403 Forbidden vs 401 Unauthorized HTTP responses

Asked 14 years, 5 months ago Modified 4 months ago

Viewed 1.6m times

3642

For a web page that exists, but for which a user does not have sufficient privileges (they are not logged in or do not belong to the proper user group), what is the proper HTTP response to serve?

`401 Unauthorized`?
`403 Forbidden`?
Something else?

What I've read on each so far isn't very clear on the difference between the two. What use cases are appropriate for each response?

http-headers http-status-code-403 http-status-codes http-status-code-401 http-response-codes

Share

Improve this question

Follow

edited Nov 26, 2019 at 15:37

vzwick

11k ● 5 ● 46 ● 63

asked Jul 21, 2010 at 7:21

VirtuosiMedia

53.3k ● 21 ● 98 ● 140

718 401 'Unauthorized' should be 401 'Unauthenticated', problem solved ! – Christophe Roussy May 17, 2016 at 12:33

131 I don't remember how many times me and my colleagues have come back to stackoverflow for this question. Maybe HTTP standards should consider modifying the names or descriptions for 401 and 403. – neurite Feb 4, 2017 at 1:14

3 @Qwerty no, the new RFC7231 obsoletes RFC2616. 403 has a different meaning now. – fishbone Aug 1, 2018 at 13:15

2 @fishbone you also did not note that status code 401 has been removed from that RFC :D – Barkermn01 Nov 22, 2018 at 12:06

3 @fishbone it's been added back to that proposal now but uses a different RFC now 7235 tools.ietf.org/html/rfc7235#section-3.1 – Barkermn01 Jan 16, 2020 at 15:24

22 Answers

Sorted by: Highest score (default)

▲

5163

▼

A clear explanation from Daniel Irvine [original link]:

> There's a problem with *401 Unauthorized*, the HTTP status code for authentication errors. And that’s just it: it’s for authentication, not authorization. Receiving a 401 response is the server telling you, “you aren’t authenticated–

> either not authenticated at all or authenticated incorrectly–but please reauthenticate and try again.” To help you out, it will always include a *WWW-Authenticate* header that describes how to authenticate.
>
> This is a response generally returned by your web server, not your web application.
>
> It’s also something very temporary; the server is asking you to try again.
>
> So, for authorization I use the *403 Forbidden* response. It’s permanent, it’s tied to my application logic, and it’s a more concrete response than a 401.
>
> Receiving a 403 response is the server telling you, “I’m sorry. I know who you are–I believe who you say you are–but you just don’t have permission to access this resource. Maybe if you ask the system administrator nicely, you’ll get permission. But please don’t bother me again until your predicament changes.”
>
> In summary, a *401 Unauthorized* response should be used for missing or bad authentication, and a *403 Forbidden* response should be used afterwards, when the user is authenticated but isn’t authorized to perform the requested operation on the given resource.

Another nice pictorial format of how http status codes should be used.

Share Improve this answer

Follow

edited Jul 21, 2022 at 17:42

Nick T

26.7k ● 14 ● 87 ● 127

answered Aug 4, 2011 at 6:24

JPReddy

65.4k ● 17 ● 66 ● 93

62 The default IIS 403 message is "This is a generic 403 error and means the authenticated user is not authorized to view the page", which would seem to agree. – Ben Challenor Sep 16, 2011 at 13:19

466 @JPReddy Your answer is correct. However, I would expect that 401 to be named "Unauthenticated" and 403 to be named "Unauthorized". It is very confusing that 401, which has to do with Authentication, has the format accompanying text "Unauthorized"....Unless I am not good in English (which is quite a possibility). – p.matsinopoulos Jun 20, 2012 at 21:48

75 @ZaidMasud, according to RFC this interpretation is not correct. Cumbayah's answer got it right. 401 means "you're missing the right authorization". It implies "if you want you might try to authenticate yourself". So both a client who didn't authenticate itself correctly and a properly authenticated client missing the authorization will get a 401. 403 means "I won't answer to this, whoever you are". RFC states clearly thath "authorization will not help" in the case of 403. – Davide R. Nov 24, 2012 at 10:38

112 401 is Authentication error, 403 is Authorization error. Simple as that. – Gelmir Mar 25, 2013 at 14:09

55 To all downvoters referring to an RFC (most likely 2616), you are all wrong. As specified in the answer by @Idrut, "Forbidden means that the client has authenticated successfully, but is not authorized.". He references RFC7231 and RFC7235 which **obsolete** RFC 2616. – Tom Lint Sep 22, 2015 at 8:43

530

Edit: RFC2616 is obsolete, see RFC9110.

