



# ANGULAR 2 FUNDAMENTALS

---

*December 13, 2017*

# PART 17

**FORMS: CODE DRIVEN**

# CODE DRIVEN

---

- Angular also provides a code driven way to do forms
- To use code driven we have to import ReactiveFormsModule instead of FormsModule
- Open app.module.ts and import Find:

```
import { FormsModule } from '@angular/forms';
```

- Replace:

```
import {FormsModule, ReactiveFormsModule} from '@angular/forms';
```

# CODE DRIVEN

---

- Add it to imports:

```
imports: [  
  routes,  
  BrowserModule,  
  FormsModule,  
  ReactiveFormsModule,  
  HttpClientModule  
],
```



# LET US START BUILDING FORMS

---

- ng g c admin-user-create
- Open admin.routes.ts and add

```
export const adminRoutes: Routes = [  
  { path: '', component: DashboardComponent },  
  { path: 'images', component: AdminImageListComponent },  
  { path: 'images/create', component: AdminImageCreateComponent },  
  { path: 'images/edit/:id', component: AdminImageEditComponent },  
  { path: 'images/delete/:id', component: AdminImageDeleteComponent },  
  { path: 'users/create', component: AdminUserCreateComponent },  
];
```

- Don't forget to import:

```
import { AdminUserCreateComponent } from './admin-user-create/admin-user-create.component';
```

- We can use a helper class called FormBuilder

# ADMIN-USER-CREATE.COMPONENT.TS

---

## ➤ Import:

```
import { FormBuilder, FormGroup } from '@angular/forms';
```

## ➤ Find:

```
export class AdminUserCreateComponent implements OnInit {
```

## ➤ Add below:

```
userForm: FormGroup;
```

## ➤ Find:

```
constructor() { }
```

# ADMIN-USER-CREATE.COMPONENT.TS

---

➤ Replace:

```
constructor(fb: FormBuilder) {  
  this.userForm = fb.group({  
    email: fb.control(''),  
    name: fb.control(''),  
    password: fb.control('')  
  });  
}
```



# EXPLANATION

---

- We created a new userForm

```
userForm: FormGroup;
```

- Then we inject FormBuilder

```
constructor(fb: FormBuilder) {
```

- We use fb.group to create a group which contains several controls like email, username etc.
- fb.control means the value of the field
- Finally we can bind those controls to input fields in our view

# ADMIN-USER-CREATE.COMPONENT.HTML

---

```
<div class="col-md-10 col-md-offset-1">
  <div>
    <a [routerLink]="['/admin/']" class="btn btn-default"> Back</a>
  </div>
</div>
```

# ADMIN-USER-CREATE.COMPONENT.HTML

.....

```
<div class="col-md-10 col-md-offset-1">
  <div class="well well bs-component">
    <form novalidate class="form-horizontal" (ngSubmit)="createUser()" [formGroup]="userForm">
      <fieldset>
        <legend>Add a new user</legend>
        <div class="form-group">
          <label for="email" class="col-lg-2 control-label">Email</label>
          <div class="col-lg-10">
            <input type="text" class="form-control" id="email" name="email" placeholder="Email of the user" formControlName="email">
          </div>
        </div>
        <div class="form-group">
          <label for="name" class="col-lg-2 control-label">Full name</label>
          <div class="col-lg-10">
            <input type="text" class="form-control" id="name" name="name" placeholder="Name of the user" formControlName="name">
          </div>
        </div>
        <div class="form-group">
          <label for="password" class="col-lg-2 control-label">Password</label>
          <div class="col-lg-10">
            <input type="password" class="form-control" id="password" name="password" placeholder="Password" formControlName="password">
          </div>
        </div>
      </fieldset>
    </form>
  </div>
</div>
```

# ADMIN-USER-CREATE.COMPONENT.HTML

---

```
<div class="form-group">
  <div class="col-lg-10 col-lg-offset-2">
    <a [routerLink]="['/admin/']" class="btn btn-default"> Cancel</a>
    <button type="submit" class="btn btn-primary">Create</button>
  </div>
</div>
</fieldset>
</form>
</div>
</div>
```

