**Фреймворк Angular 9**

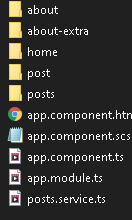
**Лабораторная работа № 7**

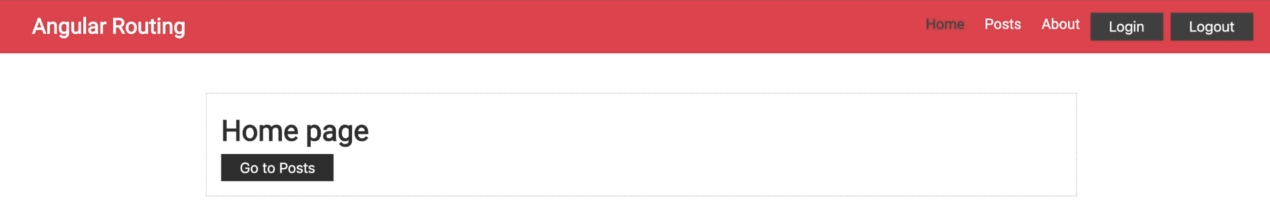
**Мультистраничность с роутингом**

**Задача. Создать сайт из нескольких страниц и организовать роутинг в SPA.**

**Ход работы**

1. **Скачать с гита исходные файлы (src\_lr7) для последующей работы. Создать новый проект используя эти файлы в которых содержаться компоненты различных страниц.**





1. **Зарегистрируйте набор роутов. Для этого в корне приложения создайте файл app-routing.module.ts.**

**Компонент HomeComponent должен именть путь** [**http://localhost:4200/**](http://localhost:4200/)

**Компонент AboutComponent должен именть путь** [**http://localhost:4200/about**](http://localhost:4200/about)

**Компонент PostsComponent должен именть путь** [**http://localhost:4200/**](http://localhost:4200/)**posts**

import {*NgModule*} from "@angular/core";  
import {RouterModule, Routes} from "@angular/router";  
import {HomeComponent} from "./home/home.component";  
import {AboutComponent} from "./about/about.component";  
import {PostsComponent} from "./posts/posts.component";  
  
const router: Routes = [  
 {path: '', component: HomeComponent},  
 {path: 'about', component: AboutComponent},  
 {path: 'posts', component: PostsComponent},  
 ]  
@NgModule({  
 imports:[RouterModule.forRoot(router)],  
 exports: [RouterModule]  
})  
export class AppRoutingModule {}

**Также в главном модуле app.module.ts необходимо зарегистрировать новый модуль для роутов.**

imports: [  
 BrowserModule,  
 FormsModule,  
 AppRoutingModule  
],

**Чтобы Angular понимал какую страницу необходимо рендерить при указании разного пути нужно добавить специальный компонент router-outlet в главный шаблон app.component.html**

<div class**="container"**>  
 <div class**="card"**>  
 <router-outlet></router-outlet>  
 </div>

**Проверьте грузиться ли нужная страница при указании соответствующего пути, например** [**http://localhost:4200/about**](http://localhost:4200/about)

1. **Теперь необходимо оживить ссылки в навигации. Для этого нужно указать правильный адрес в шаблоне**

<nav class**="navbar"**>  
 <h1>  
 <a href**="**/**"**>**Angular Routing**</a>  
 </h1>  
 <ul>  
 <li class**="active"**><a href**="**/**"** >**Home**</a>  
 </li>  
 <li ><a href **="/posts"**>**Posts**</a></li>  
 <li> <a href **="/about"**>**About**</a></li>  
 <li ><button class**="btn"** >**Login**</button></li>  
 <li><button class**="btn"Logout**</button></li>  
 </ul>  
</nav>