401 Unauthorized:

> If the request already included Authorization credentials, then the 401 response indicates that authorization has been refused for those credentials.

403 Forbidden:

> The server understood the request, but is refusing to fulfill it.

From your use case, it appears that the user is not authenticated. I would return 401.

Share Improve this answer Follow

edited Aug 11, 2022 at 23:07

emery
9,613 ● 11 ● 49 ● 53

answered Jul 21, 2010 at 7:28

Oded
498k ● 102 ● 893 ● 1k

30 Thanks, that helped clarify it for me. I'm using both - the 401 for unauthenticated users, the 403 for authenticated users with insufficient permissions. – VirtuosiMedia Jul 21, 2010 at 7:51

56 I didn't downvote but I find this answer quite misleading. 403 forbidden is more appropriately used in content that will never be served (like .config files in asp.net). its either that or a 404. imho, it wouldn't be appropriate to return 403 for something that can be accessed but you just didn't have the right credentials. my solution would be to give an access denied message with a way to change credentials. that or a 401. – Mel Dec 22, 2011 at 5:07

34 "The response MUST include a WWW-Authenticate header field (section 14.47) containing a challenge applicable to the requested resource." It would seem that if you don't want to use HTTP-style authentication, a 401 response code is not appropriate. – Brilliand Mar 20, 2012 at 1:42

9 I'll back Billiand here. The statement is "If the request already included Authorization credentials". That means if this is a response from a request which provided the credential (e.g. the response from a RFC2617 Authentication attempt). It is essentially to allow the server to say, "Bad account/password pair, try again". In the posed question, the user is presumably authenticated but not authorized. 401 is never the appropriate response for those circumstances. – ldrut Feb 5, 2013 at 17:20

8 Brilliand is right, 401 is only appropriate for HTTP Authentication. – Juan Pablo Rinaldi May 3, 2013 at 15:42

341

Something the other answers are missing is that it must be understood that Authentication and Authorization in the context of RFC 2616 refers ONLY to the HTTP Authentication protocol of RFC 2617. Authentication by schemes outside of RFC2617 is not supported in HTTP status codes and are not considered when deciding whether to use 401 or 403.

Brief and Terse

Unauthorized indicates that the client is not RFC2617 authenticated and the server is initiating the authentication process. Forbidden indicates either that the client is RFC2617 authenticated and does not have authorization or that the server does not support RFC2617 for the requested resource.

Meaning if you have your own roll-your-own login process and never use HTTP Authentication, 403 is always the proper response and 401 should never be used.

Detailed and In-Depth

From RFC2616

> 10.4.2 401 Unauthorized

> The request requires user authentication. The response MUST include a WWW-Authenticate header field (section 14.47) containing a challenge applicable to the requested resource. The client MAY repeat the request with a suitable Authorization header field (section 14.8).

and

> 10.4.4 403 Forbidden The server understood the request but is refusing to fulfil it. Authorization will not help and the request SHOULD NOT be repeated.

The first thing to keep in mind is that "Authentication" and "Authorization" in the context of this document refer specifically to the HTTP Authentication protocols from RFC 2617. They do not refer to any roll-your-own authentication protocols you may have created using login pages, etc. I will use "login" to refer to authentication and authorization by methods other than RFC2617

So the real difference is not what the problem is or even if there is a solution. The difference is what the server expects the client to do next.

401 indicates that the resource can not be provided, but the server is REQUESTING that the client log in through HTTP Authentication and has sent reply headers to initiate the process. Possibly there are authorizations that

will permit access to the resource, possibly there are not, but let's give it a try and see what happens.

403 indicates that the resource can not be provided and there is, for the current user, no way to solve this through RFC2617 and no point in trying. This may be because it is known that no level of authentication is sufficient (for instance because of an IP blacklist), but it may be because the user is already authenticated and does not have authority. The RFC2617 model is one-user, one-credentials so the case where the user may have a second set of credentials that could be authorized may be ignored. It neither suggests nor implies that some sort of login page or other non-RFC2617 authentication protocol may or may not help - that is outside the RFC2616 standards and definition.

Edit: RFC2616 is obsolete, see RFC7231 and RFC7235.

