Angular Training Notes

[Package.json 1](#_Toc68441779)

[Angular Forms 1](#_Toc68441780)

[Forms properties 1](#_Toc68441781)

[Angular Forms Agenda 2](#_Toc68441782)

[Angular Reactive Forms 2](#_Toc68441783)

[Forms basics 2](#_Toc68441784)

[Data Binding 2](#_Toc68441785)

[Form Validations 3](#_Toc68441786)

# Package.json

**Dependencies**

* Packages required for development & deployment

**devDependencies**

* Packages only required for development

# Angular Forms

## Forms properties

1. ng-touched [touched]
2. ng-untouched [untouched]
3. ng-pristine [pristine]
4. ng-dirty [dirty]
5. ng-valid [valid]
6. ng-invalid [invalid]

## Angular Forms Agenda

[To create functional and great looking forms for Angular Application]

1. Building a form from scratch
2. Data Binding
3. Validating our form data
4. Sending a form data to a server
5. Third-party form controls

# Angular Reactive Forms

2-types of Angular form Technologies

* **Template-Driven Forms**
  + Use a component’s Template
  + Unit Test Against DOM
* **Reactive Forms**
  + Use a component’s Template
  + Create a Form Model in TypeScript [Must by in sync with the template]
  + Unit Test Against Form Model
  + Validation in Form Model

# Forms basics

* Create simple user settings form with wide verity of controls i.e. checkboxes, radio buttons, Textbox, submit button, Drop Down

# Data Binding

* Store data somewhere in Angular. We store in regular JavaScript object or TS class using Interface
* Interface gives types information about different fields on the form & store data in JavaScript object
* We are using 2-way data binding to object

# Form Validations

* HTML 5 Attributes
  + required
  + pattern
  + minlength
  + maxlength
  + min
  + max

# HTTP Posting

* Form data is posted to a server & store in sort of data store such as database
* Here we are validating the data, which we need to posting properly to the server is an asynchronous event, for that we need to work with RXJS observables in handling the posting and showing error when needed

# Third-party Controls

* Do with the forms
* Quickly look at **PrimeNG**, a good source of form controls
* **Angular material**, which has a set of form controls,
* **ngx-bootstrap**

# Template-driven Forms

* **Template-Driven Forms**
  + Use a component’s Template
  + Unit Test Against DOM
  + Things done quickly if we use HTML-5 type forms Angular Process using Template Driven form
* **Reactive Forms**
  + Use a component’s Template
  + Create a Form Model in TypeScript [Must by in sync with the template]
  + Unit Test Against Form Model
  + Validation in Form Model
  + Need to include form model, if we change template, then need to change form-model and we need to keep two in Sync, which cause extra work
  + As big benefit is that we can write Unit Test Against form model, which create very fast Unit Test
  + For Unit Testing, Reactive form is the better choice