**Но при таком походе при переходе по ссылкам, страница будет перезагружаться, что не соответствует духу SPA.**

**Для этого используют специальную директиву routerLink**

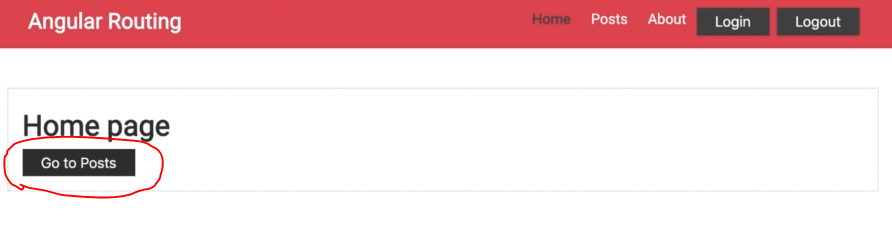
<a [routerLink]**="**['/']**"**>**Angular Routing**</a>  
 </h1>  
 <ul>  
 <li routerLinkActive**="active"** >  
 < a [routerLink]**="**['/']**"** >**Home**</a>  
 </li>  
 <li ><a routerLink**="/posts"**>**Posts**</a></li>  
 <li ><a routerLink**="/about"**>**About**</a></li>

1. **Следующая задача это показ пользователю активной ссылки. По умолчанию активной ссылкой является страница Home <li class="active"><a href="/" >Home</a>**

**И при переходе на другую страницу класс "active" все равно остается на странице Home. Чтоб это исправить существует директива routerLinkActive и routerLinkActiveOptions (ее нужно использовать чтобы задать полный и точный путь при котором будет отрабатывать директива routerLinkActive).**

<nav class**="navbar"**>  
 <h1>  
 <a [routerLink]**="**['/']**"**>**Angular Routing**</a>  
 </h1>  
 <ul>  
 <li routerLinkActive**="active"** [routerLinkActiveOptions]**="**{exact:  
true}**"** >  
 <a [routerLink]**="**['/']**"** >**Home**</a>  
 </li>  
 <li routerLinkActive**="active"** ><a routerLink**="/posts"**>**Posts**</a></li>  
 <li routerLinkActive**="active"**><a routerLink**="/about"**>**About**</a></li>  
 <li ><button class**="btn"** >**Login**</button></li>  
 <li><button class**="btn"** >**Logout**</button></li>  
 </ul>  
</nav>

1. **Сделаем программную навигацию для кнопки «Go to Post», по нажатию на которую будет осуществлен переход на страницу постов без перезагрузки приложения.**



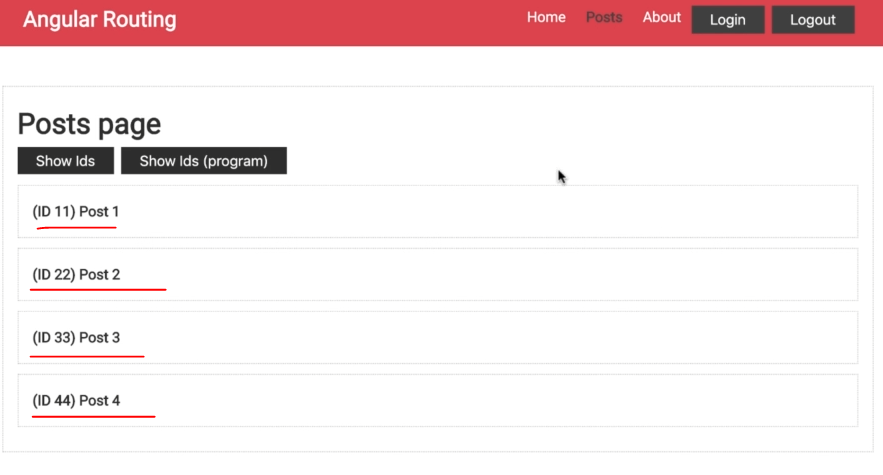
**Для начала создадим в шаблоне компонента** [home.component.html](http://home.component.html) **прослушку событий.**

