Basics

Creating Custom Routing Constraint In ASP.N   Core MVC

Nemi Chand   Jun 25 2017   25.5k   7   4

In this article, we'll learn how to create a custom routing constraint in ASP.NET Core MVC. Routing constraint is responsible for matching the constraint rules with incoming requests.

Download Free .NET & JAVA Files API
Try Free File Format APIs for Word/Excel/PDF

Introduction

In this article, we'll learn how to create custom routing constraint in ASP.NET Core MVC. The routing plays a vital role in MVC framework. It takes care of how a request is handled and mapped to the route handler. This is configured in the start up class of configure method. Routing constraint is responsible for matching the constraint rules with incoming request. It returns yes/no (true /false) response depending on the incoming request. If the request is matched, then it serves with the appropriate response/content; else it returns 404 error.

Constraint Usages

1. It helps to avoid the unnecessary requests on url where we are restricted to route value pattern. Let's say - in action parameter, only integer values are allowed; so except integers, all the requests are rejected by the route constraint and a 404 (Page not found) response is returned to users.

IRouteConstraint interface

It defines the contract that a class must implement in order to check whether a URL parameter value is valid for a constraint or not.

This interface contains only single API in order to match the constrains.

```
01.  //https://github.com/aspnet/Routing/blob/dev/src/Microsoft.AspNetCore.Routing.Abstractions/IRouteCons
02.  namespace Microsoft.AspNetCore.Routing
03.  {
04.      /// <summary>
05.      /// Defines the contract that a class must implement in order to check whether a URL parameter
06.      /// value is valid for a constraint.
07.      /// </summary>


10.          /// <summary>
11.          /// Determines whether the URL parameter contains a valid value for this constraint.
12.          /// </summary>
13.          /// <param name="httpContext">An object that encapsulates information about the HTTP request.
     </param>
14.          /// <param name="route">The router that this constraint belongs to.</param>
15.          /// <param name="routeKey">The name of the parameter that is being checked.</param>
16.          /// <param name="values">A dictionary that contains the parameters for the URL.</param>
17.          /// <param name="routeDirection">
18.          /// An object that indicates whether the constraint check is being performed
19.          /// when an incoming request is being handled or when a URL is being generated.
20.          /// </param>
21.          /// <returns>
     <c>true</c> if the URL parameter contains a valid value; otherwise, <c>false</c>.</returns>
22.          bool Match(
23.              HttpContext httpContext,
24.              IRouter route,
25.              string routeKey,
26.              RouteValueDictionary values,
27.              RouteDirection routeDirection);
```

Basics

◄   ǫme/aℂmember   Login

Contribute   Ask Question

The Match method has 5 parameters . Let's discuss each, in details.

1. *HttpContext* encapsulates all http specific information about an http request like request, response, sessions, and more. You can check the httpcontext as well if the request is authenticated or not, and take the appropriate action accordingly.

2. *IRouter* is the router which belongs to constraints.

3. *RouteKey* is the name of the parameter that is being checked. The same name is defined in route template.

4. *Routevalues* contain the parameters of the URL.

5. *RouteDirection* indicates whether ASP.NET routing is processing a URL from an HTTP request or generating a URL. This is

   i. IncomingRequest
      A URL from a client is being processed.

   ii. UrlGeneration
      A URL is being created based on the route definition.

ConstraintMap

It is a key value pair dictionary which contains the list of route constraints. The routing framework adds the list of default constraints with their types. You can learn more about default constraints here on github.

https://github.com/aspnet/Routing/blob/dev/src/Microsoft.AspNetCore.Routing/RouteOptions.cs#L41-L71

You have to add custom route constraint in this dictionary with the help of route options.

Here, I'm going to create an alphanumeric constraint which checks if the incoming request param
value or not.

Alphanumeric route constraint is implemented from the IRouteConstraint interface.

```
01.  public class AlphaNumericConstraint : IRouteConstraint
02.      {
03.          private static readonly TimeSpan RegexMatchTimeout = TimeSpan.FromSeconds(10);
04.
05.          public bool Match(HttpContext httpContext,
06.              IRouter route,
07.              string routeKey,
08.              RouteValueDictionary values,
09.              RouteDirection routeDirection)
10.          {
11.              //validate input params
12.              if (httpContext == null)
13.                  throw new ArgumentNullException(nameof(httpContext));
14.
15.              if (route == null)
16.                  throw new ArgumentNullException(nameof(route));
17.
18.              if (routeKey == null)
19.                  throw new ArgumentNullException(nameof(routeKey));
20.
21.              if (values == null)
22.                  throw new ArgumentNullException(nameof(values));
23.
24.              object routeValue;
25.
26.              if(values.TryGetValue(routeKey, out routeValue))
27.              {
28.                  var parameterValueString = Convert.ToString(routeValue, CultureInfo.InvariantCulture)
29.                  return new Regex(@"^[a-zA-Z0-9]*$",
30.                              RegexOptions.CultureInvariant
31.                              | RegexOptions.IgnoreCase, RegexMatchTimeout).IsMatch(parameterVal
```

Basics

```
34.                return false;
35.            }
36.        }
```

I have alphanumeric regex to check the incoming request parameter that contains the a-z or 0-9.

