

GUIDE FOR FA3

In angular.json:-

Add "node_modules/bootstrap/dist/css/bootstrap.min.css" in styles.

In app.module.ts

1)Add imports:-

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

```
import { HttpClientModule } from '@angular/common/http';
```

add the same in imports array under @NgModule

In app-routing.module.ts

1)Add the path in the routes array. Eg :-

```
const routes: Routes = [  
  {path:'search',component:SearchComponent},  
];
```

In app.component.html

1)Add the route to link. Eg :-

```
<a class="nav-link" [routerLink]="['search']" class="text-primary">Search</a>
```

2)Add router-outlet tag:-

```
<router-outlet></router-outlet>
```

In pipe:-

1)Implement the logic in transform method as needed. Eg:-

```
transform(value: unknown, ...args: unknown[]): unknown {  
  if(value == 'Not Found'){  
    return "No pokemon exists with that name"  
  }else{return value;}  
}
```

In service:-

1) Import the following:-

```
import { HttpClient } from '@angular/common/http';  
import { Observable } from 'rxjs';
```

2) Use DI in constructor for http. Eg:-

```
constructor(private http : HttpClient) { }
```

3) Initialize to the given URL. Eg:-

```
url:string = http://localhost:3333/infyLaptopApi/register;
```

4) Implement the given method with the given http method:-

Get method Example:-

```
fetchPokemonData(pokemonName: string) : Observable<Pokemon>{  
    return this.http.get<Pokemon>(this.url+pokemonName);  
}
```

Post method example:-

```
Register(laptopData:InfyLaptop) : Observable<InfyLaptop> {  
    return this.http.post<InfyLaptop>(this.url,laptopData);  
}
```

In the component for from i.e. whateverForm.component.ts:-

1)Import the following:-

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

2)Initialize the variables as needed and initialize make the form to FormGroup. Eg:-

```
pokemon : Pokemon;  
errorMessage : string;  
successMessage : string;  
pokemonSearchForm : FormGroup;
```

3)using DI inject the service and form:-

```
constructor(private pokemonDataService: PokemonDataService,private formBuilder:FormBuilder) { }
```

4)In ngOnInit initialize the form gusing formBuilder.group. Eg:-

```
this.pokemonSearchForm=this.formBuilder.group({  
  'pokemonName':[],[Validators.required,PokemonNameValidator.checkName]]  
})
```

*Note:- PokemonNameValidator.checkName is a custom validator

5)Implement the method:-

For Get Example:-

```
getPokemon(){  
  this.pokemon = null;  
  this.errorMessage = null;  
  let pokemonName : string = this.pokemonSearchForm.value.pokemonName;  
  this.pokemonDataService.fetchPokemonData(pokemonName.split(" ").join("-").toLowerCase()).subscribe(  
    pokemon => this.pokemon = pokemon,  
    response => this.errorMessage = response.error  
  );
```

For post example:-

```
register(){  
  this.errorMessage=null;  
  this.successMessage=null;  
  this.registrationId=null;  
  this.infyLaptopService.register(this.registrationForm.value).subscribe(  
    (data) => {  
      this.registrationId = data.registrationId+"";  
    },  
    (response) => {  
      this.errorMessage = response.error.message;  
    });  
}
```

In the HTML for from i.e. whateverForm.component.html:-

1)Add form-group and the error message span for every input mentioned eg:-

```
<div class="form-group">  
  <label for="searchPokemon"><span class="display-4">Search for  
pokemon:</span></label>  
  <input type="text" class="form-control" id="searchPokemon"  
formControlName="pokemonName" placeholder="Pokemon Name">  
</div>  
<p><span id="searchPokemonError" class="text-danger"  
*ngIf="pokemonSearchForm.controls.pokemonName.invalid &&  
pokemonSearchForm.controls.pokemonName.dirty">The Pokemon Name should have  
atleast 3 letters.</span></p>
```

2)Add the button for submitting if not already present. Eg:-

```
<button type="submit" [disabled]="pokemonSearchForm.invalid" class="btn btn-primary">Get  
Pokemon</button>
```

3)In the last after form add the error and success message spans:-

```
<p><span id="errorMessage" class="text-danger"  
*ngIf="errorMessage">{{errorMessage}}</span></p>  
  
<p><span id="successMessage" class="text-success" *ngIf="successMessage">{{successMessage |  
successMessagePipe}}</span></p>
```