



ANGULAR4 & SPRING BOOT

Разделение по компоненти (Рутинг)

Съдържание

- Какво е рутинг
- Създаване на рутинг
- Използване на рутинг навигацията
- Разделение на проекта по компоненти

Какво е рутинг

Познатата ни навигация

Браузър модела е познатия ни до сега за навигация на приложението:

- Въвеждаме URL и браузърът ни навигира на дадената страница
- Натискаме на линк от страницата и браузърът ни навигира на нова
- Натискаме бутона назад или напред на браузърът и той ни навигира според историята на страниците, които сте посетили

Какво е рутинг

Рутерът на Angular взима назаем този модел. Той интерпретира URL от браузъра, като навигация към създаденото вече от клиента view. Може да предава параметри на view-то, за да знае какво конкретно съдържание да зареди. Може да закачите рутера към конкретен линк, така че да се зареди определено приложение, когато потребителя достъпи този линк., натисне бутон и тн. Също така рутерът пази история в браузърът и може да се използват бутоните за напред и назад.

Създаване на рутинг

Създаване на рутинг

Създаваме файл, който се казва име-на-компонента.routes.ts и изглежда по следния начин:

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

import { PoemPlayerComponent } from './poem-player.component';
import { FindPoemsComponent } from './find-poems.component';
import { UploadPoemComponent } from './upload-poem.component';

export const appRoutes: Routes = [
  { path: '', redirectTo: 'find-poems', pathMatch: 'full' },
  { path: 'poem-player/:id', component: PoemPlayerComponent },
  { path: 'upload-poem', component: UploadPoemComponent },
  { path: 'find-poems', component: FindPoemsComponent }
];

@NgModule({
  imports: [RouterModule.forRoot(appRoutes, { useHash: true })],
  exports: [RouterModule]
})

export class AppRoutes { }
```

Създаване на рутинг

След което го добавяме към съответния модул:

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { RouterModule } from '@angular/router';
import { PoemsRoutes } from './poems.routes';
```

```
@NgModule({
  imports: [
    BrowserModule,
    PoemsRoutes,
    RouterModule
  ],
  declarations: [ ]
})
export class PoemsModule { }
```


Използване на рутинг навигацията

Използване на рутинг навигацията

В главния темплейт на апп-а трябва да се добави `<router-outlet></router-outlet>` след което ще бъдат заредени съответните компоненти на съответния път. За да се разбере към кой компонент да навигира, освен чрез URL, това става и чрез таг:

template: `

```
<h1>Angular Router</h1>
```

```
<nav>
```

```
  <a routerLink="/crisis-center" routerLinkActive="active">Crisis Center</a>
```

```
  <a routerLink="/heroes" routerLinkActive="active">Heroes</a>
```

```
</nav>
```

```
<button routerLink="/sidekicks">Go to sidekicks</button>
```

```
<router-outlet></router-outlet>
```

```
`
```

Използване на рутинг навигацията

Също така при рутинга може да се зададат параметри, като ID например:

```
{ path: 'hero/:id', component: HeroDetailComponent }
```

Така когато посетим “localhost:3000/hero/15” ще ни се зареди информацията за героя с ид 15.

В компонента добавяме

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

```
import 'rxjs/add/operator/switchMap';
```

Към конструкция задължително добавяме :

```
constructor(  
  private route: ActivatedRoute,  
  private router: Router  
) {}
```

Използване на рутинг навигацията

След което в `ngOnInit` метода използваме `ActivatedRoute` сървиса, за да вземем параметъра от пътя към компонента:

```
ngOnInit() {  
  this.hero$ = this.route.paramMap  
    .switchMap((params: ParamMap) =>  
      this.service.getHero(params.get('id')));  
}
```

или

```
ngOnInit() {  
  let id = this.route.snapshot.paramMap.get('id');  
  
  this.hero$ = this.service.getHero(id);  
}
```

Използване на рутинг навигацията

Може да навигирате към отделни компоненти чрез методи:

```
gotoHeroes() {  
  this.router.navigate(['/heroes']);  
}
```

И да предавате параметри:

```
gotoHeroes(hero: Hero) {  
  let herold = hero ? hero.id : null;  
  // Pass along the hero id if available  
  // so that the HeroList component can select that hero.  
  this.router.navigate(['/heroes', id: herold]);  
}
```

Разделение на проекта по компоненти

Домашно

Домашно

*Разгледайте tutorиала: <https://angular.io/tutorial/toh-pt5>
и прочетете тази страница <https://angular.io/guide/router>*

Рефакторирайте предното си домашно, така, че да бъде разделено по компоненти и чрез рутинга да се предава информацията от единия на другия. Добавете сървис, който да симулира записване на информацията и четене. (За сървисите може да прочетете от тук <https://angular.io/tutorial/toh-pt4>).