WebApi 2 POST with single string parameter not working

26

7

I have the following controller:

```
public class ValuesController : ApiController
{
    // POST api/values
    public IHttpActionResult Post(string filterName)
    {
        return new JsonResult<string>(filterName, new JsonSerializerSettings(),
Encoding.UTF8, this);

    }
}
```

WebApi config

```
config.Routes.MapHttpRoute(
    name: "DefaultApi",
    routeTemplate: "api/{controller}/{id}",
    defaults: new { id = RouteParameter.Optional });
```

I use this js code to call the api

```
$.ajax(
{
    url: "/api/values/",
    type: "POST",
    dataType: 'json',
    data: { filterName: "Dirty Deeds" },
    success: function (result) {
        console.log(result);
    },
    error: function (xhr, status, p3, p4) {
        var err = "Error " + " " + status + " " + p3;
```

I get a 405 method not allowed (post)

javascript c# asp.net-web-api asp.net-web-api2

edited May 4 at 12:23
Cœur
21.5k 10 123 169

asked Jun 15 '16 at 17:41
Danny
317 1 4 14

To the web api method add `[HttpPost]` below your comment. Also I believe that the URL is case sensitive, in the javascript it should probably read `Values` with a capital `V` – Igor Jun 15 '16 at 17:50

Tried that but same error – Danny Jun 15 '16 at 17:53

1 Also if you are sending data via the http message (and not the url) you should add `[FromBody]` to the method signature. `([FromBody] string filterName)` – Igor Jun 15 '16 at 17:54

1 Your controller method is expecting a string, but you are passing it an object { filterName: "Dirty Deeds" }. Thats a javascript object with one property of filterName. Your controller isn't expecting an object with a property of filterName, just a string. – peinearydevelopment Jun 15 '16 at 17:57

When I use FromBody the param filterName is null – Danny Jun 15 '16 at 17:58

3 Answers

31

✓

c#

```
public class ValuesController : ApiController
{
    // POST api/values
    [HttpPost] // added attribute
    public IHttpActionResult Post([FromBody] string filterName) // added FromBody as
this is how you are sending the data
    {
        return new JsonResult<string>(filterName, new JsonSerializerSettings(),
Encoding.UTF8, this);
```

```
$.ajax(
{
    url: "/api/Values/", // be consistent and case the route the same as the
ApiController
    type: "POST",
    dataType: 'json',
    data: "=Dirty Deeds", // add an = sign
    success: function (result) {
        console.log(result);
    },
    error: function (xhr, status, p3, p4) {
        var err = "Error " + " " + status + " " + p3;
        if (xhr.responseText && xhr.responseText[0] == "{")
            err = JSON.parse(xhr.responseText).message;
        console.log(err);
    }
});
```

Explanation

Because you are only sending a single value add the = sign in front of it so it will be treated like forms encoding. You can also add the content type if you want to make it clear that this is what you are doing to the ajax call.

```
contentType: 'application/x-www-form-urlencoded'
```

Alternatively you could also send the content via URL OR wrap the content in an object on the server as well as in the ajax call and stringify it.

```
public class Filter {
    public string FilterName {get;set;}
}

public class ValuesController : ApiController
{
    // POST api/values
    [HttpPost] // added attribute
    public IHttpActionResult Post([FromBody] Filter filter) // added FromBody as this is
how you are sending the data
```

JavaScript

```
$.ajax(
{
    url: "/api/Values/", // be consistent and case the route the same as the
ApiController
    type: "POST",
    dataType: 'json',
    contentType: 'application/json',
    data: JSON.stringify({FilterName: "Dirty Deeds"}), // send as json
    success: function (result) {
        console.log(result);
    },
    error: function (xhr, status, p3, p4) {
        var err = "Error " + " " + status + " " + p3;
        if (xhr.responseText && xhr.responseText[0] == "{")
            err = JSON.parse(xhr.responseText).message;
        console.log(err);
    }
});
```

edited Jun 15 '16 at 18:57

answered Jun 15 '16 at 17:58
Igor
44.5k 4 56 114

This gives no http error but the param filterName is null – Danny Jun 15 '16 at 18:07

@DannyH - I updated the javascript and tested it, this works. – Igor Jun 15 '16 at 18:20

Hi, It works for me too. three questions: Why Do I need the [FormBody] Is this also needed when passing an object? Why the ContentType: 'application/x-www-form-urlencoded' why "=Dirty Deeds" – Danny Jun 15 '16 at 18:24

And how do I approach this when passing 1 string and 1 object? Same way? – Danny Jun 15 '16 at 18:26

@DannyH - Web API will only ever allow a single object to be passed in the message body. So for that you should create an aggregate type that you expect that contains all the data you expect and pass that in using my 2nd example I posted. If you have simple types (int, date, etc) you can pass those in as FromUri and pass them in the query string. – Igor Jun 15 '16 at 18:27

answered Nov 27 '16 at 10:35
misha
1,990 2 22 28

1 This worked. Thanks. How do I send an integer? – Sachin Pakale Feb 15 '17 at 9:04

2 @SachinPakale you can do the same way just by changing the parameter type. (`int filterNumber`). The `Controller` should automatically cast the passed value to a `String` – Curiousity Oct 13 '17 at 3:12

0

Add [HttpPost] attribute to method in controller

edited Jun 15 '16 at 18:17

answered Jun 15 '16 at 17:49
meJustAndrew
3,336 4 27 53

1 You mean HttpPost attribute? Tried that and that isn't working – Danny Jun 15 '16 at 18:01

Yea, httppost attribute. Right now, as you have it, it should work if you invoke it as a GET method from js – meJustAndrew Jun 15 '16 at 18:12

HttpPost attribute gives me the 405 method not allowed – Danny Jun 15 '16 at 18:15

And if you use HttpPost attribute, and you don't use FromBody attribute, you should send data as a query string as mentioned in the answer above – meJustAndrew Jun 15 '16 at 18:16