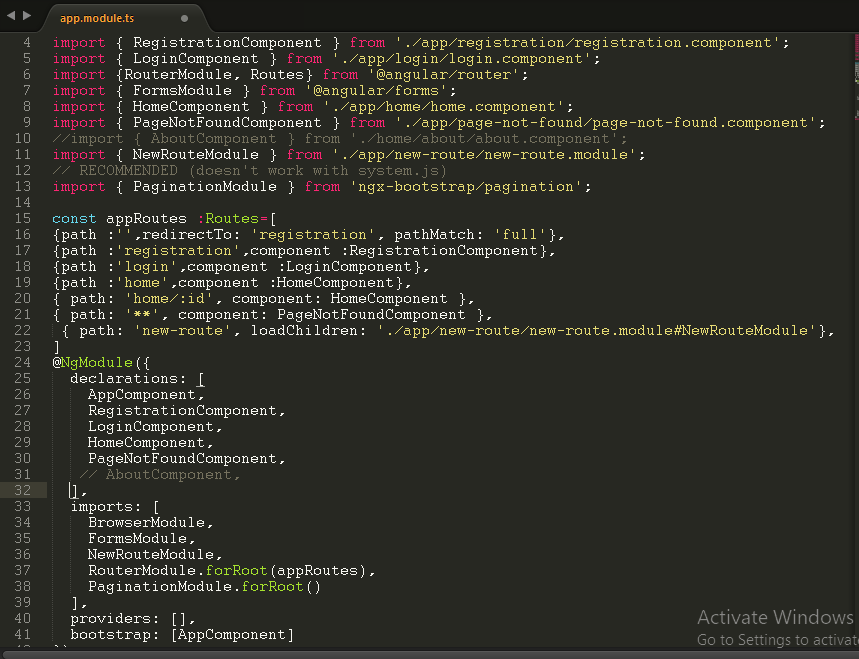
**Routing and Navigation**

Routing :-

i). First we need to imports angular library package '@angular/router' in app.module.ts:

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



ii). We then importing RouterModule.forRoot(routes) into our NgModule:

@NgModule({

imports: [

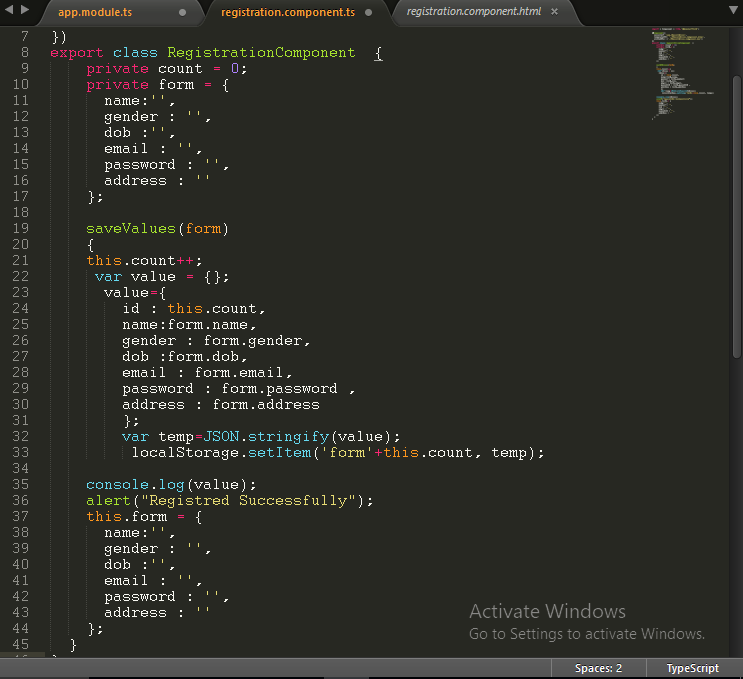
RouterModule.forRoot(routes, {useHash: true})

]

.

.

})



iii). The mapping of URLs to Components we want displayed on the page is done via something called a Route Configuration, at it’s core it’s just an array (app.module.ts):

const routes: Routes = [

{ path: 'registration', component: RegistrationComponent },

{ path: 'login', component: LoginComponent }

];

NOTE :

i). Path : The path property describes the URL this route will handle.

ii).Component : The component property is the name of the component we want to display when the URL in the browser matches this path.

OR

Redirects : There are a few more ways to configure our routes, for example we might like to change our routes to add some redirects like so:

const routes: Routes = [

{ path: ' ', redirectTo: ‘registration', pathMatch: 'full' },

{ path: 'registration', component: RegistrationComponent },

{ path: 'login', component: LoginComponent }

];

and

Wildcard route: Any other URL causes the router to throw an error and crash the app.

Instead of adding the "/sidekicks" route, define a wildcard route instead and have it navigate to a simple PageNotFoundComponent.

{ path: '\*\*', component: PageNotFoundComponent }

iv). RouterOutlet : the router matches that URL to the route 'Path' and displays the 'Component' after a RouterOutlet that you've placed in the host view's HTML (app.component.html).

