Angular Training

(Intermediate to Advanced)

Angular

Narasimha

Sr. Corporate Trainer, Mentor

tnrao.trainer@gmail.com

Schedule for Angular Training

Day#	Date	Topic
Day-1	16-Nov-2023	Custom Pipes in Angular
Day-2	17-Nov-2023	Parent-Child Communication
Day-3	20-Nov-2023	Custom Directives in Angular
Day-4	**21-Nov-2023**	**Working with Reactive Forms**
Day-5	22-Nov-2023	Dependency Injection and Services in Angular
Day-6	23-Nov-2023	Http Client – Server calls in Angular
Day-7	24-Nov-2023	Routing and Security in Angular
Day-8	27-Nov-2023	Unit Testing in Angular

Duration : 8 days (2hours per day) ; 2pm to 4 pm; 16th Nov – 27th Nov

Day4
Working with Reactive Forms

Angular

Narasimha
Sr. Corporate Trainer, Mentor
tnrao.trainer@gmail.com

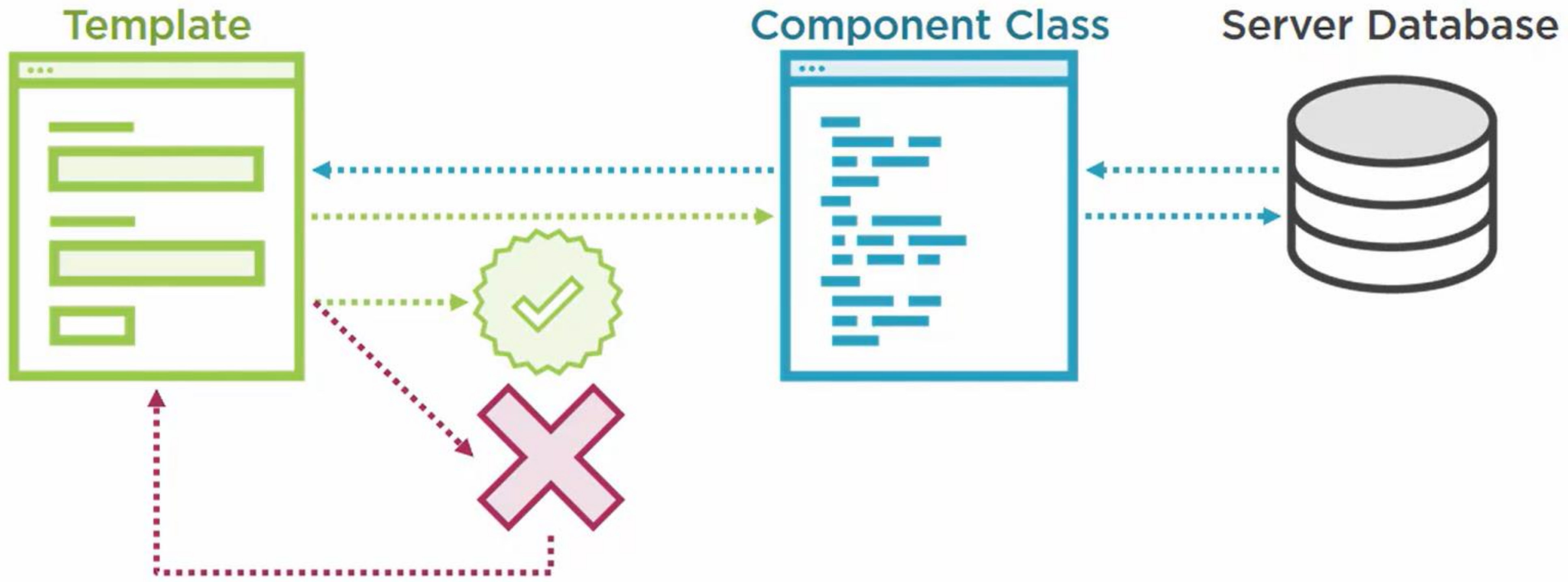
Index – Day4

Introduction to Angular Forms

Narasimha

Sr. IT Trainer/Consultant

Angular Forms



Template Driven Forms

Narasimha

Sr. IT Trainer/Consultant

Template Driven Forms

- Template Driven Forms Rely on directives in the template to create and manipulate the underlying object model.
- It is useful for adding a simple form to an app, such as an email signup form.
- Straight forward to add to an app, but they don't scale as well as reactive forms.
- If you have very basic form requirements and logic that can be managed solely in the template, template-driven forms could be a good fit.

Template Driven Forms

1. Create Template variable for form tag
   <form   #form1="ngForm">
2. Create Template variable for input tag
   <input #t1="ngModel">
3. Apply Validation directives
   <input required="true">
4. Prepare error message *ngIf - invalid, dirty
   <span *ngIf="t1.invalid"></span>
5. Apply disabled prop for submit button   - invalid
   <input [disabled]="form1.invalid" … />

Practical HandsOns

Reactive Forms

Narasimha
Sr. IT Trainer/Consultant

Reactive Forms

- Provide direct, explicit access to the underlying form's object model.
- Compared to template-driven forms, they are more robust: they're more scalable, reusable, and testable.
- If forms are a key part of your application, or you're already using reactive patterns for building your application, use reactive forms.

Steps to Create Reactive Forms

Component

1. Import **ReactiveFormsModule** in app.module
2. Import **FormGroup, FormControl, Validators** classes in component.
3. Define Validation rules using above classes

Template

4. Bind form and input tags with corresponding objects and props in template file.
5. Prepare error messages
6. Disabled button based on form validations.

Reactive Forms

```
import  {FormGroup,  FormControl,  Validators} from "@angular/forms“;
```

```
public customerForm = new FormGroup(
  {
      fname: new  FormControl(null, Validators.required),
     lname:  new  FormControl(null, Validators.required),
      …
  });
```

Reactive Forms – Template Binding

```
<form [formGroup]="customerForm">
First Name  :
  <input type="text"  formControlName="fname"  />
</form>
```

Final Implementation

```
public customerForm = new FormGroup(
  {
          fname: new  FormControl(null, Validators.required),
         lname:  new  FormControl(null, Validators.required),
          …
  });
```

```
<form [formGroup]="customerForm">
First Name  :
  <input type="text"  formControlName="fname"  />
</form>
```

Practical HandsOns

Q & A