Angular Training (Intermediate to Advanced)

Narasimha

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

Schedule for React JS Training

Day1 : Custom Pipes in Angular

Day2 : Parent-Child Communication

Day3 : Custom Directives in Angular

Day4: Working with Reactive Forms

Day5 : Dependency Injection and Services in Angular

Day6: RxJS Library – Observables

Day7: Http Client – Server calls in Angular

Day8: Routing and Security in Angular

Day4 Working with Reactive Forms



Narasimha

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

Index – Day4

- Introduction to Angular Forms
- Template Driven Forms
- Reactive Forms
- More on Reactive Forms

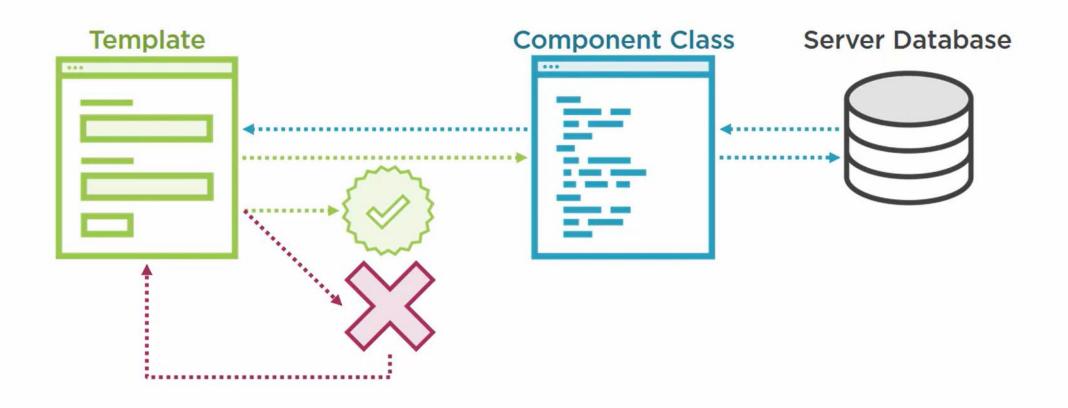


Introduction to Angular Forms

Narasimha

Sr. IT Trainer/Consultant

Angular Forms





Template Driven Forms

Template Driven Forms

- Rely on directives in the template to create and manipulate the underlying object model.
- Useful for adding a simple form to an app, such as an email list signup form.
- Straightforward 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, templatedriven 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 *nglf invalid, dirty
- 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

- 1. Import ReactiveFormsModule in app.module
- 2. Import FormGroup, FormControl, Validators classes in component.
- 3. Define Validation rules using above classes
- 4. Bind form and input tags with corresponding objects and props in template file.
- 5. Disabled button based on form validations.

Reactive Forms

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

Reactive Forms – Template Binding

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

Practical HandsOns

title