<h1>**Home page**</h1>  
<div>  
 <button class**="btn"** (click)**="**goToPosts()**"**>**Go to Posts**</button>  
</div>

**В самой компоненте** [**home.component.ts**](http://home.component.ts) **реализуем метод goToPosts()**

import {*Component*} from '@angular/core'  
import {Router} from "@angular/router";  
  
@Component({  
 selector: 'app-home',  
 templateUrl: './home.component.html',  
 styleUrls: ['./home.component.scss']  
})  
export class HomeComponent {  
constructor(private router: Router) {}  
 goToPosts() {  
 this.router.navigate(['/posts'])  
 }  
}

1. **Разберём процесс создания динамических путей. Данные пути будут присутствовать на странице постов**



**Для начала нам нужно указать что наше приложение будет обрабатывать еще и какую-то динамику для этого в app-routing.module.ts добавляем:**

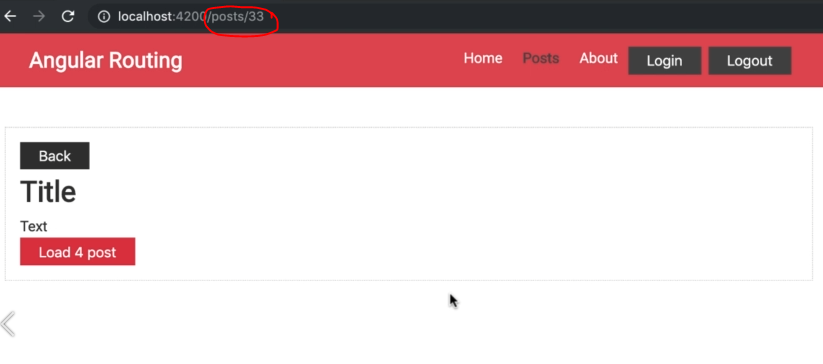
const router: Routes = [  
 {path: '', component: HomeComponent},  
 {path: 'about', component: AboutComponent, canActivateChild: [AuthGuard], children: [  
 {path: 'extra', component: AboutExtraComponent}]  
 },  
 {path: 'posts', component: PostsComponent, canActivate: [AuthGuard]},  
 {path: 'post/:id', component: PostComponent }

]  
@NgModule({  
 imports:[RouterModule.forRoot(router)],  
 exports: [RouterModule]  
})  
export class AppRoutingModule {}

**Далее в шаблоне поста нужно указать что данная ссылка будет передавать динамику, поэтому в файле posts.component.ts указываем следующую запись.**

<h1>**Posts page**</h1>  
  
<button class**="btn"**>  
 **Show Ids**</button>  
<button class**="btn"**>  
 **Show Ids (program)**</button>  
  
<div class**="card"** *\*ngFor***="**let post of postsService.posts**"**>  
 <h4>  
 <a [routerLink]**="**['/post',post.id]**"**>  
 <strong>**(ID** {{post.id}}**)**</strong>  
 {{post.title}}  
 </a>  
 </h4>  
</div>

**Проверить работу динамических ссылок.**



1. **Следующей задачей является определения какой сейчас id в адресной строке отображается и загрузить нужный пост. Т.е. считать информацию с адресной строки. Для этого используется специальный сервис ActivatedRout, который реализуем в файле post.component.ts**

import {*Component*, OnInit} from '@angular/core'  
import {ActivatedRoute} from "@angular/router";  
import {Post, PostsService} from "../posts.service";

export class PostComponent implements OnInit{  
 post: Post  
constructor(private router: ActivatedRoute,  
 private postS: PostsService  
) {}  
  
 ngOnInit(): void {  
 *this.router.params.subscribe((param)=>{  
 this.post= this.postS.getById(+param.id) }* }  
}

**!!!Самостоятельно.**  **Реализовать на странице постов методы по кнопкам “Back” и“Load 4 post”**

