

# Inject Service Into Validator in Angular

1 Comment / 4 minutes of reading / March 9, 2023



[Angular Tutorial](#)

[Validation in Template Driven](#)

[Forms](#)

[Passing Parameter to Custom](#)

[Validator](#)



Learn how to inject service into Validator in Angular. The validator may depend on some external service to do its validation. For Example, it may need to fetch data from the back end server to validate the value.

This is a continuation of our following tutorials

1. [The custom validator in Angular](#)
2. [Angular custom Validator with Parameters](#)

## Table of Contents

[Custom Validator](#)

[Validator Service](#)

[Injecting Service](#)

[Wrapper Service](#)

[Inject Service directly into the Validator](#)

[Summary](#)

## Custom Validator

Here is our greater than [custom validator](#) gte from the previous tutorial. The gte is a function that returns [ValidatorFn](#) function.

```
1
2 import { AbstractControl, ValidationErrors, ValidatorFn } from '@angular/forms'
3
4 export function gte(val: number): ValidatorFn {
5
6   return (control: AbstractControl): ValidationErrors | null => {
7
8     let v: number = +control.value;
9
10    if (isNaN(v)) {
11      return { 'gte': true, 'requiredValue': val }
12    }
13
14    if (v <= +val) {
15      return { 'gte': true, 'requiredValue': val }
16    }
17
18    return null;
19  }
20 }
21
```

We pass the parameter to the validator in the component class as shown below.

```
1
2 myForm = new FormGroup({
3   numVal: new FormControl('', [gte(10)]),
4 })
5
```

## Validator Service

Let us now see how we can inject service into the above validator. First, move the logic from the gte validator to a separate service.

Create a new service gte.service.ts

```
1
2 import { Injectable } from '@angular/core';
3
4 @Injectable({
5   providedIn: 'root',
6 })
7 export class gteService {
8
9   gte(num:any, requiredValue:Number) : Boolean {
10
11     if (isNaN(num)) {
12       return false;
13     }
14
15     if (num <= +requiredValue) {
16       return false;
17     }
18
19     return true;
20   }
21 }
22
```

The gteService is very simple.

The gte method takes val and requiredValue as the parameter. It checks if the val is a number and is greater than requiredValue . If yes returns true else returns false .

## Injecting Service

There are two ways you can inject service into the validator. One is to create a wrapper service. The other option is to inject service directly into the validator.

## Wrapper Service

Open the `gte.validator.ts`. Create the `gteValidatorService` class. In the constructor inject the `gteService`. Copy the validator function `gte` into the class and use the `gteService` as shown below

```
1
2 import { AbstractControl, ValidationErrors, ValidatorFn, Validator, NG_VALIDATORS, FormControl } from '@angular/forms';
3 import { gteService } from './gte.service';
4 import { Directive, OnInit, forwardRef, Input, Injectable } from '@angular/core';
5
6 @Injectable({
7   providedIn: 'root',
8 })
9 export class gteValidatorService {
10
11   constructor(private gteService: gteService) {
12   }
13
14   gte(val: number): ValidatorFn {
15
16     return (control: AbstractControl): ValidationErrors | null => {
17
18       let v: number = +control.value;
19       if (!this.gteService.gte(v, val)) {
20         return { 'gte': true, 'requiredValue': val };
21       }
22
23       return null;
24     }
25   }
26 }
27
```

Inject the `gteValidatorService` in the component class and use the validator as shown below.

```
1
2 import { Component } from '@angular/core';
3 import { FormGroup, FormControl, AbstractControl, ValidationErrors, ValidatorFn } from '@angular/forms';
4 import { gteValidatorService } from './gte.validator';
5
6 @Component({
7   selector: 'app-root',
8   templateUrl: './app.component.html',
9   styleUrls: ['./app.component.css']
10 })
11 export class AppComponent {
12
13   constructor(private gteValidator: gteValidatorService) {
14   }
15
16
17   myForm = new FormGroup({
18     numVal: new FormControl("", [this.gteValidator.gte(10)]),
19   })
20
21   get numVal() {
22     return this.myForm.get('numVal');
23   }
24
25
26   onSubmit() {
27     console.log(this.myForm.value);
28   }
29 }
30
31
```

## Inject Service directly into the Validator

Another option is to directly inject the service into the Validator as shown below.

```
1
2 import { AbstractControl, ValidationErrors, ValidatorFn, Validator, NG_VALIDATORS, FormControl } from '@angular/forms';
3 import { gteService } from './gte.service';
4 import { Directive, OnInit, forwardRef, Input, Injector } from '@angular/core';
5
6 export function gte(val: number): ValidatorFn {
7
8   return (control: AbstractControl): ValidationErrors | null => {
9
10    let injector = Injector.create([ { provide: gteService, useClass: gteService, deps: [] }]);
11    let service = injector.get(gteService);
12
13    let v: number = +control.value;
14    if (!service.gte(v, val)) {
15      return { 'gte': true, 'requiredValue': val };
16    }
17
18    return null;
19  }
20 }
21 }
22
```

We use the Injector to inject the instance of the service.

```
1
2 let injector = Injector.create([ { provide: gteService, useClass: gteService, deps: [] }]);
3 let service = injector.get(gteService);
4
```

## Summary

In this article, we learned how to inject the service into the validator function. One option is to create a Validator service. Copy the validator function into the Validator service. Another option is to inject the service directly into the validator function using the inject method.