

ANGULAR 7



Peter Gurský, peter.gursky@upjs.sk

Editácia používateľov

- Chceme vytvoriť komponent na editáciu a pridávanie používateľov
 - ▣ Na nových URL
 - `/users/edit/132`
 - `/users/new`
- Stačí jeden, budú sa líšiť iba tým, či na začiatku je model používateľa prázdny alebo vyplnený používateľ
- Ako si prečítať číslo z URL adresy?

Parametre URL:

- ak chceme poslať povinný parameter napr. pre URL adresu `http://localhost:4200/users/edit/25`
 - `<a [routerLink]="['/users', 'edit', 25]'>`
 - `app-routing.module.ts` doplníme
 - `{ path: '/users/edit/:id', component: UserEditComponent }`
 - v `UserEditComponent` :

```
import { ActivatedRoute } from '@angular/router';  
  
export class UserEditComponent implements OnInit {  
  
  constructor(private route: ActivatedRoute){}  
  
  ngOnInit() {  
    this.userId = this.route.snapshot.params['id'];  
  }  
}
```

táto syntax je (zatiaľ)
ok, ale iba ak sa tento
komponent už nebude
používať inde

Parametre URL:

- ak sa komponent znovupoužíva (nerenderujeme ho celý), napr. ak z jedného používateľa (rovnaká trasa) sa vieme prekliknúť na iného, počúvame na zmeny parametra:

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

ngOnInit() {
  this.route.paramMap.pipe(
    map((params: ParamMap) => params.get('id'))
  ).subscribe(userId => this.userId = userId);
}
```

Parametre URL:

- ak sa komponent znovupoužíva (nerenderujeme ho celý), napr. ak z jedného používateľa (rovnaká trasa) sa vieme prekliknúť na iného, počúvame na zmeny parametra:

```
import { ActivatedRoute, ParamMap } from '@angular/router';
import { switchMap } from 'rxjs/operators';

ngOnInit() {
  this.route.paramMap.pipe(
    switchMap((params: ParamMap) =>
      this.service.getUser(+params.get('id')))
  ).subscribe(user => this.user = user);
}
```

Parametre URL:

- ak sa komponent znovupoužíva (nerenderujeme ho celý), napr. ak z jedného používateľa (rovnaká trasa) sa vieme prekliknúť na iného, počúvame na zmeny parametra:

```
import { ActivatedRoute, ParamMap } from '@angular/router';
import { switchMap } from 'rxjs/operators';

ngOnInit() {
  this.route.paramMap.pipe(
    switchMap((params: ParamMap) =>
      this.service.getUser(+params.get('id')))
  ).subscribe(user => this.user = user);
}
```

Pre každú pritečenú hodnotu vráti prúd, z ktorého hodnoty majú odchádzať cez výstup.

Ak sa prúd vytvorený predchádzajúcou hodnotu ešte neukončil, násilne ho ukončí a posiela už iba hodnoty z nového prúdu

Parametre URL:

- ak sa komponent znovupoužíva (nerenderujeme ho celý), napr. ak z jedného používateľa (rovnaká trasa) sa vieme prekliknúť na iného, počúvame na zmeny parametra:

```
import { ActivatedRoute, ParamMap } from '@angular/router';
import { switchMap } from 'rxjs/operators';

ngOnInit() {
  this.route.paramMap.pipe(
    switchMap((params: ParamMap) =>
      this.service.getUser(+params.get('id')))
  ).subscribe(user => this.user = user);
}
```

okrem **get(parameter)**, môžeme použiť aj metódy:

- has(parameter)** – true, ak taký parameter máme
- getAll(parameter)** – ak máme pre parameter viac hodnôt
- keys()** – vráti názvy parametrov

nepovinné parametre

- Niekedy chceme v URL poslať nepovinné parametre napr. pre pagináciu

- `http://localhost:4200/users;page=2;count=10`

- v kóde:

- `this.router.navigate(['/users', { page: 2, count: 10 }]);`

matrix notácia cez ;
– nie cez ? a &

```
import { ActivatedRoute, ParamMap } from '@angular/router';  
import { switchMap } from 'rxjs/operators';
```

```
ngOnInit() {  
  this.route.paramMap.pipe(  
    switchMap((params: ParamMap) =>  
      this.page = params.has('page') ? +params.get('page') : 1);  
      this.count = params.has('count') ? +params.get('count') : 5);  
    return this.service.getUsers();  
  ).subscribe(users => this.users = users);  
}
```


Niektoré ďalšie parametre ActivatedRoute

- url
 - ▣ observable s poľom reťazcov častí cesty z URL v tejto trase
- data
 - ▣ observable s dátami pre danú trasu, vrátane dát získaných neskôr resolverom
 - ▣ do každej trasy vieme dať parameter data, ktorý môže obsahovať čokoľvek
- queryParamsMap
 - ▣ observable s parametrami pre všetky trasy
 - ▣ parametre v URL uvedené za otáznikom v tvare ?atr1=val1&atr2=val2
- fragment
 - ▣ observable s hodnotou fragmentu/časti stránky – v URL je to hodnota za #
 - ▣ referencuje element s daným atribútom id

Editácia používateľa

- Ako zabezpečiť dynamický počet checkboxov v závislosti na počte skupín?
 - ▣ Model pre každý checkbox je FormControl<boolean>
 - Vložíme ich do FormArray
 - Každý bude (ne)zaškrtnutý v závislosti od používateľa
 - ▣ Label-y si budeme musieť pamätať bokom (allGroups)

```
<div formArrayName="groups">
  @for (groupControl of groups.controls; track i; let i = $index) {
    <mat-checkbox [formControlName] = "i">
      {{allGroups[i].name}}
    </mat-checkbox>
  }
</div>
```