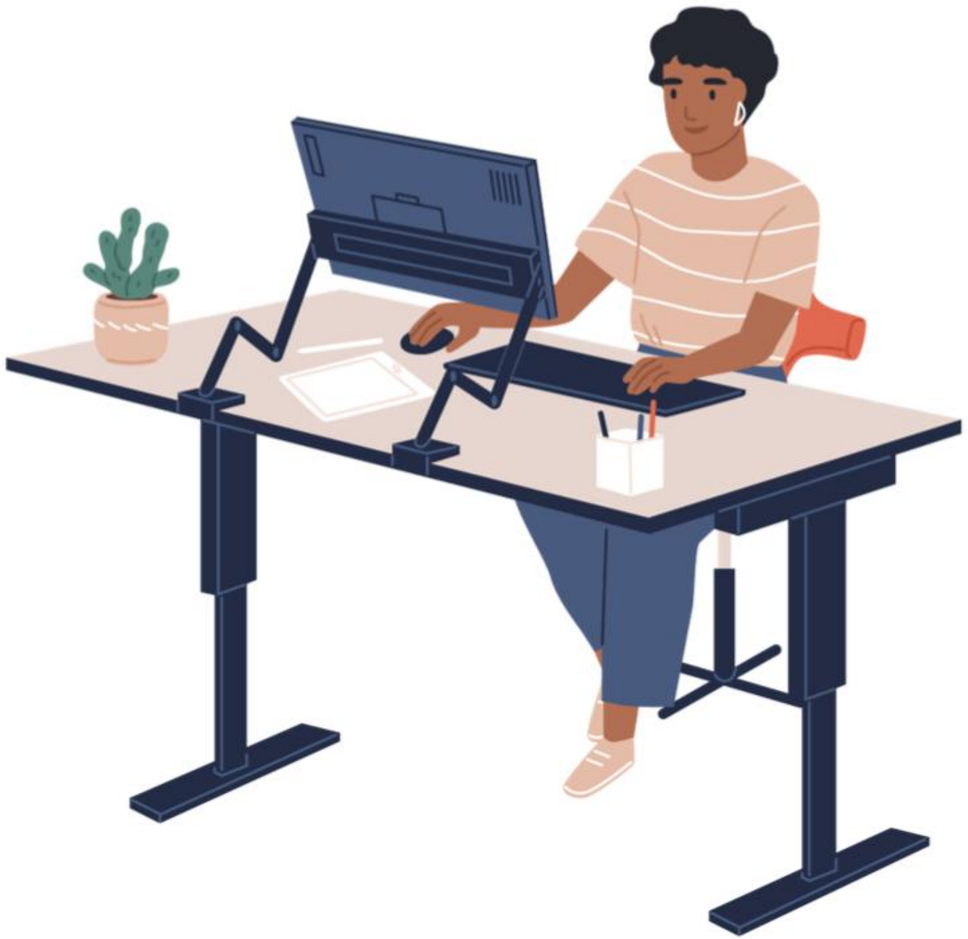
Learning Consolidation

Developing Interactive Template-Driven Forms Inside SPA



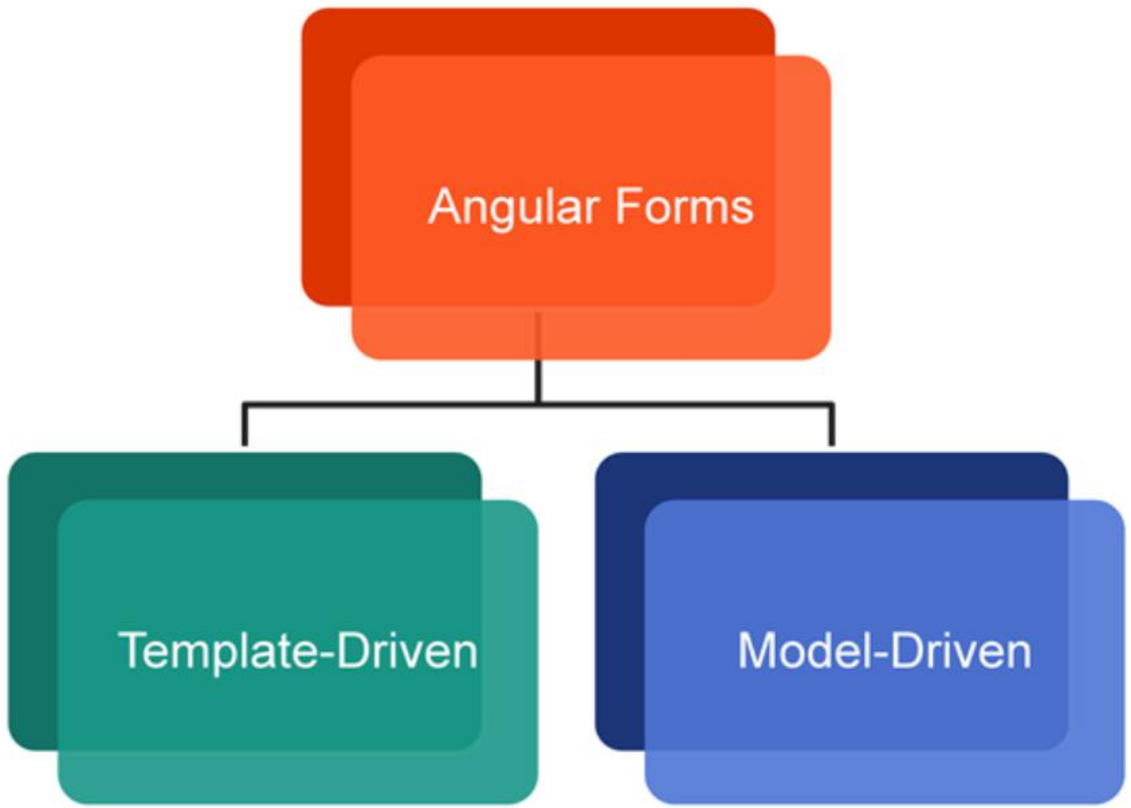
Slide Note

Angular Forms

- Forms are critical to any modern front-end application; they are used to collect data from the user and make the page interactive.
- Forms offer a variety of uses that cater to different types of businesses. Some of the different types of forms are the login form, the search form, the order form, the booking form, the donation form, etc.
- Angular forms:
  - capture user inputs from the view
  - validate user input
  - create a form model and data model to update
  - provide a way to track changes

