

ANGULAR 6



Peter Gurský, peter.gursky@upjs.sk

Vlastný validátor pre formulár

- vytvoríme si validátor na zhodu hesla a kontrolného hesla

```
passwordsMatchValidator(control: FormGroup): ValidationErrors {
  const password = control.get('password');
  const password2 = control.get('password2');
  if (password.value === password2.value) {
    password2.setErrors(null);
    return null;
  } else {
    password2.setErrors({ differentPasswords: 'Passwords do not
match' });
    return { differentPasswords: 'Passwords do not match' };
  }
}
```

Vlastný asynchrónny validátor

- vytvoríme si validátor na konflikt s menom alebo emailom pri registrácii
- použijeme na serveri endpoint `/user-conflicts`
 - ▣ vracia pole s názvami konfliktných premenných, možné hodnoty sú "email" a "name", alebo vráti prázdne pole ak konflikt nie je

Vymazanie používateľa

app/users/users-service.ts

```
deleteUser(id: number):Observable<boolean> {  
  return this.http.delete(this.url + 'user/' + id + '/' + this.token).pipe(  
    map(() => {  
      this.msgService.successMessage("User deleted successfully");  
      return true;  
    }  
  ),  
  catchError(error => this.processHttpError(error))  
);  
}
```

Dialóg na potvrdenie zmazania

- <https://material.angular.io/components/dialog/api>
- dialóg je obyčajný komponent, ktorý si nechá v konštruktore injektovať `MatDialogRef` prípadné dodatočné dáta
 - `constructor(private dialogRef: MatDialogRef<MyDialogComponent>, @Inject(MAT_DIALOG_DATA) private data: any) { }`
 - `dialogRef` má metódu `close(myResult)`, ktorým vieme odovzdať voľbu používateľa cez `Observable` kódu, ktorý dialóg vyvolal
- vyvolanie dialógu:
 - necháme si injektovať `dialog: MatDialog` v konštruktore
 - zavoláme `const dialogRef = this.dialog.open(MyDialogComponent, {data: "delete?"})`
 - počkáme si na odpoveď: `dialogRef.afterClosed().subscribe(result => { ... })`