Share Improve this answer Follow

edited Oct 7, 2021 at 7:34

Community Bot
1 ● 1

answered Feb 5, 2013 at 17:14

ldrut
3,917 ● 1 ● 18 ● 4

---

7 So what should we do when the user requests a page that requires non-http authentication? Send status code 403?
– marcovtwout Mar 25, 2014 at 11:00

13 This is important: "if you have your own roll-your-own login process and never use HTTP Authentication, 403 is always the proper response and 401 should never be used." – ggg Dec 31, 2014 at 6:25

3 RFC2616 should be burned and replaced by RFC7235, but contains no changes in this topic as far as I can see. – Alex Feb 3, 2015 at 5:57

7 Doesn't RFC7235 provide for "roll-your-own" or alternate auth challenges? Why can't my app's login flow present its challenge in the form of a `WWW-Authenticate` header? Even if a browser doesn't support it, my React app can... – jchook Oct 11, 2016 at 15:53

2 RFC 7231 (Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content) changes the meaning of 403: There is no more "Authorization will not help". It says: "If authentication credentials were provided in the request, the server considers them insufficient to grant access. The client SHOULD NOT automatically repeat the request with the same credentials. The client MAY repeat the request with new or different credentials." So, Authorization (grant more permissions) **will** help and this answer is incorrect now (formerly was correct). – Ruslan Stelmachenko Nov 9, 2018 at 4:12

217

```
  +-----------------------
  | RESOURCE EXISTS ? (if private it is often
checked AFTER auth check)
  +-----------------------
    |       |
 NO |       v YES
    v      +-----------------------
   404     | IS LOGGED-IN ? (authenticated, aka
user session)
   or      +-----------------------
   401        |              |
   403     NO |              | YES
   3xx        v              v
              401            +--------------------
---
       (404 no reveal)       | CAN ACCESS RESOURCE
? (permission, authorized, ...)
              or             +--------------------
---
             redirect          |            |
             to login       NO |            | YES
                               |            |
                               v            v
                               403          OK
200, redirect, ...
                      (or 404: no reveal)
                      (or 404: resource does not
exist if private)
                      (or 3xx: redirection)
```

Checks are usually done in this order:

- 404 if resource is public and does not exist or 3xx redirection
- OTHERWISE:
- 401 if not logged-in or session expired
- 403 if user does not have permission to access resource (file, json, ...)
- 404 if resource does not exist or not willing to reveal anything, or 3xx redirection

UNAUTHORIZED: Status code (401) indicating that the request requires **authentication**, usually this means user needs to be logged-in (session). User/agent unknown by the server. Can repeat with other credentials. NOTE: This is confusing as this should have been named 'unauthenticated' instead of 'unauthorized'. This can also happen after login if session expired. Special case: **Can be used instead of 404** to avoid revealing presence or non-presence of resource (credits @gingerCodeNinja)

FORBIDDEN: Status code (403) indicating the server understood the request but refused to fulfill it. User/agent known by the server but has **insufficient credentials**. Repeating request will not work, unless credentials changed, which is very unlikely in a short time span. Special case: **Can be used instead of 404** to avoid revealing presence or non-presence of resource (credits @gingerCodeNinja) in the case that revealing the presence of the resource exposes sensitive data or gives an attacker useful information.

NOT FOUND: Status code (404) indicating that the requested resource is not available. User/agent known but server will not reveal anything about the resource, does as if it does not exist. Repeating will not work. This is a special use of 404 (github does it for example).

As mentioned by @ChrisH there are a few options for **redirection** 3xx (301, 302, 303, 307 or not redirecting at all and using a 401):

- Difference between HTTP redirect codes
- How long do browsers cache HTTP 301s?
- What is correct HTTP status code when redirecting to a login page?

- What's the difference between a 302 and a 307 redirect?

Share Improve this answer Follow

edited Aug 5, 2021 at 10:16

answered Feb 23, 2015 at 11:00

Christophe Roussy

16.9k ● 5 ● 92 ● 85

2 if the user is not logged in or logged in but does not have permission, and the content doesn't exist at location, sometimes you probably want to return 401/403 instead of 404, so that you don't expose what is or isn't there if the user is not authenticated and logged in. Just knowing something exists can hint toward something or break NDA. So sometimes the 404 part of this diagram should be moved under logged in/authenticated. – gingerCodeNinja Feb 3, 2019 at 12:05 ✎

@gingerCodeNinja yes this is the same logic as the one for 404 instead of 403, good to mention this case. – Christophe Roussy Feb 9, 2019 at 12:34

