Angular 8 Forms

Angular forms are used to handle user's input. We can use Angular form in our application to enable users to log in, to update profile, to enter information, and to perform many other data-entry tasks.

In Angular 8, there are 2 approaches to handle user's input through forms:

* Reactive forms
* Template-driven forms

Both approaches are used to collect user input events from the view, validate the user input, create a form model and data model to update, and provide a way to track changes.

Both reactive and template-driven forms share underlying standard building blocks, which are the following.

1. [FormControl](https://angular.io/api/forms/FormControl): It tracks the value and validation status of the individual form control.
2. [FormGroup](https://angular.io/api/forms/FormGroup): It tracks the values and status for the collection of form controls.
3. [FormArray](https://angular.io/api/forms/FormArray):It tracks the same values and status for the array of the form controls.
4. [ControlValueAccessor](https://angular.io/api/forms/ControlValueAccessor): It creates a bridge between Angular FormControl instances and native DOM elements.

## ****Angular 8 Reactive Forms****

Reactive forms or Model-driven forms are more robust, scalable, reusable, and testable. If forms are the key part of your application, or you’re already using reactive patterns for building your web application, use reactive forms.

In Reactive Forms, most of the work is done in the component class.

One of the advantages of the Model-driven approach is that the Validations become very easy.

## ****Angular 8 Template-driven Forms****

Template-driven forms are useful for adding the simple form to an app, such as the email list signup form. They’re easy to add to a web app, but they don’t scale as well as the reactive forms.

If you have the requirement of basic form and logic that can be managed solely in the template, use template-driven forms.

In template-driven forms, most of the work is done in the template.

## Reactive Forms vs. Template-driven Forms

Both Reactive forms and Template-driven forms manage and process data differently. Each offers different advantages.

### Reactive Forms

* Reactive forms are more robust.
* Reactive forms are more scalable, reusable, and testable.
* They are most preferred to use if forms are a key part of your application, or your application is already built using reactive patterns. In both cases, reactive forms are best to use.

### Template-driven Forms

* Template-driven forms are best if you want to add a simple form to your application. **For example:** email list signup form.
* Template-driven forms are easy to use in the application but they are not as scalable as Reactive forms.
* Template-driven forms are mainly used if your application's requires a very basic form and logic. It can easily be managed in a template.

## Using angular CSS classes to highlight invalid form fields

If showing a plain error message is not enough, we can also adjust the appearance of each form-controll using CSS.

For that, angular automatically assigns some classes to each control, depending of its status.

.ng-valid

.ng-invalid

.ng-pending

.ng-pristine

.ng-dirty

.ng-untouched

.ng-touched