Angular Forms

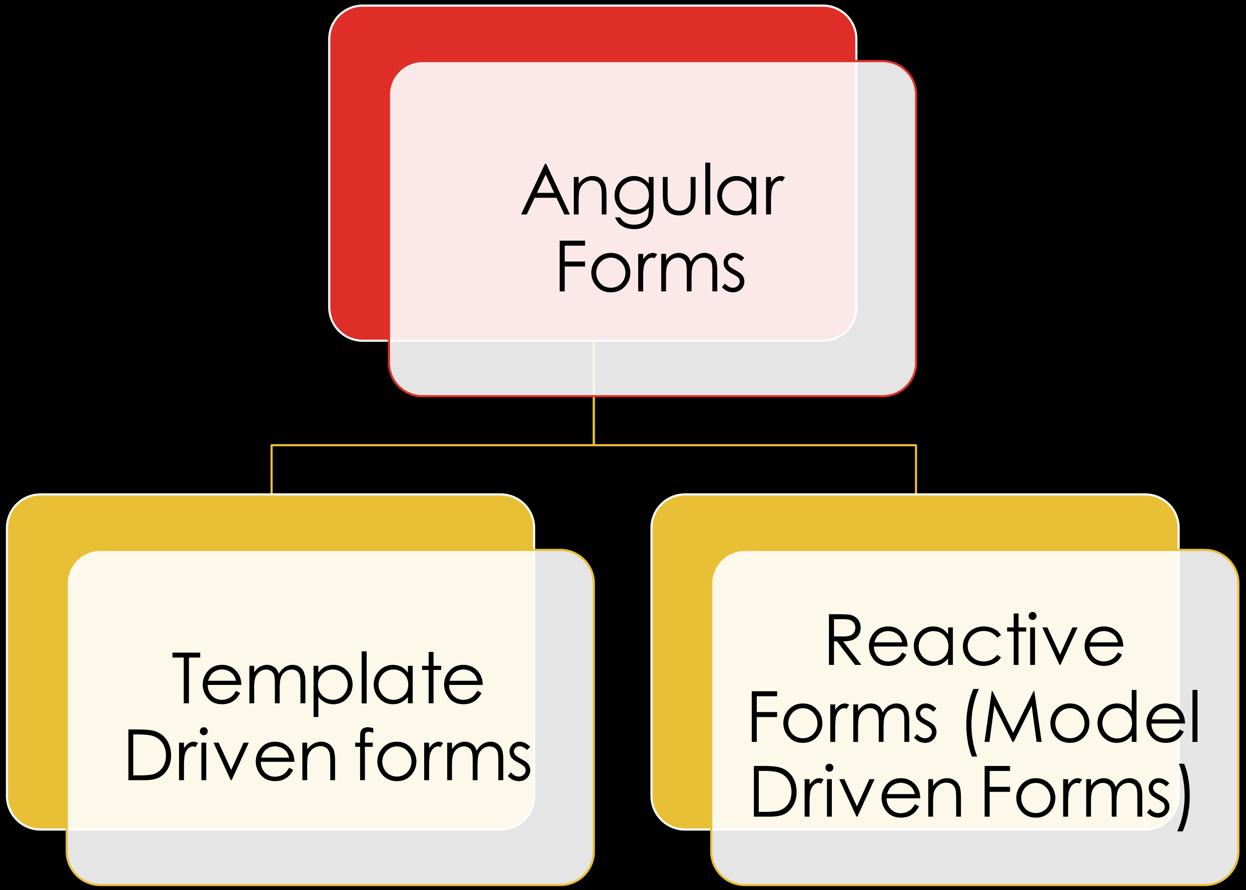
By Nikunj Soni

JAVA FSD/MERN/MEAN (Expert & Instructor)

ANGULAR FORMS

- A large category of frontend applications are very form-intensive, especially in the case of enterprise development.
- Many of these applications are basically just huge forms, spanning multiple tabs and dialogs and with non-trivial validation business logic.
- The Angular framework provides us a couple of alternative strategies for handling forms

ANGULAR FORM TYPE



Angular Forms

Template Driven forms

Reactive Forms (Model Driven Forms)

TEMPLATE DRIVEN FORMS

- Template-driven forms use **two-way data binding** to update the data model in the component as changes are made in the template and vice versa.
- You can build almost any kind of form with an Angular template
  - **login forms**
  - **contact forms**
  - **any business form.**
- You can lay out the controls creatively and bind them to the data in your object model.
- You can specify validation rules and display validation errors, conditionally allow input from specific controls, trigger built-in visual feedback, and much more.

ADVANTAGES

- Handling the forms is as simple as reactive forms.
- Template-driven forms are easy to use.
- They are similar to AngularJs.
- Two-way data binding (using [(NgModel)] syntax).
- Minimal component code.
- Template-driven form is also more than enough to build a large range of forms.

HOW TO BUILD A TEMPLATE DRIVEN FORM

- Use **FormsModule** to work with template Driven Forms.
- Directive used:

1. **NgModel:** allowing you to respond to user input with input validation and error handling.