Slide Note

Different Approaches in Angular Forms



Slide Note

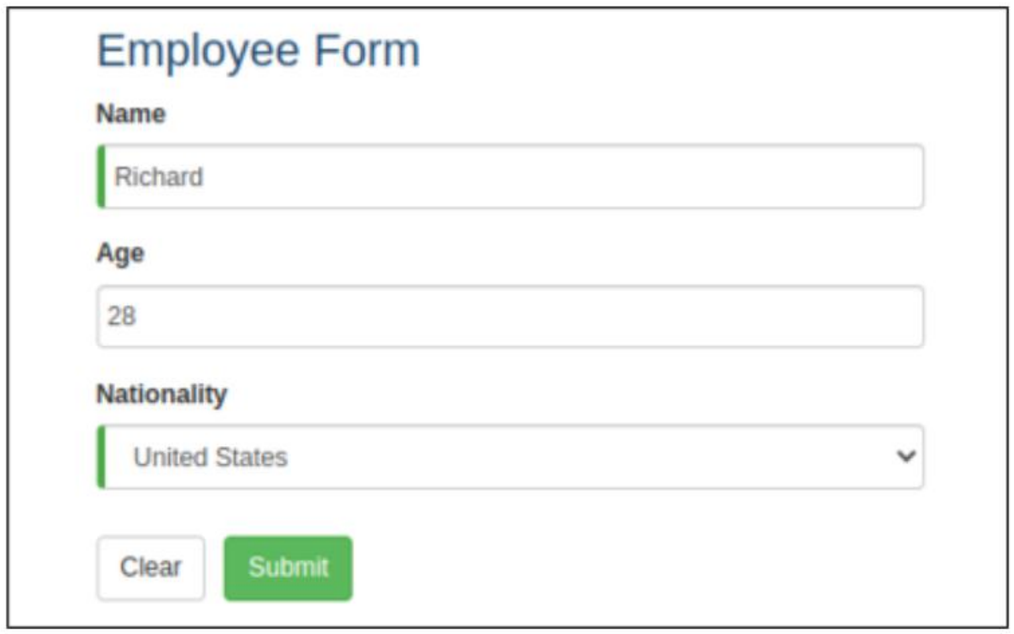
Template-Driven Forms

- Template-driven forms are the simplest forms to create with minimum TypeScript code.
- As the name suggests, form fields, form and form field validations, and form submission can be done in the HTML template itself.
- The Angular framework creates and maintains the data model internally.
- Template-driven forms are best for simple forms like user registration, feedback forms, and log in forms when the complexity of handling form fields is minimum.