Thank you for including the *very valid* `no reveal` cases at all levels. This is heavily context dependent of course, but I like that you've made it clear that it's *possibly* an option in all of those cases. – Matt Kocaj Jul 8, 2019 at 6:46

1 @MattKocaj note that the `no reveal` case can sometimes be detected via subtle timing differences and should not be seen as a security feature, it may just slow down attackers or help a little with privacy. – Christophe Roussy Sep 16, 2019 at 8:21

1 I think you meant 302 and not "301 redirect to login". – Chris H. Oct 18, 2019 at 18:55

122

According to RFC 2616 (HTTP/1.1) 403 is sent when:

> The server understood the request, but is refusing to fulfill it. Authorization will not help and the request SHOULD NOT be repeated. If the request method was not HEAD and the server wishes to make public why the request has not been fulfilled, it SHOULD describe the reason for the refusal in the entity. If the server does not wish to make this information available to the client, the status code 404 (Not Found) can be used instead

In other words, if the client CAN get access to the resource by authenticating, 401 should be sent.

Share Improve this answer Follow

answered Jul 21, 2010 at 7:26

user116587

6 And if it's not clear if they can access or not? Say that I have 3 user levels - Public, Members, and Premium Members. Assume that the page is for Premium Members only. A public user is basically unauthenticated and *could* be in either Members or Premium Members when they log in. For the Member user level, a 403 would seem appropriate. For Premium Members, the 401. However, what do you serve the Public? – VirtuosiMedia Jul 21, 2010 at 7:40

28 imho, this is the most accurate answer. it depends on the application but generally, if an authenticated user doesn't have sufficient rights on a resource, you might want to provide a way to change credentials or send a 401. I think 403 is best suited for content that is never served. In asp.net this would mean web.config files *.resx files etc. because no matter which user logs in, these files will NEVER be served so there is no point in trying again. – Mel Dec 22, 2011 at 5:01

7 +1, but an uncertain +1. The logical conclusion is that a 403 should never be returned as either 401 or 404 would be a strictly better response. – CurtainDog Jun 21, 2013 at 7:09

16 @Mel I think a file that should not be accessed by the client should be a 404. It's a file that is internal to the system; the outside should not even know it exists. By returning a 403 you are letting the client know it exists, no need to give that information away to hackers. The spec for 403 says `An origin server that wishes to "hide" the current existence of a forbidden target resource MAY instead respond with a status code of 404 (Not Found).` – Ruan Mendes Sep 2, 2014 at 20:23

4 While this seems to me like it's probably an accurate interpretation of the old RFC 2616, note that RFC 7231 [defines the semantics of a 403 differently](), and in fact explicitly states that *"The client MAY repeat the request with new or different credentials."* So while this answer was

accurate in 2010, it's completely wrong today, because the meaning of the status code has been rewritten beneath our feet. (Annoyingly, the Changes from RFC 2616 appendix doesn't acknowledge the change!) – Mark Amery Apr 30, 2017 at 17:00

58

Assuming HTTP authentication (*WWW-Authenticate* and *Authorization* headers) **is in use**, if authenticating as another user would grant access to the requested resource, then 401 Unauthorized should be returned.

403 Forbidden is used when access to the resource is forbidden to everyone or restricted to a given network or allowed only over SSL, whatever as long as it is no related to HTTP authentication.

If HTTP authentication is not in use and the service has a cookie-based authentication scheme as is the norm nowadays, then a 403 or a 404 should be returned.

Regarding 401, this is from RFC 7235 (Hypertext Transfer Protocol (HTTP/1.1): Authentication):

> 3.1. 401 Unauthorized
>
> The 401 (Unauthorized) status code indicates that the request has not been applied because it lacks valid authentication credentials for the target resource. **The origin server MUST send a WWW-Authenticate header field** (Section 4.4) containing at least one challenge applicable

> to the target resource. **If the request included authentication credentials, then the 401 response indicates that authorization has been refused for those credentials**. The client MAY repeat the request with a new or replaced Authorization header field (Section 4.1). If the 401 response contains the same challenge as the prior response, and the user agent has already attempted authentication at least once, then the user agent SHOULD present the enclosed representation to the user, since it usually contains relevant diagnostic information.

The semantics of 403 (and 404) have changed over time. This is from 1999 (RFC 2616):

> 10.4.4 403 Forbidden
>
> The server understood the request, but is refusing to fulfill it. **Authorization will not help** and the request SHOULD NOT be repeated. If the request method was not HEAD and the server wishes to make public why the request has not been fulfilled, it SHOULD describe the reason for the refusal in the entity. If the server does not wish to make this information available to the client, the status code 404 (Not Found) can be used instead.

