

Sql server, .net and c# video tutorial

Free C#, .Net and Sql server video tutorial for beginners and intermediate programmers.

[Support us](#) [.Net Basics](#) [C#](#) [SQL](#) [ASP.NET](#) [ADO.NET](#) [MVC](#) [Slides](#) [C# Programs](#) [Subscribe](#) [Buy DVD](#)

Angular custom validator example template driven forms

Suggested Videos

[Part 22 - Angular checkbox validation](#) | [Text](#) | [Slides](#)

[Part 23 - Angular select list validation](#) | [Text](#) | [Slides](#)

[Part 24 - Angular value vs ngValue](#) | [Text](#) | [Slides](#)

In this video, we will discuss creating a **custom validator in angular with an example**. We will make this validator reusable and configurable. Along the way, we will also learn how to create a **custom directive in angular**.

Example : When you have a default option like **"Select Department"** in a SELECT list, the REQUIRED validation does not work, if the default option value is anything else other than NULL. So if the default option value is not null, then that default option is also treated as a valid selection, and we do not get to see the required validation error. If "Select Department" is selected, we want the validation to fail, and display **"Department is required"** validation error message. To make this work the way we want, we implement a custom validator.

PRAGIM
TECHNOLOGIES
Training + Placements

Best software training and placements in marathahalli, bangalore. For further details please call 09945699393.

Complete Tutorials

[JavaScript tutorial](#)

[Bootstrap tutorial](#)

[Angular tutorial for beginners](#)

[Angular 5 Tutorial for beginners](#)

Important Videos

[The Gift of Education](#)

[Web application for your business](#)

Department

Select Department ▼

Department is required

Create a custom Directive

To use a custom validator in template driven forms, we have to create the validator as a directive. Once we have the directive created, we can then use that directive as an attribute on the select element that we want to validate. This is going to be a configurable and reusable validator. We can use it with any SELECT list in an angular application. So create a **"shared"** folder in the **"app"** folder. In the "shared" folder create a file with name "select-required-validator.directive.ts". Copy and paste the following code.

```
import { Validator, AbstractControl, NG_VALIDATORS } from '@angular/forms';
import { Directive } from '@angular/core';

@Directive({
  selector: '[appSelectValidator]',
  providers: [
    {
      provide: NG_VALIDATORS,
      useExisting: SelectRequiredValidatorDirective,
      multi: true
    }
  ]
})
export class SelectRequiredValidatorDirective implements Validator {
  validate(control: AbstractControl): { [key: string]: any } | null {
    return control.value === '-1' ? { 'defaultSelected': true } : null;
  }
}
```

Code Explanation :

Since we are creating a directive, we decorate the class with **@Directive** decorator

[How to become .NET developer](#)

[Resources available to help you](#)

Dot Net Video Tutorials

[ASP.NET Core Tutorial](#)

[Angular 6 Tutorial](#)

[Angular CRUD Tutorial](#)

[Angular CLI Tutorial](#)

[Angular 2 Tutorial](#)

[Design Patterns](#)

[SOLID Principles](#)

[ASP.NET Web API](#)

[Bootstrap](#)

[AngularJS Tutorial](#)

[jQuery Tutorial](#)

[JavaScript with ASP.NET Tutorial](#)

[JavaScript Tutorial](#)

[Charts Tutorial](#)

[LINQ](#)

[LINQ to SQL](#)

[LINQ to XML](#)

[Entity Framework](#)

NG_VALIDATORS is a collection of validators. It already contains all the built-in validators like **required**, **pattern** etc. Before we can use our custom validator we have to add it to the list of validators by adding it to NG_VALIDATORS token. To specify that we want to add our validator to the list of validators, we set **multi** property to true

```
providers: [  
  {  
    provide: NG_VALIDATORS,  
    useExisting: SelectRequiredValidatorDirective,  
    multi: true  
  }  
]
```

Implement **Validator** interface as we are creating a custom validator
export class SelectRequiredValidatorDirective **implements** Validator

Since we are implementing validator interface, we have to provide implementation for the interface **validate()** method. This method has one input parameter and its type is **AbstractControl**. AbstractControl extends both **FormControl** and **FormGroup**. In some cases you may want to validate a FormGroup instead of a single FormControl. So to cater for both scenarios, the parent type - AbstractControl is specified. **This function returns an object if the validation fails or null if the validation succeeds.** The object that is returned when the validation fails contains a key/value pair. The key is a string and the value can be anything.

```
validate(control: AbstractControl): { [key: string]: any } | null
```

If the selected value in the SELECT list is the default value (-1), then we return an object with key **'defaultSelected'** and value of **true** to indicate that the validation has failed. Otherwise we return **NULL** to indicate validation succeeded. In the HTML we can use the "defaultSelected" key to display the validation error specific to this custom validator.

```
return control.value === '-1' ? { 'defaultSelected': true } : null;
```

Import the custom directive in a module where you want to use it.

