

We now integrate with Microsoft Teams, helping you to connect your internal knowledge base with your chat. [Learn more.](#)

Model binding in ASP.NET Core to map underscores to title case property names

Asked 3 years, 3 months ago Active 6 months ago Viewed 5k times



20



1

I have a model class that I want to bind a query string to in my ASP.NET MVC Core (RC2) application.

I need to support underscores in query string keys to confirm to OAuth specs, but I want to work with title case property names in my application.

My model class looks like this:

```
class OAuthParameters
{
    public string ClientId {get; set;}

    public string ResponseType {get; set;}

    public string RedirectUri {get; set;}
}
```

so I'd like to bind query strings like `client_id` , `response_type` and `redirect_uri` to it.

Is there a way for ASP.NET MVC Core to do this automagically or through an attribute annotation?

I've read some articles about writing custom model binders, but these seem to (1) be overly complex for what I'm trying to achieve and (2) are written for RC1 or earlier in mind and some of the syntax has changed.

Thanks in advance.

c#

asp.net

asp.net-mvc

asp.net-core

asp.net-core-mvc

edited Jul 11 '16 at 12:35



Kiran Challa

49.4k 10 158 145

asked Jul 11 '16 at 11:03



JamieNewman

455 1 9 16

1 Model binders don't have to be incredibly complicated, and they are definitely the right entry point for what you are trying to achieve – [ste-fu](#) Jul 11 '16 at 11:16

1 docs.asp.net/en/latest/mvc/models/model-binding.html – [Set](#) Jul 11 '16 at 11:30

But what's the problem, data view objects transformed (controller model) in data transfer objects (Business Logic DTO) are common in mvc model biding don't see any problem with this approach – [SilentTremor](#) Jul 11 '16 at 11:33

2 Answers



You can use the `FromQuery` attribute's `Name` property here.

33

Example:



```
public class OauthParameters
{
    [FromQuery(Name = "client_id")]
    public string ClientId { get; set; }

    [FromQuery(Name = "response_type")]
    public string ResponseType { get; set; }

    [FromQuery(Name = "redirect_uri")]
    public string RedirectUri { get; set; }
}
```

answered Jul 11 '16 at 12:34



[Kiran Challa](#)

49.4k 10 158 145

Amazing. I'll give it a try, thanks! – [JamieNewman](#) Jul 11 '16 at 14:30

Perfect. Exactly what I was looking for. – [JamieNewman](#) Jul 11 '16 at 15:24

1 Is there an equivalent for POST requests? – [dalemac](#) Jul 6 '17 at 10:35

6 @dalemac Use `[FromForm]` instead of `[FromQuery]` for POST request. – [Andrew Basarab](#) Oct 22 '17 at 12:27

3 For those looking for the json body solution: use e.g. `[JsonProperty("json_name")]` – [Geoff](#) Dec 8 '17 at 15:29



Solution for .net core 2.1 and 2.2

-3

Or without attributes you can do something like this which is cleaner I think (of course if the model properties are same as query parameters).



Meanwhile I use it in .net core 2.1 and 2.2

```
public async Task<IActionResult> Get([FromQuery]ReportQueryModel queryModel)
{
}
}
```

edited May 5 at 10:34

answered Jul 12 '18 at 12:19



[peyman gilmour](#)

552 2 8 20