In 2014 RFC 7231 (Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content) changed the meaning of 403:

> 6.5.3. 403 Forbidden
>
> The 403 (Forbidden) status code indicates that the server understood the request but refuses to authorize it. A server that wishes to make public why the request has been forbidden can describe that reason in the response payload (if any).
>
> **If authentication credentials were provided in the request, the server considers them insufficient to grant access. The client SHOULD NOT automatically repeat the request with the same credentials. The client MAY repeat the request with new or different credentials. However, a request might be forbidden for reasons unrelated to the credentials.**
>
> An origin server that wishes to "hide" the current existence of a forbidden target resource MAY instead respond with a status code of 404 (Not Found).

Thus, a 403 (or a 404) might now mean about anything. Providing new credentials might help... or it might not.

I believe the reason why this has changed is RFC 2616 assumed HTTP authentication would be used when in practice today's Web apps build custom authentication schemes using for example forms and cookies.

Share Improve this answer

Follow

edited Oct 7, 2021 at 7:34

Community Bot
1 ● 1

answered Feb 27, 2013 at 9:44

Erwan Legrand
4,395 ● 28 ● 27

---

2 This is interesting. Based on RFC 7231 and RFC 7235, I don't see an obvious distinction between 401 and 403 – Brian Feb 27, 2015 at 15:20

---

2 403 means "I know you but you can't see this resource." There's no reason for confusion. – Michael Blackburn Aug 22, 2016 at 16:10

"If the request included authentication credentials, then the 401 response indicates that authorization has been refused for those credentials. The client MAY repeat the request with a new or replaced Authorization header field (Section 4.1)." However, then "4.2. The 'Authorization' header field allows a user agent to authenticate itself with an origin server". Looks like in RFC7235 they use the term "authorization" like it was "authentication". In that case, it might seem that an authenticated but not authorized user should not get a 401, but rather 403 – arcuri82 Mar 23, 2018 at 8:49

2 @Brian The main distinction is that you return a 401 if your system uses HTTP auth as specced in RFC 7235 (and thus you must return a WWW-Authenticate header with such a response), and a 403 otherwise. – Mark Amery Mar 8, 2021 at 20:18

4 @MichaelBlackburn No, that's not the case. The server doesn't need to know you to return a 403. For one thing, neither the old RFC 2616 spec nor the newer RFC 7231 spec ever says that; for another, the phrase *"**If** authentication credentials were provided in the request"* in the new spec only makes sense if it's possible to return a 403 in some cases where there were *not* authentication credentials included in the request (i.e. cases where the server definitely does not "know you"). – Mark Amery Mar 8, 2021 at 20:21

▲

57

▼

- **401 Unauthorized**: I don't know who you are. *This an authentication error.*
- **403 Forbidden**: I know who you are, but you don't have permission to access this resource. *This is an authorization error.*

Share Improve this answer Follow

edited Mar 4, 2020 at 9:26

Premraj

74.4k ● 26 ● 243 ● 184

answered Aug 6, 2019 at 12:37

Akshay Misal

723 ● 6 ● 6

Not sure it specifically "always" mean the sender was unknown. Just whatever they requested was not authorised. – James May 29, 2020 at 1:01

While your explanation looks convincing, but I am not satisfied or trsuting it coz the error 401 says authorization in name itself and you are mixing with authentication. Well, can I tell a scenario, using credentials I obtain token means authenticated successfully, and use that to access "unathorized resource" for that token. Thats unauthorized 401. What you have to say for this? – Jasmine Aug 19, 2020 at 6:40

3 @Jasmine your concern is understandable, but the above explanation is correct. The conflict in terminology is caused by the http spec not conforming to the currently widely used definitions to the terms 'authentication' and 'authorization'. Likely caused by these definitions not being universally used the way they are now. We are stuck with the conflict and the confusion it causes. Evidence supporting this is that the default behavior of browsers is to prompt for credentials on a 401 response. – Jim Reineri Oct 17, 2020 at 10:15

1 This is an admirably pithy summary of the distinction described in the accepted answer. Like the accepted answer, though, it's just plain wrong. Nothing written in the HTTP spec supports this distinction and what's more for typical website login systems that *don't* use `WWW-Authenticate` and `Authorization` headers returning 401s isn't allowed by spec at all. – Mark Amery Mar 8, 2021 at 21:01

