

Template driven Forms

1. Usuwamy zawartość pliku app.component.html i tworzymy „normalny” formularz w HTML.
2. Musimy zaimportować moduł o nazwie „FormsModule” w module głównym aplikacji app.module.ts
3. Po stornie htmla do formularza dodajemy następujące atrybuty:

```
<form #myForm="ngForm" (ngSubmit)="onSubmit(myForm)">  
  <input type="text" name="login" ngModel><br>  
  <button type="submit">Wyślij</button>  
  
</form>
```
4. W pliku app.component.ts dodajemy metodę o nazwie onSubmit:

```
onSubmit(myForm: any) {  
  console.log(myForm.value); //tu mamy pobranie wartości z całego formularza (jako object)  
  console.log(myForm.value.login); //tu pobieramy zawartość z inputa z name = "login"  
}
```
5. Dodajemy bibliotekę Bootstrap: w konsoli wpisujemy npm i bootstrap (na egzaminie to będzie gotowe!).
6. W pliku angular.json we właściwości „styles” dodajemy ścieżkę do Bootstrapa:

```
"styles": [  
  "src/styles.scss",  
  "node_modules/bootstrap/dist/css/bootstrap.min.css"  
],
```
7. Restartujemy cały projekt Angulara, np. kill konsoli i wpisujemy na nowo: **ng s -o**
8. Wstawiamy label oraz klasy z Bootstrapa do formularza:

```
<label for="login" class="form-label">Login</label>  
<input class="form-control" type="text" name="login" id="login" ngModel>
```
9. Dodajemy style dla form w pliku app.component.scss:

```
form {  
  width: 25%;  
  margin: 30px;  
}
```

Reactive Forms

1. Usuwamy zawartość pliku app.component.html i tworzymy „normalny” formularz w HTML.
2. Musimy zaimportować moduł o nazwie „ReactiveFormsModule” w module głównym aplikacji app.module.ts
3. Do formularza dodajemy następujące atrybuty:

```
<form [formGroup]="form" (ngSubmit)="onSubmit()">
```
4. Do kontrolki formularza dodaję atrybut FormControlName

```
<input type="text" formControlName="name">
```
5. W pliku app.component.ts tworzymy właściwość (zmienną) **form** oraz metodę (funkcję) o nazwie **onSubmit**. Ostecznie kod wygląda następująco:

```
6. export class AppComponent {
```

```
7.  
8. form = new FormGroup({  
9.     name: new FormControl()  
10.});  
11.  
12. onSubmit(){  
13.     alert('test');  
14. }  
15.  
16.}
```

6. Zmieniamy zawartość metody onSubmit() na następującą:

```
onSubmit(){  
    console.log(this.form.value);  
}
```

7. Gdybyśmy mieli się odwołać do name bezpośrednio, to trzeba to wykonać w taki sposób:

```
onSubmit(){  
    console.log(this.form.value);  
    console.log(this.form.value.name);  
}
```