Add your constraint into *ConstraintMap* using route option in configure services method in the startup class. Here, you have to add route key with the type which is created above.

```
01.  public void ConfigureServices(IServiceCollection services)
02.          {
03.              // Add framework services.
04.              services.AddMvc();
05.
06.              // add here your route constraint
07.              services.Configure<RouteOptions>(routeOptions =>
08.              {
09.                  routeOptions.ConstraintMap.Add("alphanumeric", typeof(AlphaNumericConstraint));
10.              });
11.          }
```

Add your constraint in route template here. You have to use ":" seperator in between route parameter and constraints. The format looks like *{parameter name: route constraint name }.*

```
01.  app.UseMvc(routes =>
02.          {
03.              routes.MapRoute(
04.                  name: "default",
05.                  template: "{controller=Home}/{action=Index}/{id:alphanumeric}");
06.          });
```
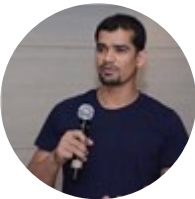
Summary

In this article, you have learned how to create custom route constraints. I've demonstrated an example for the same to help you understand it more clearly.

Basics

Custom Route Constraints in ASP.Net MVC 5

This article describes Custom Route Constraints in ASP.Net MVC 5 with parameter constrain name after the parameter name separated by a colon.

Constraint　　　IRouteConstraint　　　Route Constraint

Nemi Chand　*TOP 500*

Microsoft® Certified Professional,Pragmatic

https://www.c-sharpcorner.com/members/nemi-chand-jat

468　　　**562.3k**　　　　　**1**

4　　　**7**

Type your comment here and press Enter Key (Minimum 10 characters)

Thanks@Nemi Chand for this article , i understood the concept. I have small question, one of the API action method is taking string is parameters, i want to implement notnullorEmpty constraint to that parameter, so that i no need to write if else logic in my action method, i would like to achieve the same model that you implemented here, but i don't want apply routes in conventional way, i want to do it attribute routing manner, [alphanumeric] , does this possible? any examples will be appreciated

Sandeep Ch　　　　　　　　　　　　　　　　　　　　　　Sep 27, 2019
　1487　**350**　**2.2k**　　　　　　　　　　　　　　0　　　　0　　　Reply

Nice and to the point, +1

M. Moore　　　　　　　　　　　　　　　　　　　　　　Nov 02, 2018
　1834　**3**　**0**　　　　　　　　　　　　　　　　1　　　1　　　Reply

Thanks for reading. hope you liked it.

Login

Contribute   Ask Question

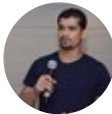Hello! please answer me how I can handle errors from constraint validation? I need to show errors

Sergei Calabonga

Sep 04, 2018

1825   **12**   **0**   1   1   Reply

Purpose of route constraint to match/validate the requested parameter so if you want to show any custom message, there is no direct way. you can use tempdata to store message in route constraint and show that to user.

Nemi Chand

Sep 05, 2018

468   **3.9k**   **562.3k**   1

Nice article. Thank you for sharing

Manoj Kulkarni

Jun 26, 2017

615   **2.5k**   **316.8k**   1   1   Reply

Thanks for reading. hope you liked it.

Nemi Chand

Jun 27, 2017

468   **3.9k**   **562.3k**   0

Basics

C# Corner   Login

Contribute   Ask Question

FEATURED ARTICLES

Setup Azure CI/CD Pipelines Using Visual Studio

Create A Simple Blazor Server Application With .NET Core 3.0

Linear Regression

Reverse Geo Coding Using Google Maps API In Power Query

gRPC Using C# And .Net Core - Day One

View All ◯

TRENDING UP

01   Learn Angular 8 Step By Step in 10 Days – View Encapsulation (Day 6)

02   Learn Angular 8 Step By Step In 10 Days - Angular Forms (Day 7)

03   Connect Your Virtual Machine Using Bastion

04   Overview Of Polymorphism In C#

05   Establishing VPN Gateway Connection Between VNet To VNet

06   Overview Of SharePoint Column Indexing

07   MS Team - Sync SharePoint Files With One Drive

08   Reusing A Deleted SharePoint Online Team Site Name

09   ASP.NET Core with AutoWrapper: Customizing the Default Response Output

10   Theft Detection Using Microsoft Cognitive Services

View All 〇