Post array of strings to web API method

Asked 6 years, 8 months ago Active 2 years, 7 months ago Viewed 32k times

9

2

this is my client side ajax call:

```
var list = ["a", "b", "c", "d"];

var jsonText = { data: list };

$.ajax({
    type: "POST",
    url: "/api/scheduledItemPriceStatus/updateStatusToDelete",
    data: jsonText,
    dataType: "json",
    traditional: true,
    success: function() { alert("it worked!"); },
    failure: function() { alert("not working..."); }
});
```

this is chrome network header:

```
Request URL:http://localhost:2538/api/scheduledItemPriceStatus/updateStatusToDelete

Request Method:POST

Request Headersview source

Accept:application/json, text/javascript, */*; q=0.01

Accept-Charset:ISO-8859-1,utf-8;q=0.7,*;q=0.3

Accept-Encoding:gzip,deflate,sdch

Accept-Language:en-US,en;q=0.8

Connection:keep-alive

Content-Length:27
```

```
Host:localhost:2538

Origin:http://localhost:2538

Referer:http://localhost:2538/Pricing/ScheduledItemPrices

User-Agent:Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.11 (KHTML, like Gecko)
Chrome/23.0.1271.97 Safari/537.11

X-Requested-With:XMLHttpRequest
```

Form Dataview URL encoded

```
data:a
data:b
data:c
data:d
```

this is my webapi controller method:

```
public HttpResponseMessage UpdateStatusToDelete(string[] data)
```

result:

when I debug, the data parameter in UpdateStatusToDelete returns `{string[0]}` instead of data:a data:b data:c data:d

What am I doing wrong? Any help is really appreciated.

c# javascript ajax post asp.net-web-api

edited Jan 8 '13 at 5:53
COLD TOLD
11.9k 3 28 47

asked Jan 8 '13 at 5:43
Eunyoung Ro
56 1 1 3

Hmmm, I think you should directly pass the `list` variable as the data field in your ajax call. Right now, it looks like you are passing an object which has an array of strings as a member field called "data". – Ameen Jan 8 '13 at 5:51

does that first line compile..? var list = ["a", "b", "c", "d"]? it should be like this `string[] list = {"a", "b", "c", "d"};` – MethodMan Jan 8 '13 at

6 Answers

31

For passing simply types, the data to post must take the form of a name value pair with the name portion being an empty string. So you need to make the Ajax call like so:

```
$.ajax({
  type: "POST",
  url: "/api/values",
  data: { "": list },
  dataType: "json",
  success: function() { alert("it worked!"); },
  failure: function() { alert("not working..."); }
});
```

Additionally, on your Web API action, annotate it w/ the [FromBody] attribute. Something like:

```
public void Post([FromBody]string[] values)
```

That should do the trick.

answered Jan 11 '13 at 22:55

Mark Berryman
879 7 9

this should be the marked correct answer. – Alberto Montellano Nov 23 '15 at 20:56

I still get a null values for values using this approach. the one different is My API also has Non body parameters. Something like public void Post(string id, [FromBody]string[] values) – Shaswat Rungta Oct 5 '16 at 3:54

3

```
var list = ["a", "b", "c", "d"];
```

in

```
$.ajax({
        type: "POST",
        url: "/api/scheduledItemPriceStatus/updateStatusToDelete",
        // Pass the list itself
        data: list,
        dataType: "json",
        traditional: true,
        success: function() { alert("it worked!"); },
        failure: function() { alert("not working..."); }
    });
```

Your method signature on server is correct.

answered Dec 8 '14 at 7:21

Shaulian

351 4 8

---

use `var jsonText = { data: JSON.stringify(list)}`

2

answered Jan 8 '13 at 5:52

Cris

8,279 5 28 44

1 Chris there is now from what I can tell that the list portion will compile.. check the code out and tell me.... [] should be { } – MethodMan Jan 8 '13 at 5:54

yes [] should be {} – Cris Jan 8 '13 at 5:57

In the backend, you could use `FormDataCollection.GetValues(string key)` to return an array of strings for that parameter.

```
    ...
 }
```

answered Feb 17 '17 at 8:28

Stelian Matei

9,630 2 20 28

0

use the method above to send array as suggested by cris in your jquery ajax call. JSON data is usually in key value pair.

```
   var tmp = [];
   tmp.push({
   //this is the key name 'a'  "a": "your value", //it could be anything here in
    //string format.
     "b": "b",
     "c": "c",
     "d": "d"
    });

  { data: JSON.stringify(tmp);}
```

You can also accomplish above by using two dimensional array

Additionally do this in the webapi project.

under the models folder in your web api project create a class file. Possibly **class1.cs**.

Create 4 properties

```
 public string a {get; set;}
 public string b {get; set;}
 public string c {get; set;}
 public string d {get; set;}
```

Now do this in your controller

```
 using projectname.models
```

```
  //Further logic goes here
}
```

I am sure this will work.

edited Dec 22 '14 at 1:05

answered Dec 21 '14 at 16:17

Gaurav Shah
306 3 7

0

Setting the dataType won't help you.

This is how I do it :

```
var SizeSequence = {};
SizeSequence.Id = parseInt(document.querySelector("dd#Sequence_Id").textContent);
SizeSequence.IncludedSizes = [];
var sizes = document.querySelectorAll("table#IncludedElements td#Size_Name");
// skipping the first row (template)
for (var i = 1, l = sizes.length ; i != sizes.length ;
SizeSequence.IncludedSizes.push(sizes[i++].textContent));

$.ajax("/api/SizeSequence/" + SizeSequence.Id, {
     method: "POST",
     contentType: "application/json; charset=UTF-8",
     data: JSON.stringify(SizeSequence.IncludedSizes),
...
```

The Server Part

```
// /api/SizeSequence/5
public async Task<IHttpActionResult> PostSaveSizeSequence(int? Id, List<String>
IncludedSizes)
    {
        if (Id == null || IncludedSizes == null || IncludedSizes.Exists( s =>
String.IsNullOrWhiteSpace(s)))
            return BadRequest();
        try
        {
```

```
    catch ( Exception exc)
    {
        return Conflict();
    }
}
```

References

jQuery.ajax()

answered May 29 '15 at 18:07

Sergey

1