<router-outlet></router-outlet>

Navigation :-

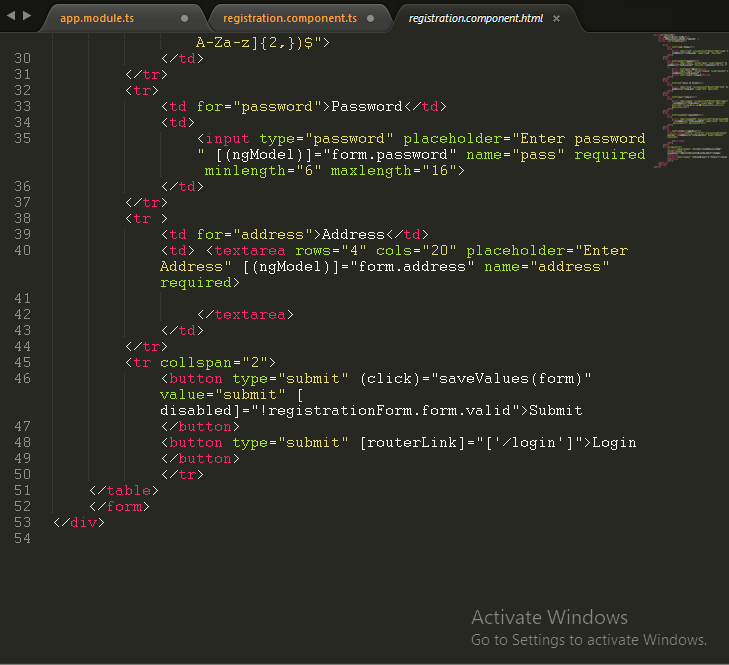
v). Router links : If we want to navigate between routes, we use the RouterLink directive. So if we wanted to link to our registration and login page from a navigation (registration.component.html).

<button type="submit" (click)="saveValues(form)" value="submit" [disabled]="!registrationForm.form.valid">Submit

</button>

<button type="submit" [routerLink]="['/login']">Login

</button>



NOTE: - For moreinfo: https://angular.io/guide/router#basics-wrap-up

a). import lib. in your page if we could navigate to the route programmatically.

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

constructor(private router: Router){ }

onSubmit(){

for(var i=1;i<localStorage.length;i++)

{

var emailVal=JSON.parse(localStorage.getItem('form'+i));

console.log(emailVal.email);

if((log.email==emailVal.email)&&(log.password==emailVal.password))

{

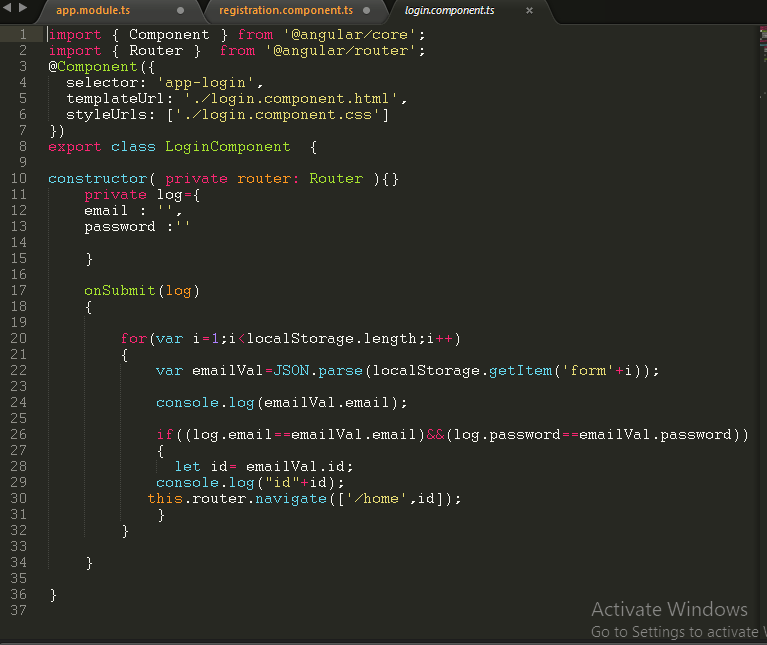
let id= emailVal.id;

console.log("id"+id);

this.router.navigate(['/home',id]);

}

}



vi). Using Route Parameters : we need a route parameter for the ID of that product. We could implement this using the following Routes:

const routes: Routes = [

{ path: '', redirectTo: 'registration', pathMatch: 'full' },

{path :'registration',component :RegistrationComponent},

{path :'login',component :LoginComponent},

{path :'home',component :HomeComponent},

{ path: 'home/:id', component: HomeComponent },

];

vii). Linking to Routes with Parameters : The routerLink directive passes an array which specifies the path and the route parameter. Alternatively we could navigate to the route programmatically:

private id: any = 15;

goToSearch(id) {

this.router.navigate(['/search', id]);

}

viii). Reading Route Parameters : The Search component must read the parameter, then load the search based on the ID given in the parameter.

