

# ASP.NET Web API : Correct way to return a 401/unauthorised response

Asked 4 years, 3 months ago   Active 11 days ago   Viewed 87k times



I have an MVC webapi site that uses OAuth/token authentication to authenticate requests. All the relevant controllers have the right attributes, and authentication is working ok.

82



The problem is that not all of the request can be authorised in the scope of an attribute - some authorisation checks have to be performed in code that is called by controller methods - what is the correct way to return a 401 unauthorised response in this case?



10

I have tried `throw new HttpException(401, "Unauthorized access");` , but when I do this the response status code is 500 and I get also get a stack trace. Even in our logging DelegatingHandler we can see that the response is 500, not 401.

c#

asp.net-mvc

authorization

edited Jul 3 '15 at 12:02



LukeH

217k

47

320

388

asked Jul 3 '15 at 11:20



GoatInTheMachine

1,857

1

19

28

- 1 To anyone picking up this answer down the line, I'd suggest thinking about the appropriate time to throw a `HttpResponseException` versus when to return an `Unauthorized()` . Using the exception for an 'expected' error is a bit of an anti-pattern, so if there are cases you expect the call to make this mistake, returning `Unauthorized()` is probably the right call. Save `HttpResponseException` for the truly unexpected. – Rikki Sep 17 '18 at 18:06

See [github.com/aspnet/Mvc/issues/5507](https://github.com/aspnet/Mvc/issues/5507) for some discussion. – Rikki Sep 17 '18 at 18:07

@Rikki, 401 is not an "expected" error. -- It's an exceptional circumstance that should cause you to abort your workflow (except maybe for logging, which you should already be doing for any exception...) -- Anyway, if you want to return a strong typed result from your controller (e.g. for ease of unit testing), an Exception is clearly the best route. – BrainSlugs83 Oct 9 '18 at 20:58

## 6 Answers



You should be throwing a [HttpResponseException](#) from your API method, not [HttpException](#) :

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

Or, if you want to supply a custom message:



```
var msg = new HttpResponseMessage(HttpStatusCode.Unauthorized) { ReasonPhrase =  
    "Oops!!!" };  
throw new HttpResponseException(msg);
```

answered Jul 3 '15 at 12:26



[LukeH](#)

217k 47 320 388

Just got it working doing exactly this, thanks! – [GoatInTheMachine](#) Jul 3 '15 at 12:50

Just return the following:

76

```
return Unauthorized();
```

answered Aug 1 '16 at 20:39



[JohnWrensby](#)

1,344 11 14

- 1 I think the accepted answers the OP's question specifically. My answer answers the question's title "ASP.NET Web API : Correct way to return a 401/unauthorised response" – [JohnWrensby](#) Dec 18 '16 at 21:34
- 3 Anybody know why there's no overloaded version of this with a message? – [Simon\\_Weaver](#) Jul 9 '18 at 8:07
- 5 @Simon\_Weaver No idea why, but you could use a `return Content<string>(HttpStatusCode.Unauthorized, "Message");` to do this. – [Rikki](#) Sep 17 '18 at 18:03
- 2 This should be the correct answer. 1 it is correct. 2) If this changes in a later framework, you don't have to change code. 3) You don't need to provide a reason to a 401. This should be handled by the client and not the server. – [Nick Turner](#) Jan 2 at 20:15

Which library is this in? – [Nae](#) May 13 at 11:46

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

## 14 ASP.NET

```
return Content(HttpStatusCode.Unauthorized, "My error message");
```

## Update: ASP.NET Core

Above code does not work in ASP.NET Core, you can use one of these instead:

```
return StatusCode((int)System.Net.HttpStatusCode.Unauthorized, "My error message");
return StatusCode(401, "My error message");
```

Apparently the reason phrase is pretty optional ([Can an HTTP response omit the Reason-Phrase?](#))

edited Jan 28 at 15:53

answered Feb 14 '18 at 11:15



Alex AIT

3,110 3 13 26

1 This no-longer works in ASP.NET Core, the `ControllerBase` class (used by ASP.NET Core WebAPI) no-longer has a `Content` overload that accepts a HTTP status code. – Dai Sep 7 '18 at 23:54

This is wrong. A `Content` response is a 200 Ok status. The server should send a 401 and the client should handle accordingly. You can't send a 200 as a 401. It doesn't make sense. If the client gets a 401, it's not an Oops, it's a your breaking the law. – Nick Turner Jan 2 at 20:19

This code is sending a 401 status code( `HttpStatusCode.Unauthorized` ), not 200. `Content(...)` simply a shorthand for returning any given content with a given HTTP status code. If you want to send 200 you can use `Ok(...)` – Alex AIT Jan 2 at 21:39

@NickTurner -- that's an argument for the `webapi2 Content()` method being poorly named not for this being the wrong answer. Since the `(status,message)` method is renamed in `NetCore`, I guess the devs agree it was poorly named. – Chris F Carroll Apr 9 at 13:36

You get a 500 response code because you're throwing an exception (the `HttpException` ) which indicates some kind of server error, this is the wrong approach.

7

Just set the response status code .e.g

```
Response.StatusCode = (int)HttpStatusCode.Unauthorized;
```

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).



It's a bit odd then that the exception takes the HTTP status code as a parameter, and intellisense docs say that this is the status code sent to the client - I was hoping to avoid mutating the response myself directly as this seems error prone, seeing as its global state – [GoatInTheMachine](#) Jul 3 '15 at 12:06

1 The base Web API controller doesn't expose a `Response` property. – [LukeH](#) Jul 3 '15 at 12:07

you can use follow code in asp.net core 2.0:

0

```
public IActionResult index()
{
    return new ContentResult() { Content = "My error message", StatusCode =
(int)HttpStatusCode.Unauthorized };
}
```

answered Jun 12 at 10:39



To add to an existing answer in ASP.NET Core >= 1.0 you can

0

```
return Unauthorized();

return Unauthorized(object value);
```

To pass info to the client you can do a call like this:

```
return Unauthorized(new { Ok = false, Code = Constants.INVALID_CREDENTIALS, ...});
```

On the client besides the 401 response you will have the passed data too. For example on most clients you can `await response.json()` to get it

By using our site, you acknowledge that you have read and understand our [Cookie Policy](#), [Privacy Policy](#), and our [Terms of Service](#).