Slide Note

Template-Driven Forms: Preview

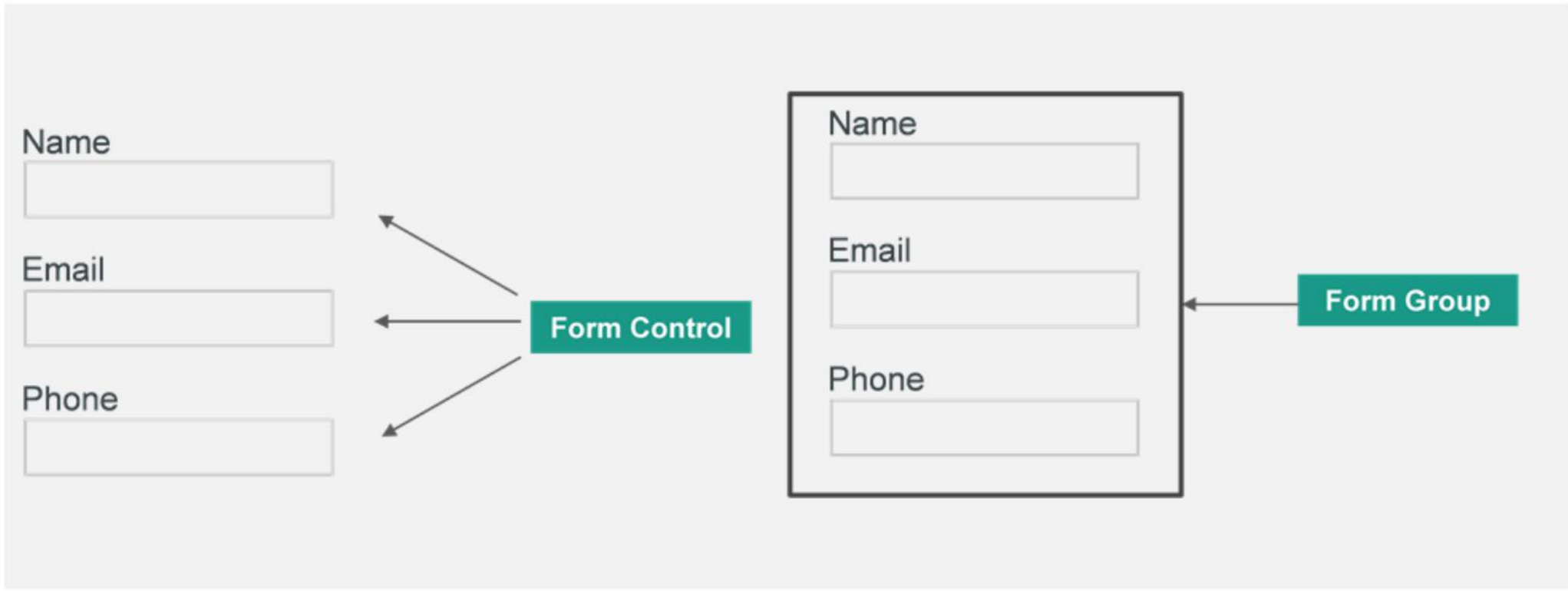
Form With Empty Values



Form With Valid Values

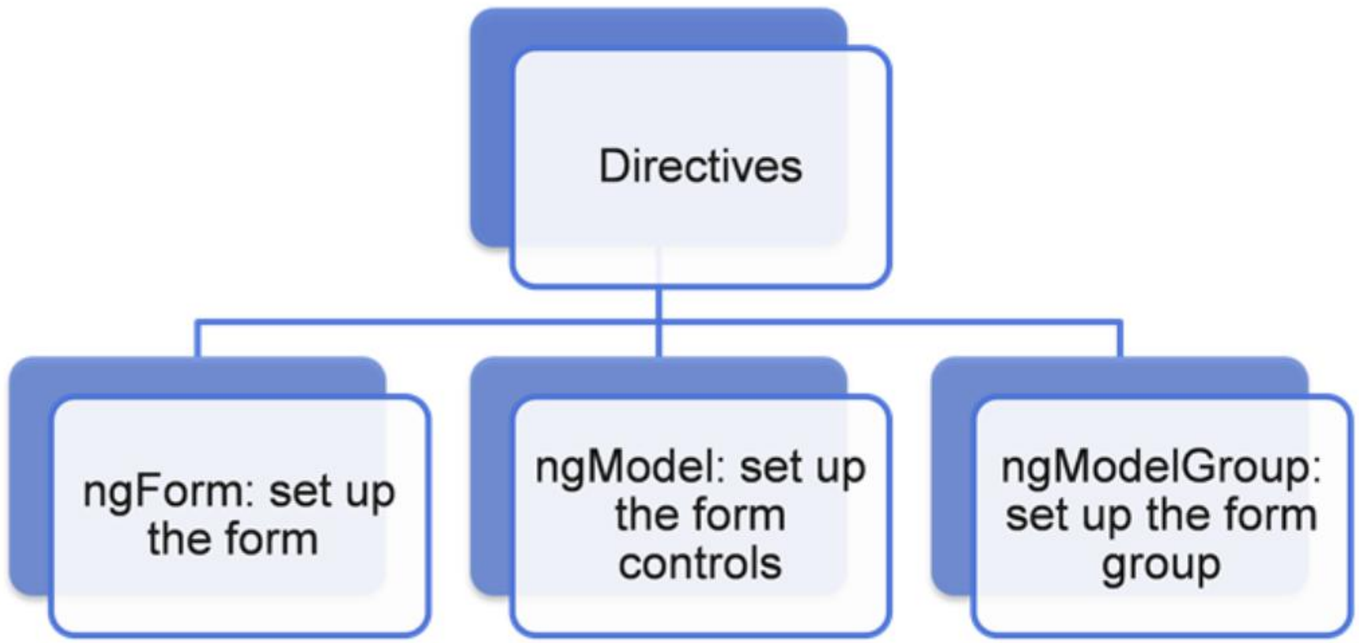
Slide Note

Angular Form Building Blocks



Slide Note

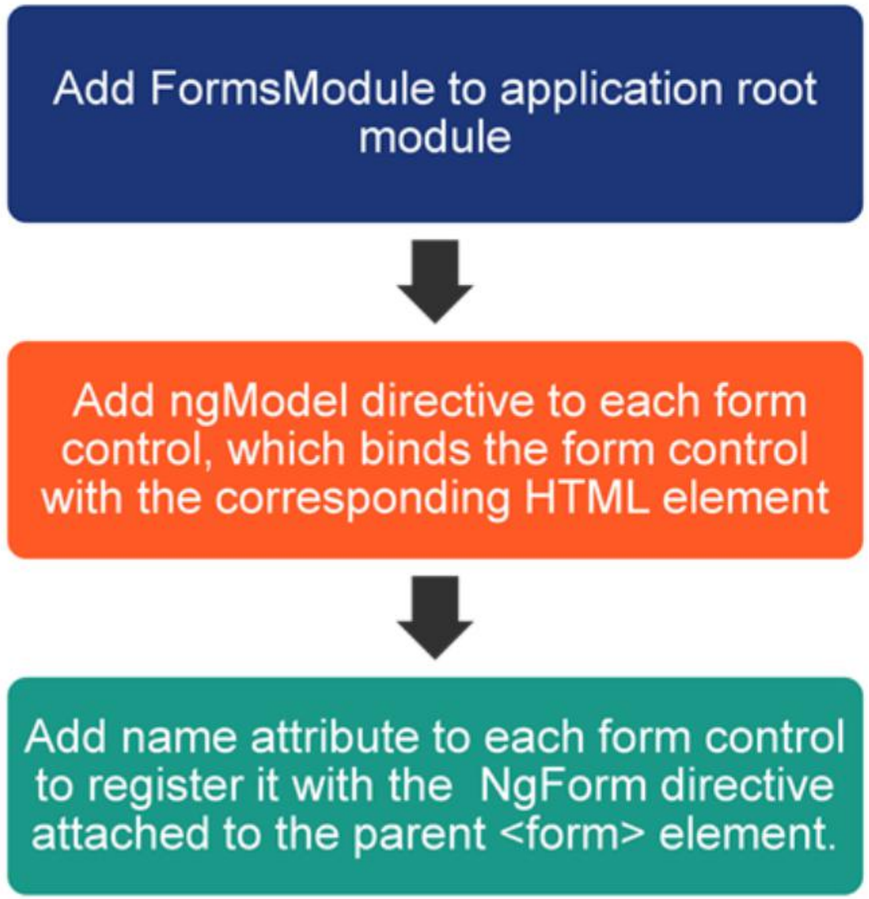
Angular Form Directives



Slide Note

Steps for Creating Template-Driven Forms

- Here are the steps to create template-driven forms:



Slide Note

Tracking Form Status

- The template local variable assigned to the NgForm directive allows us to check the status of the form, like whether it is valid and submitted, and to get the values of the form elements.

Property	Description
Value	Returns the object containing the value of every FormControl
Valid	Returns true if the form is valid else returns false.
Touched	True if the user has entered a value in at least in one field.
Submitted	Returns true if the form is submitted, otherwise returns false.

Slide Note

Pristine means that the user hasn't changed the value since it was displayed in this form.

Tracking Form Control Status

- The NgModel directive on a form control tracks the state of that control.
- It tells us if the user touched the control, if the value changed, or if the value became invalid.
- Angular sets special CSS classes on the control element to reflect the state.

State	Class if true	Class if false
The control has been visited.	ng-touched	ng-untouched
The control's value has changed.	ng-dirty	ng-pristine
The control's value is valid.	ng-valid	ng-invalid

- Angular also applies the ng-submitted class to form elements upon submission, but not to the controls inside the form element.

Slide Note

Tracking Form Control Status Values

#name="ngModel"