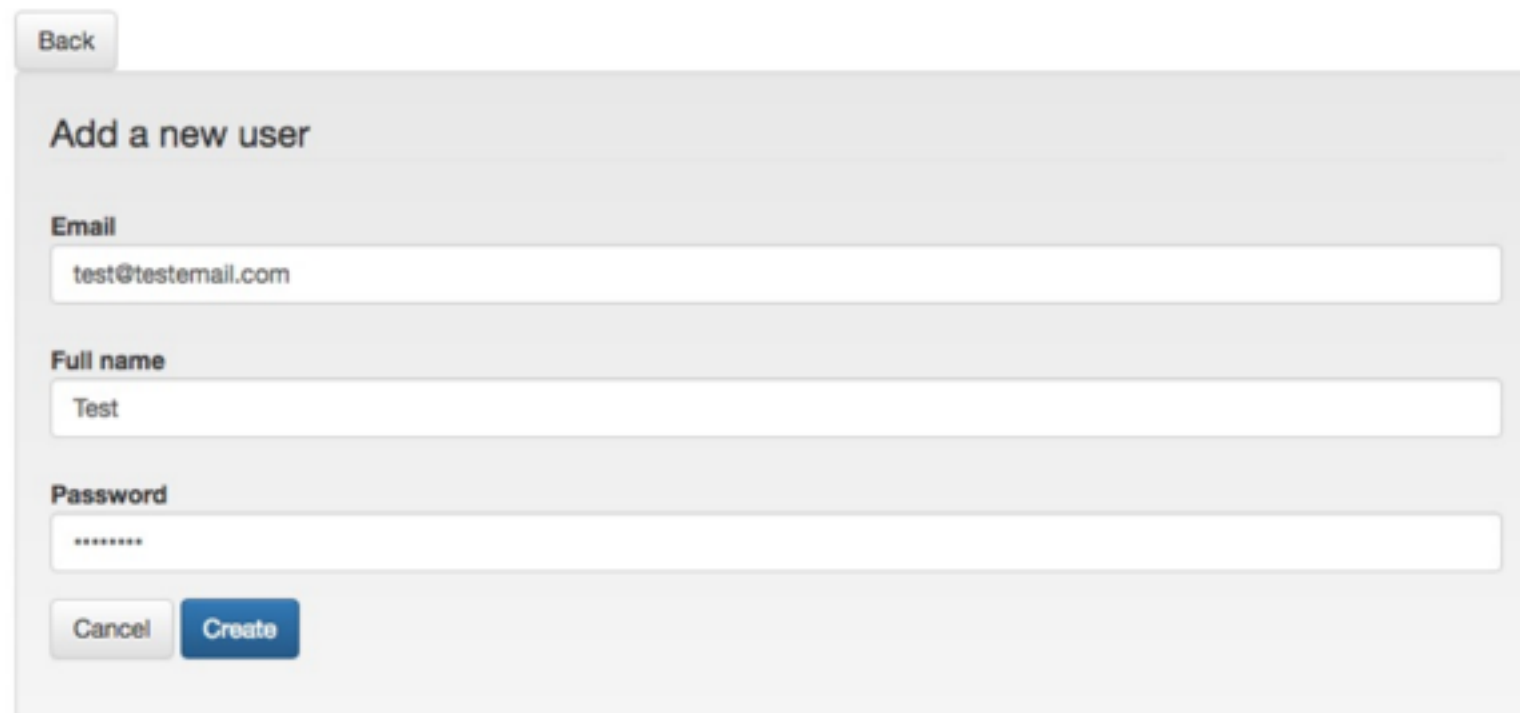
# LET'S MAKE THE CREATE USER METHOD

---

- Open admin-user-create.component.ts add:

```
createUser() {  
  console.log(this.userForm.value);  
}
```

- Go to <http://localhost:4200/admin/users/create> check console



The screenshot shows a web form titled "Add a new user" with a "Back" button at the top left. The form contains three input fields: "Email" with the value "test@testemail.com", "Full name" with the value "Test", and "Password" with masked characters "\*\*\*\*\*". At the bottom, there are two buttons: "Cancel" and "Create".

# FORM VALIDATION

---

**Validators.required:** the control should not be empty.

**Validators.minLength(number):** the control should have at least **number** of characters.

**Validators.maxLength(number):** the control should have at most **number** of characters.

**Validators.pattern(pattern):** the control must match the regular expression **pattern**.



# HOW TO USE VALIDATORS

---

- Open admin-user-create.component.ts find:

```
import {FormBuilder, FormGroup} from '@angular/forms';
```

- Update:

```
import {FormBuilder, FormGroup, Validators} from '@angular/forms';
```

- After that we can add some rules:

```
constructor(fb: FormBuilder) {  
  this.userForm = fb.group({  
    email: fb.control('', [Validators.required, Validators.minLength(3)]),  
    name: fb.control('', [Validators.required, Validators.minLength(3)]),  
    password: fb.control('', [Validators.required, Validators.minLength(6)])  
  });  
}
```

# EDIT ADMIN-USER-CREATE.COMPONENT.HTML

---

➤ Find:

```
<button type="submit" class="btn btn-primary">Create</button>
```

➤ Modify to:

```
<button type="submit" class="btn btn-primary" [disabled]="userForm.invalid">Create</button>
```

➤ Check the site:



The screenshot shows a user creation form with three input fields and two buttons. The first field is labeled 'Email' and contains the placeholder text 'Email of the user'. The second field is labeled 'Full name' and contains the placeholder text 'Name of the user'. The third field is labeled 'Password' and contains the placeholder text 'Password'. Below the fields are two buttons: 'Cancel' and 'Create'. The 'Create' button is highlighted in blue.

# UPDATED ADMIN-USER-CREATE.COMPONENT.HTML

---

```
<div class="col-md-10 col-md-offset-1">
  <div>
    <a [routerLink]="['/admin/']" class="btn btn-default"> Back</a>
  </div>
</div>
```





# UPDATED ADMIN-USER-CREATE.COMPONENT.HTML

.....

```
<div class="form-group">
  <label for="password" class="col-lg-2 control-label">Password</label>
  <div class="col-lg-10">
    <input type="password" class="form-control" id="password" name="password"
placeholder="Password" formControlName="password">
    <div *ngIf="userForm.get('password').dirty && userForm.get('password').has
Error('required')" class="alert alert-danger">Password is required</div>
    <div *ngIf="userForm.get('password').dirty && userForm.get('password').has
Error('minlength')" class="alert alert-danger">Password should have at least 6 charact
ers</div>
  </div>
</div>

<div class="form-group">
```

```
  <div class="col-lg-10 col-lg-offset-2">
    <a [routerLink]="['/admin/']" class="btn btn-default"> Cancel</a>
    <button type="submit" class="btn btn-primary" [disabled]="userForm.invalid"
>Create</button>
  </div>
</div>
</fieldset>
</form>
</div>
</div>
```



# CHECK FORM AGAIN

.....

[Back](#)

## Add a new user

**Email**

**Full name**

Name should have at least 3 characters

**Password**

Password should have at least 3 characters

[Cancel](#) [Create](#)

**END OF PART 17**