

Angular supports two ways forms and validations:

1. Template Driven Forms
2. Reactive Forms

Template Driven Forms or TDF, is achieved by driving all the business logic using in-built attributes (novalidate, #ref, ngForm, ngSubmit, ngModel, required, minlength, maxlength, pattern) inside **mark-up** (.html or templates)

This generates the following structure at run-time: ngForm instance is created and multiple FormControl instances created and these FormControl instances are associated with ngForm instance

```
Angular is running in the development mode. Call enableProdMode() to enable the production mode. core.js:3077
>XHR finished loading: GET "http://localhost:4200/sockjs-node/info?t=1532281677213". zone.js:2969
    >Event {isTrusted: true, type: "submit", target: form.ng-dirty.ng-touched.ng-valid, currentTarget: form.ng-dirty.ng-touched.ng-valid, eventPhase: 2, ...} template-driven-form_ent.component.ts:11
    >NgForm {submitted: true, _directives: Array(2), ngSubmit: EventEmitter, form: FormGroup}
    >control: FormGroup
    >controls: Object
    >email: FormControl {validator: f, asyncValidator: null, _onCollectionChange: f, pristine: false, touched: true, ...}
    >name: FormControl {validator: f, asyncValidator: null, _onCollectionChange: f, pristine: false, touched: true, ...}
    >__proto__: Object
    dirty: (...)
    disabled: (...)
```

Form has one to many form controls

Each form control has zero to many validators(required, minlength, maxlength, pattern)

**Form is considered validated if and only if all the form controls are validated!**