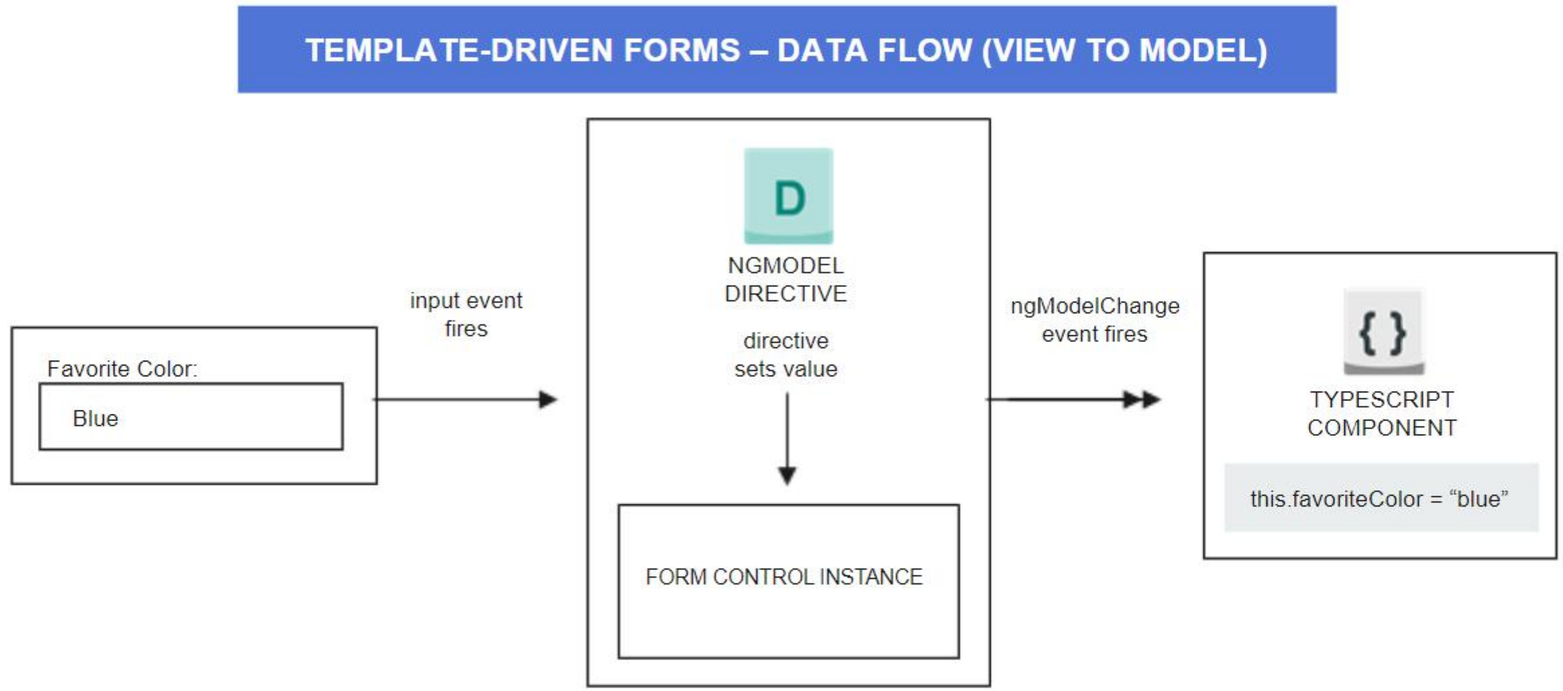
The control name input is valid	name.valid	name.invalid
The control name input value has changed	name.dirty	name.pristine
The control name has been visited	name.touched	name.untouched

The values in green boxes returns true
The values in the red boxes returns false

