

1. How to apply validations in AngularJS?

Ans:

- 1) AngularJS provides you built-in validation directives to validate form client side. This provides a better user experience than server-side validation alone because the user gets instant feedback on how to correct the error.
- 2) Keep in mind that while client-side validation plays an important role in providing good user experience, it can easily be circumvented and thus can not be trusted. Server-side validation is still necessary for a secure application.
- 3) AngularJS form validations are based on the HTML5 form validators.
- 4) AngularJS directives for form validation

Here is a list of AngularJS directive which can be applied on an input field to validate its value.

```
<input type="text"
ng-model="{ string }"
[name="{ string }"]
[ng-required="{ boolean }"]
[ng-minlength="{ number }"]
[ng-maxlength="{ number }"]
[ng-pattern="{ string }"]
[ng-change="{ string }"]>
</input>
```

2. Explain the Angular Form Properties \$pristine, \$dirty, \$valid, \$invalid?

Ans:

\$pristine : This property is used for checking whether the form elements are modified. Returns true if form input elements are unmodified and vice versa.

Ex: formName.fieldName.\$pristine

And we can set the form in pristine mode. That means form input fields will get reset.

Ex: formName.\$setPristine();

\$dirty: This property acts exactly in reverse condition as \$pristine. Where \$dirty returns true when form input fields are modified and vice versa.

Ex:

- formName.dirty
- formName.fieldName.\$dirty

\$valid: This property is used for checking whether the form input elements are valid or not. Returns true if all form input elements are valid and vice versa.

Ex:

formName.\$valid

formName.fieldName.\$valid

\$invalid : this property is used for checking whether form input elements are valid or not. Returns true if all form input elements are invalid.

Ex:

formName.\$invalid.

formName.fieldName.\$invalid.

3. Explain about \$touched and \$untouched angular form properties?

Ans:

\$touched: This property is used to know whether the input fields in the form are blurred / focussed or not. If the fields lost focussed then it return true and vice versa.

Ex:

formName.\$touched

formName.fieldName.\$touched

\$untouched: This property is used to know whether the input fields in the form are blurred / focussed or not. If the fields are focussed then it returns true and vice versa.

Ex:

formName.\$untouched

formName.fieldName.\$untouched

4. What is the importance of ng-model-options?

Ans: By default any change to the content will trigger a model update and form validation. This directive allows you to modify the behaviour of ngModel directives within your application. You can specify an ngModelOptions directive on any element.

Following are the option properties to override default settings:

- 1) updateOn -----> ng-model-options="{ updateOn: 'blur' }"
- 2) debounce -----> ng-model-options="{ debounce: { 'default': 500, 'blur': 0 } }"
- 3) getterSetter -----> ng-model-options="{ getterSetter: true }"
- 4) allowInvalid -----> ng-model-options="{ getterSetter: true, allowInvalid: true }"
- 5) timezone -----> ng-model-options="{ timezone: 'UTC' }" />

5. What is Custom Validation in AngularJS?

Ans: By using custom validations one can customize or override the predefined validation functions given by angular. For custom validation we have to do is build a directive that requires ngModel and you can add validation functions to get control over the ng-controller.

Each function receives the model value and view value as params. These validation functions will be called when the inputs are changed or when the bound model changes.

6. Why angular 1 form validations do not work without ng-models?

Ans:

ngModel directive holds an instance of NgModelController containing the services for data-binding, validation, CSS updates, and value formatting and parsing. If ngModel itself is not there, validation will not work. For form validation to work you need both form instance which can be published to scope using name attribute and the form input control (ngModel).