At the moment we only have one module - Root module. So in app.module.ts file include the following import statement

[WCF](#)[ASP.NET Web Services](#)[Dot Net Basics](#)[C#](#)[SQL Server](#)[ADO.NET](#)[ASP.NET](#)[GridView](#)[ASP.NET MVC](#)[Visual Studio Tips and Tricks](#)[Dot Net Interview Questions](#)

Slides

[Entity Framework](#)[WCF](#)[ASP.NET Web Services](#)[Dot Net Basics](#)[C#](#)[SQL Server](#)[ADO.NET](#)[ASP.NET](#)

```
import { SelectRequiredValidatorDirective } from './shared/select-required-validator.directive';
```

Also include `SelectRequiredValidatorDirective` in the declarations array of the `NgModule()` decorator

Using the custom required validator on the SELECT element

Modify the "Department" SELECT element in `create-employee.component.html` file as shown below.

```
<div class="form-group"
  [class.has-error]="department.touched && department.errors?.defaultSelected">
  <label for="department" class="control-label">Department</label>
  <select id="department" #department="ngModel" name="department"
    [(ngModel)]="employee.department" appSelectValidator
    class="form-control">
    <option value="-1">Select Department</option>
    <option *ngFor="let dept of departments" [value]="dept.id">
      {{dept.name}}
    </option>
  </select>
  <span class="help-block"
    *ngIf="department.touched && department.errors?.defaultSelected">
    Department is required
  </span>
</div>
```

Code Explanation :

- `[class.has-error]="department.touched && department.errors?.defaultSelected"`. Notice, in this conditional class binding we are using the key `"defaultSelected"` to style the Department SELECT element and it's label when there is a validation error. This key is set by the required custom validator when the validation fails.
- `*ngIf="department.touched && department.errors?.defaultSelected"`. Notice here also we are using the key `"defaultSelected"` to display the validation

[GridView](#)[ASP.NET MVC](#)[Visual Studio Tips and Tricks](#)

Java Video Tutorials

[Part 1 : Video | Text | Slides](#)[Part 2 : Video | Text | Slides](#)[Part 3 : Video | Text | Slides](#)

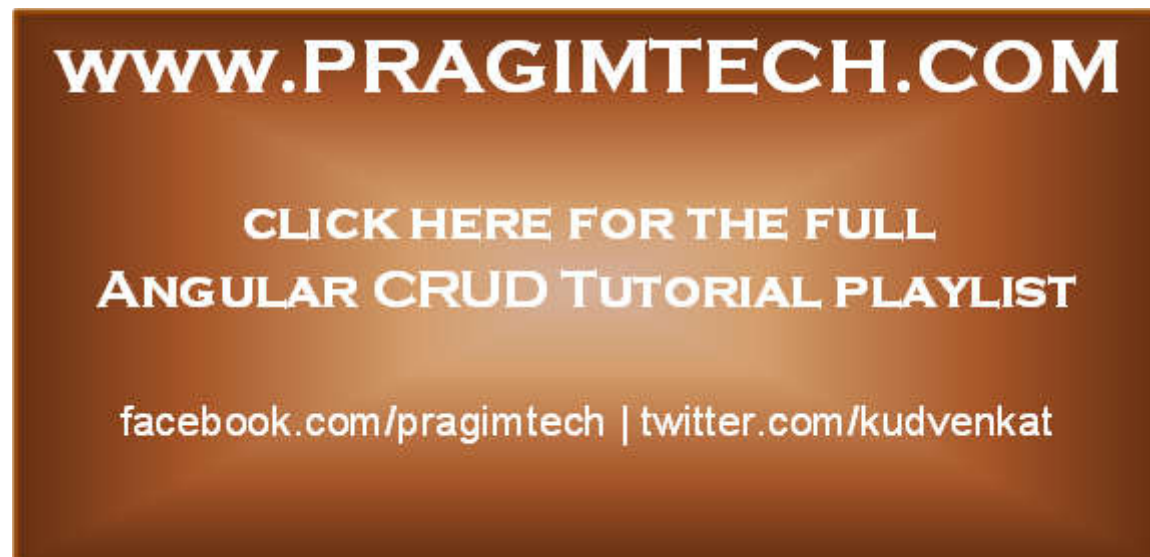
Interview Questions

[C#](#)[SQL Server](#)[Written Test](#)

error message.

- appSelectValidator is the selector we gave for our custom validation directive and we are using it as an attribute on the SELECT list that we want to validate.

At the moment, we can only use this custom validator with a SELECT list whose default option value is -1. We will discuss how to make this custom validator configurable and reusable in our next video.



1 comment:



Nanda Kishore September 22, 2018 at 10:49 AM

Hi Venkat,
here validation is fired when the mouse is clicking on outside of the control as we select 'Select Department' option at first time.

But i have select any other department after the completion of page loading(first time) and after that i am flipped to 'Select Department' option it is not firing validation.(Click outside the control it fires the validation).

this is happens at first time only. After that when we select 'Select Department' option it is working properly.

how to resolve this issue ?

Can you please share knowledge on this query... ?

[Reply](#)

Enter your comment...



Comment as:

toilati123vn@g ▼

[Publish](#)

[Preview](#)

If you like this website, please share with your friends on facebook and Google+ and recommend us on google using the g+1 button on the top right hand corner.

Links to this post

[Create a Link](#)

[Newer Post](#)

[Home](#)

[Older Post](#)

Subscribe to: [Post Comments \(Atom\)](#)

Powered by Blogger.