Control is **dirty**: User changes the value in the field
Control is **touched**: User blurs the form control element

Slide Note

View-to-Model in Template-Driven Forms



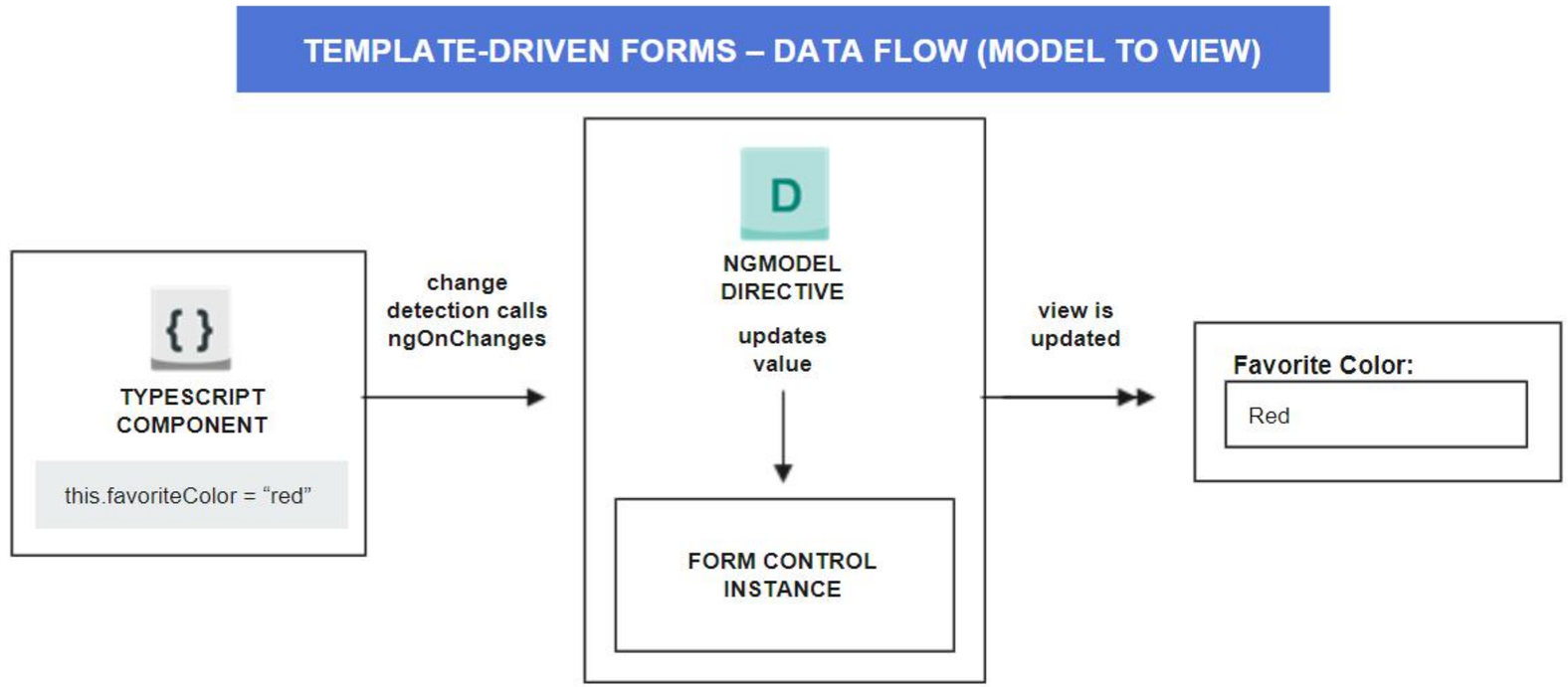
START	
this.favoriteColor (ngModel)	RED
FormControl instance value	RED
view	BLUE

