

Angular Basics – Learning Journal

Hello! I am currently learning Angular by following the Angular Basics course from Simplilearn.

This document summarizes the main concepts from the second part

Angular Forms

Allow you to capture user data, validate inputs and react to interactions in the form — in an organized way and integrated with Angular.

Types

- **Template-Driven**

- Method used to create a form object (Directives)

- Module Classes to controls

- Json values

- Validations

- **Reactive Forms**

- Data-flow

- Code-driven

- No two-way data binding

- Data binding is a technique that connects data sources to user interfaces (UI). This allows data to be updated automatically when the interface changes.

- Synchronous creating

Form Control

class that enables validation.



FormGroup

represents a group of controls.

also provides all the validation errors.



Development

****Creating a user form registration****

Generate the:

ng new democomponents

ng g c form-component

Verify that folder assets is on the angular.json to add an image

"assets": [

"src/assets",

"src/favicon.ico"

],

```
democomponents > src > app > app.component.html > app-form-component
1 
2 <app-form-component></app-form-component>
```

Import forms module:

```
app.component.ts M X logo.jpeg U angular.json M app.component.html M form-component.compe
democomponents > src > app > app.component.ts > ...
1  import { Component } from '@angular/core';
2  import { RouterOutlet } from '@angular/router';
3  import { FormComponentComponent } from './form-component/form-component.component';
4  import { FormsModule } from '@angular/forms';
5
6
7  @Component({
8    selector: 'app-root',
9    imports: [RouterOutlet, FormComponentComponent, FormsModule],
10    templateUrl: './app.component.html',
11    styleUrls: ['./app.component.css']
12  })
13  export class AppComponent {
14    title = 'democomponents';
15  }
16
```

Form page:

```
<div class="container">
  <h1>User Registration</h1>

  <form (ngSubmit)="submit(login)" #login="ngForm">
    <div class="form-group">
      <label for="firstname">First Name</label><br>
      <input #name="ngModel" required minlength="2"
maxlength="7" type="text" name="firstname" class="form-control"
ngModel>
      <div *ngIf="name.touched && name.invalid" class="alert
alert-danger">
        <p *ngIf="name.errors?.['required']">FirstName
Required</p>
        <p *ngIf="name.errors?.['minlength']">Sorry short
name!!</p>
      </div>
    </div>
    <pre></pre>
    <div class="form-group">
      <label for="lastname">Last Name</label><br>
      <input #lastname="ngModel" required minlength="2"
maxlength="7" type="text" name="lastname" class="form-control"
ngModel>
```

```

    </div>
    <pre></pre>
    <div class="form-group">
        <label for="email">Email</label><br>
        <input #email="ngModel" required minlength="2"
maxlength="7" type="text" name="email" class="form-control" ngModel>
    </div>
    <pre></pre>
    <div class="form-group">
        <label for="password">Password</label><br>
        <input #password="ngModel" required minlength="2"
maxlength="7"
        type="password"
        name="password"
class="form-control"ngModel>
    </div>
    <pre></pre>
    <button class="btn btn-primary"
type="submit">Submit</button>
  </form>
</div>

```

Necessary imports:

```

import { CommonModule } from '@angular/common';
import { Component } from '@angular/core';
import { FormsModule, NgForm } from '@angular/forms';

```

form_component.component.ts

```

import { Component } from '@angular/core';
import { RouterOutlet } from '@angular/router';
import { FormComponentComponent } from
'./form-component/form-component.component';
import { FormsModule } from '@angular/forms';

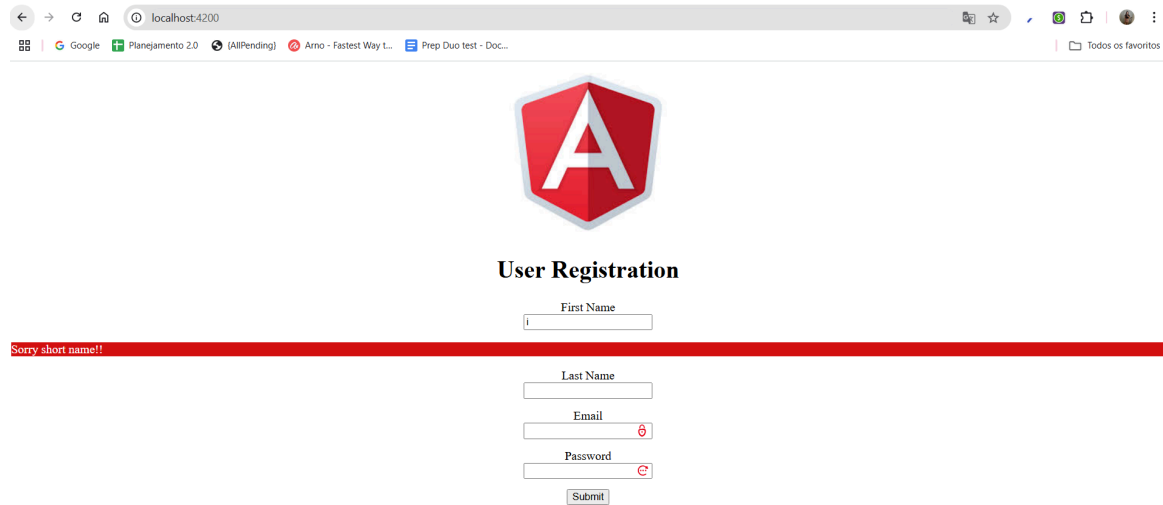
@Component({
  selector: 'app-root',
  imports: [RouterOutlet, FormComponentComponent, FormsModule],

```

```

    templateUrl: './app.component.html',
    styleUrls: ['./app.component.css']
  })
  export class AppComponent {
    title = 'democomponents';
  }

```



The screenshot shows a web browser window with the URL 'localhost:4200'. The page features the Angular logo at the top center, followed by the title 'User Registration'. Below the title is a registration form with four input fields: 'First Name', 'Last Name', 'Email', and 'Password'. Each field has a red error icon to its right. A red banner at the top of the form area displays the message 'Sorry short name!!'. At the bottom of the form is a 'Submit' button. The browser's address bar and tabs are visible at the top.

Angular Services

Services in Angular are responsible for handling business logic, HTTP calls, data storage, and any other reusable function that does not involve displaying content on the screen (UI).

Examples:

A service that logs in and out

A service that searches for products in the API

A service that saves data to localStorage



Features

- A service is a class
- Decorated with Injectable

@Injectable lets Angular know how to create and provide an instance of this class when needed.

- They share the same piece of code

- Hold the business logic
- Interact with the backend
- Share data among components
- Services are singleton
- Registered on modules or components

Development

Employee detail list

****Create a service****

ng g service data

Made with  while learning Angular with Simplilearn – Angular Basics.