FormGroup

Tracks the value and validity state of a group of [FormControl](https://angular.io/api/forms/FormControl) instances.

Ng Submit

This event is normally used to send information which the user might have entered on a web page to the server for further processing.

<div>

<input type="text" [formControl]="myForm.controls.firstName"/>

<input type="text" [formControl]="myForm.controls.lastName"/>

<input type="text" [formControl]="myForm.controls.email"/>

<input type="text" [formControl]="myForm.controls.title"/>

</div>

Is equivalent to:

<div [formGroup]="myForm">

<input type="text" formControlName="firstName"/>

<input type="text" formControlName="lastName"/>

<input type="text" formControlName="email"/>

<input type="text" formControlName="title"/>

</div>

formControlName assigns a string for the forms module to look up the control by name.

**Component**

@Component({

...

})

export class ProfileEditorComponent {

profileForm = new FormGroup({

firstName: new FormControl(''),

lastName: new FormControl(''),

});

}

**Template**

<form [formGroup]="profileForm">

<label>

First Name:

<input type="text" formControlName="firstName">

</label>

<label>

Last Name:

<input type="text" formControlName="lastName">

</label>

</form>

NgClass

DIRECTIVE

Adds and removes CSS classes on an HTML element.

| required() [mode\_edit](https://github.com/angular/angular/edit/master/packages/forms/src/validators.ts?message=docs(forms)%3A%20describe%20your%20change..." \l "L136-L157" \o "Suggest Edits)[code](https://github.com/angular/angular/tree/7.2.0-rc.0/packages/forms/src/validators.ts" \l "L136-L157" \o "View Source) |
| --- |
| Validator that requires the control have a non-empty value. |

JSON

A common use of JSON is to exchange data to/from a web server.

When sending data to a web server, the data has to be a string.

Convert a JavaScript object into a string with JSON.stringify().

<!DOCTYPE html>

<html>

<body>

<h2>Create JSON string from a JavaScript object.</h2>

<p id="demo"></p>

<script>

var obj = { name: "John", age: 30, city: "New York" };

var myJSON = JSON.stringify(obj);

document.getElementById("demo").innerHTML = myJSON;

</script>

</body>

</html>

minlength Validation

We can use its selector minlength with formControlName, formControl and ngModel in HTML template. Validators.minLength can be passed in FormControl while creating FormGroup.

In reactive form we need to use Validators.minLength with FormControl while creating FormGroup.

userName: new FormControl('', Validators.minLength(5))

We will write code in HTML template as following.

<input formControlName="userName">

To display validation error message, we need to writer a getter method as given below.

get userName() {

return this.userForm.get('userName');

}

Now display validation error message as following.

<div \*ngIf="userName.errors?.minlength">

Name must be at least 5 characters long.

</div>

To validate minimum length in reactive form we can also use minlength attribute with formControlName as following.

<input formControlName="userName" minlength="5">

we want to add a validators that makes sure that firstname and lastname is set. Angular comes with a Validators class that has some common validators built-in

firstName: ['', Validators.required],

lastName: ['', Validators.required],

We can access the validity state of a form control like

<form [formGroup]="registerForm">

<label>Firstname:</label>

<input type="text" formControlName="firstname">

<p \*ngIf="registerForm.controls.firstname.errors">This field is required!</p>

...

</form>