DIRECTIVE SETS VALUE	
this.favoriteColor (ngModel)	RED
FormControl instance value	BLUE
view	BLUE

END RESULT	
this.favoriteColor (ngModel)	BLUE
FormControl instance value	BLUE
view	BLUE

Slide Note

Model-to-View in Template-Driven Forms



START

this.favoriteColor (ngModel)	RED
FormControl instance value	BLUE
view	BLUE

DIRECTIVE UPDATES VALUE

this.favoriteColor (ngModel)	RED
FormControl instance value	RED
view	BLUE

END RESULT

this.favoriteColor (ngModel)	RED
FormControl instance value	RED
view	RED

Slide Note

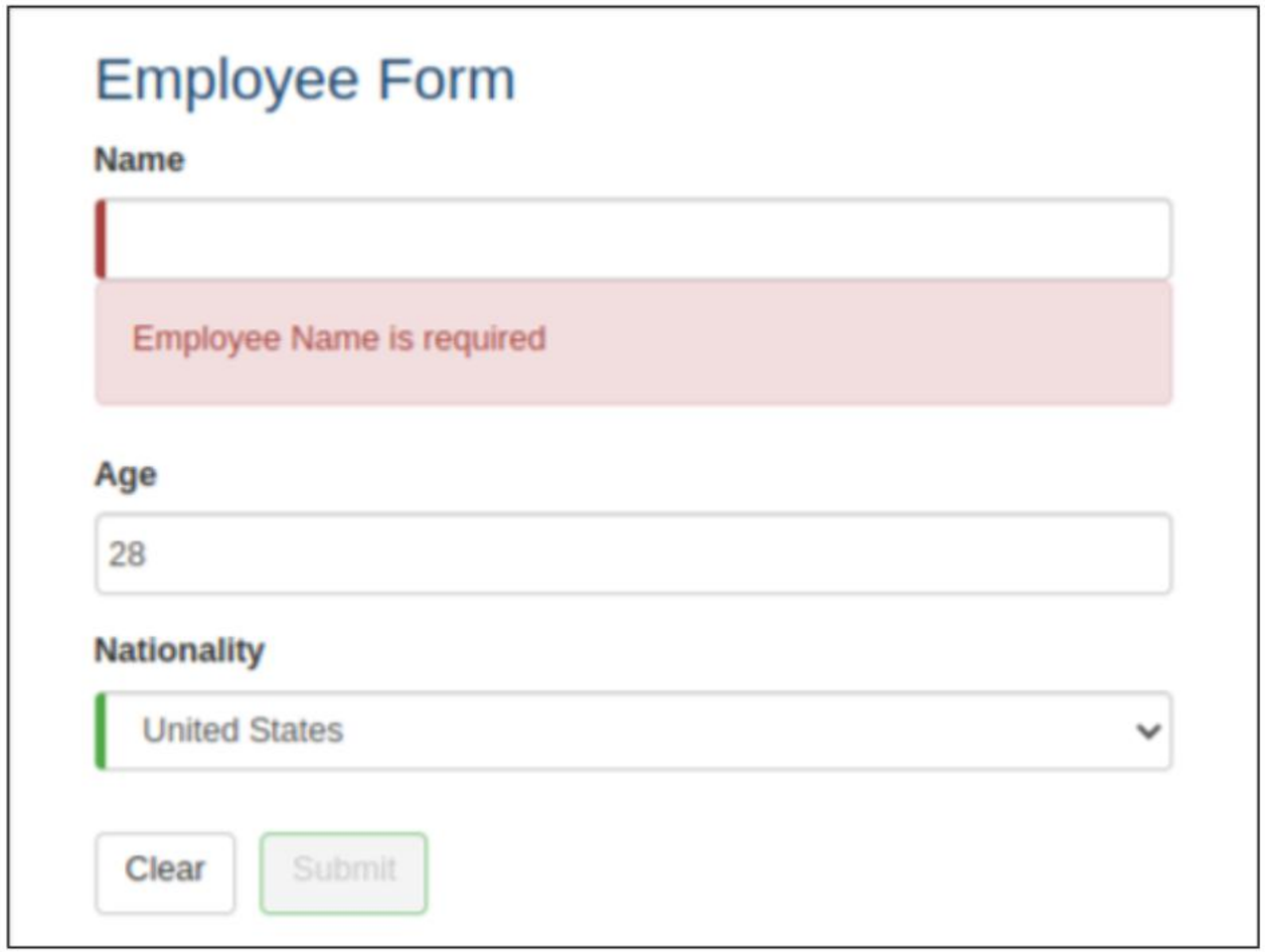
Validating Input in Template-Driven Forms