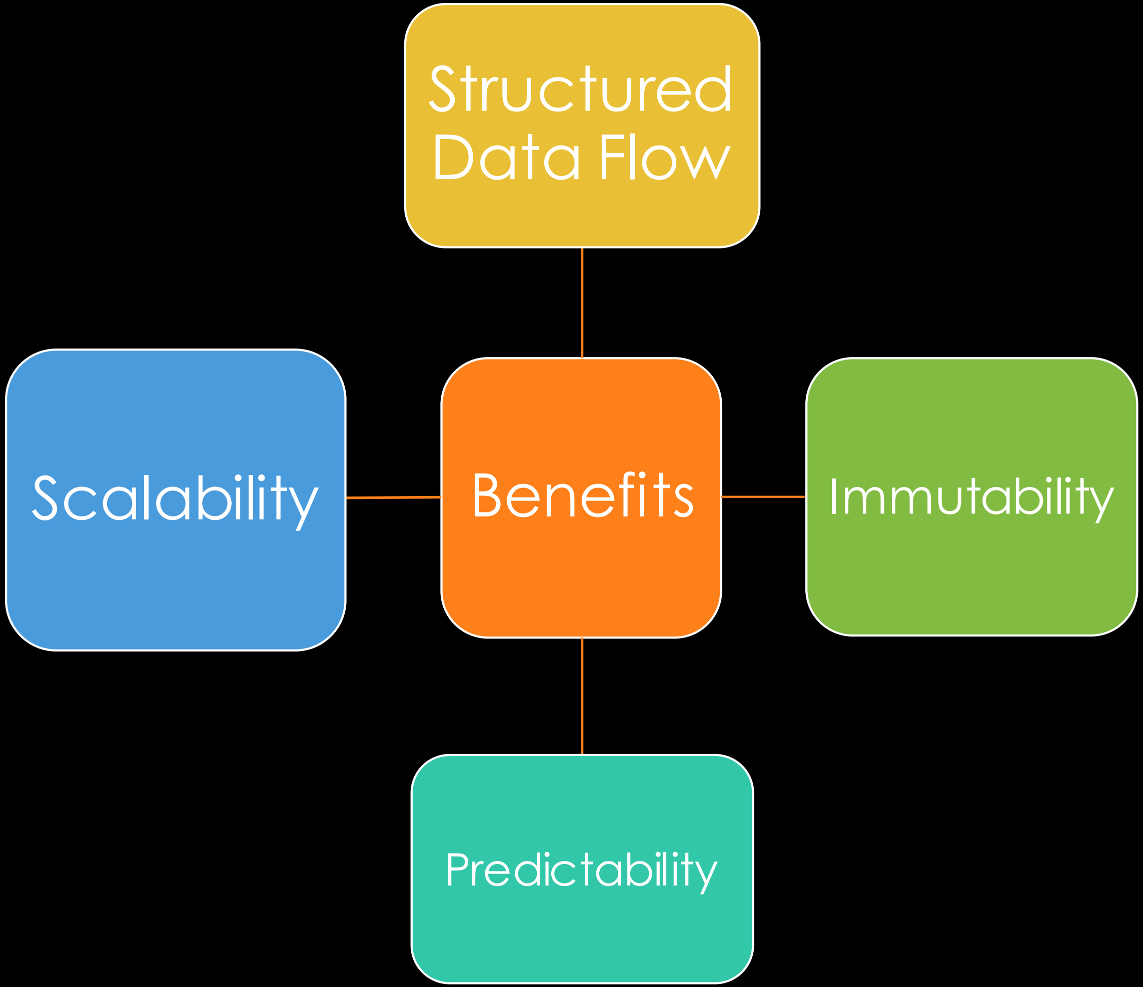
2. **Ng Form:** use with <form> tag.

3. **NgModelGroup:** Create & bind FormGroup instance to a DOM element

REACTIVE FORM

- Reactive forms uses a model-driven approach to handling form inputs.
- Each change to the form state returns a new state, which maintains the integrity of the model between changes.
- Reactive forms are built around observable streams, where form inputs and values are provided as streams of input values, which can be accessed synchronously.

BENEFITS OF USING REACTIVE FORMS



Structured Data Flow

Scalability

Benefits

Immutability

Predictability

HOW TO BUILD A RECTIVE FORM

- Use **RectiveFormsModule** to work with template Driven Forms.
- Directive used:

1. **FormControl:** allows individual form control, monitors value, validation status, and gives a wide variety of public API methods

2. **FormGroup:** It's a group of form fields that have the ability to be manipulated and validated together.

3. **FromBuilder:** helps in making things easy as it helps one to forget about all the new form groups and form controls. FormBuilder replaces FormGroup and FormControls directly and stays as an alternate of both together.

TEMPLATE DRIVEN VS MODEL DRIVEN

Template Driven Forms	Model Driven Forms
Uses FormsModule	Used ReactiveFormsModule
Asynchronous in nature	synchronous in nature
draws logic from the template	the logic resides mainly in the component or typescript code.
used to create simple forms	used to create complex forms.