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

private id: any;

private sub: any;

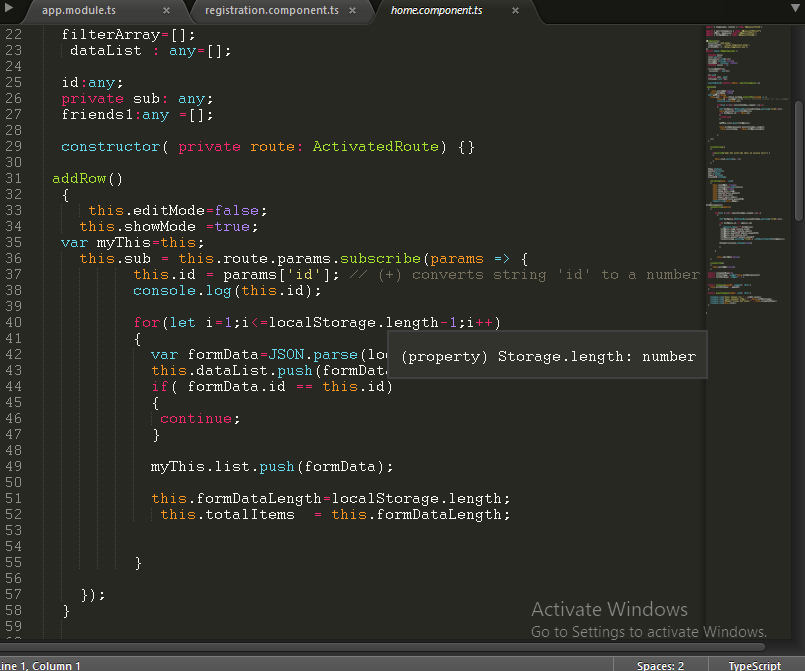
constructor( private route: ActivatedRoute){

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

this.sub = +params['id'];

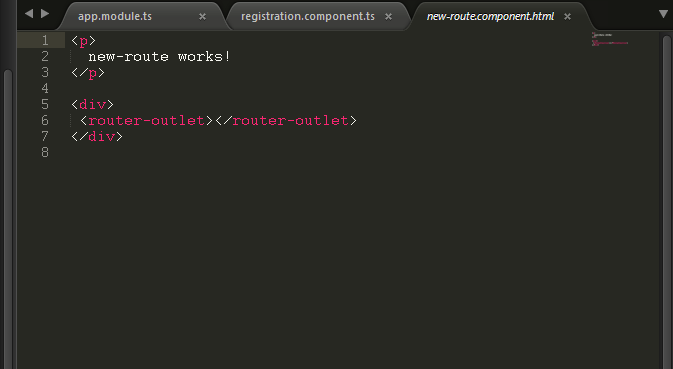
});

}

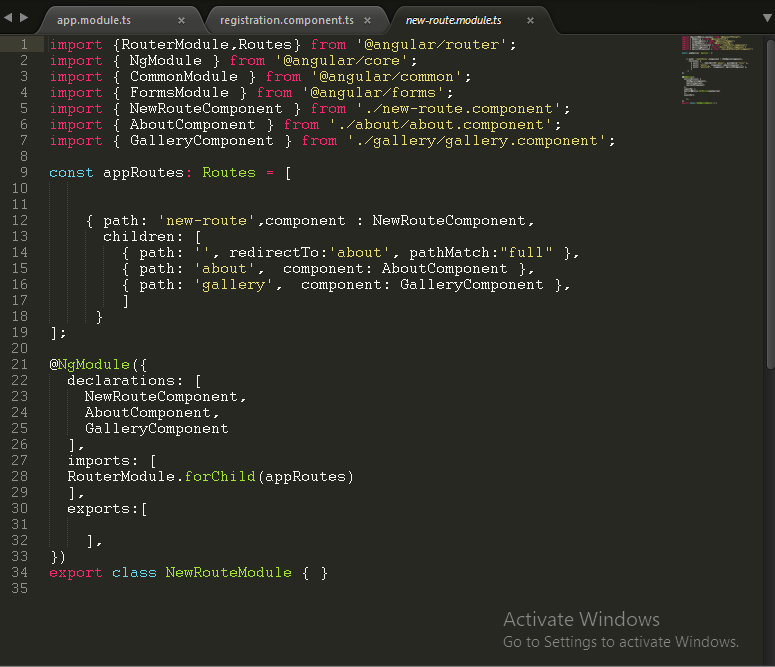


CHILD ROUTING

For child routing, we create another component with module. Here we create new-route child component



New-route.module.ts



Its other component create as app component .

Remember that, NewRouteModule define in app module and here we doesn’t declare it in ngModule . In the place of declaration , we import it.

Its routing define as

{ path: 'new-route', loadChildren: './app/new-route/new-route.module#NewRouteModule'}