- The HTML validation attributes like `required, minlength` used for validating user inputs can be used to add validation in template-driven forms.
- Angular uses directives to match HTML native validation attributes with validator functions in the framework.
- Examples of built-in validators: min, max, required, minlength, etc.
- Custom validators can also be defined as validator functions to match business needs. They can be added as a directive to the template later.
- Every time the value of a form control changes, Angular runs validation and generates either a list of validation errors that results in an INVALID status, or null, which results in a VALID status.

Slide Note

Validation error messages are displayed after checking their invalid status.

Form With Validation Errors



Slide Note

Built-in Validator Functions

Validator Functions	Description
min()	It requires the control's value to be greater than or equal to the provided number.
max()	The control's value should be less than or equal to the provided number.
required()	The control's value should be a non-empty value.
email()	The control's value should pass an email validation test.
minLength()	The length of the control's value should be greater than or equal to the provided minimum length.
maxLength()	The length of the control's value should be less than or equal to the provided maximum length.
pattern()	The control's value must match a regex pattern.
nullValidator()	This validator performs no operations.

Slide Note

Self-Check

Which of the following Angular CSS classes is used to indicate that an input element's value has not changed from the time it was displayed in the form?

A. ng-dirty

B. ng-pristine

C. ng-valid

D. ng-touched

Slide Note

Self-Check: Solution

Which of the following Angular CSS classes is used to indicate that an input element's value has not changed from the time it was displayed in the form?

A. ng-dirty

B. **ng-pristine**

C. ng-valid

D. ng-touched

Slide Note

Self-Check

Which of the following module(s) must be imported in Ng Module to get access to template-driven from features?

A. BrowserModule

B. FormsModule

C. TemplateDrivenFormsModule

D. Both option 1 and option 2

Slide Note

Self-Check: Solution

Which of the following module(s) must be imported in NgModule to get access to template-driven form features?

A. BrowserModule
B. **FormsModule**
C. TemplateDrivenFormsModule
D. Both option 1 and option 2

Explanation:

- Option 1 is incorrect: BrowserModule provides services that are essential to launch and run a browser app.
- Options 2 and 3 incorrect: There is no such TemplareDrivenFormsModule.