The only thing wrong with your answer is that it's hard to find it among the mess of other overly verbose and confusing answers (and unnecessary comments, like mine). – Sam Watkins Nov 16 at 18:31

33

This is an older question, but one option that was never really brought up was to return a 404. From a security perspective, the highest voted answer suffers from a potential information leakage vulnerability. Say, for instance, that the secure web page in question is a system admin page, or perhaps more commonly, is a record in a system that the user doesn't have access to. Ideally you wouldn't want a malicious user to even know that there's a page / record there, let alone that they don't have access. When I'm building something like this, I'll try to record unauthenticate / unauthorized requests in an internal log, but return a 404.

OWASP has some more information about how an attacker could use this type of information as part of an attack.

Share Improve this answer Follow

edited Feb 1, 2020 at 21:11

answered Dec 25, 2014 at 9:09

Patrick White
862 ● 8 ● 18

3 The use of a 404 has been mentioned in previous answers. You're on point re: information leakage and this should be an important consideration for anyone rolling their own authentication/authorization scheme. +1 for mentioning OWASP. – Dave Watts Mar 10, 2015 at 11:53

4 Ironically the OWASP link now goes to a 404 page. I found something similar (I think) on owasp.org/index.php/… – Anne Douwe Jan 31, 2020 at 12:38

Depends on the API and how access is given. But "leaking" is not a problem if it returns 401 for username/password it's the same as for a web form surely? – James May 29, 2020 at 1:24

1 @anned20 Ironically, the link you posted also returns a 404 page. – Amir Asyraf Jun 20, 2022 at 9:12

1 There are two separate (potential) issues here: allowing an unauthenticated user to discover api endpoints, and allowing an unauthenticated user to discover the existence (or non-existence) of resources. With a well-designed, secure API, public enumeration of your api endpoints is not a problem. If a request matches a pattern like `GET /users/:id`, then the response for an authenticated user must be the same regardless of whether a user with the given id exists or not, but returning 403 for all ids is just as secure as returning 404 for all ids. – mbbush Feb 9, 2023 at 19:09

23

This question was asked some time ago, but people's thinking moves on.

Section 6.5.3 in this draft (authored by Fielding and Reschke) gives status code 403 a slightly different meaning to the one documented in RFC 2616.

It reflects what happens in authentication & authorization schemes employed by a number of popular web-servers and frameworks.

I've emphasized the bit I think is most salient.

> **6.5.3. 403 Forbidden**
>
> The 403 (Forbidden) status code indicates that the server understood the request but refuses to authorize it. A server that wishes to make public why the request has been forbidden can

> describe that reason in the response payload (if any).
>
> If authentication credentials were provided in the request, the server considers them insufficient to grant access. ***The client SHOULD NOT repeat the request with the same credentials. The client MAY repeat the request with new or different credentials.*** However, a request might be forbidden for reasons unrelated to the credentials.
>
> An origin server that wishes to "hide" the current existence of a forbidden target resource MAY instead respond with a status code of 404 (Not Found).

Whatever convention you use, the important thing is to provide uniformity across your site / API.

Share Improve this answer

Follow

edited Oct 7, 2021 at 7:13

Community Bot

1 ● 1

answered May 22, 2014 at 10:54

Dave Watts

890 ● 7 ● 11

3 The draft was approved and is now RFC 7231.
– Vebjorn Ljosa Apr 20, 2017 at 12:36

17

These are the meanings:

401: User not (correctly) authenticated, the resource/page require authentication

403: User's role or permissions does not allow to access requested resource, for instance user is not an administrator and requested page is for administrators.

Note: Technically, 403 is a superset of 401, since is legal to give 403 for unauthenticated user too. Anyway is more meaningful to differentiate.

Share Improve this answer Follow

edited Aug 24, 2021 at 6:45

answered Nov 19, 2019 at 10:17

Luca C.

12.5k ● 2 ● 88 ● 82

This is a great TLDR answer to this question. – Kousha Dec 2, 2019 at 23:57

1 This is clear and straightforwardly written, but wrong. It's totally fine to return 403s when the user is not authenticated. Nothing in the spec says otherwise, and often you *can't* use a 401 in that situation because returning a 401 is only legal if you include a WWW-Authenticate header. – Mark Amery Mar 8, 2021 at 21:44

1 tx @MarkAmery , i slightly corrected the sentence to include maybe autentication – Luca C. Mar 9, 2021 at 7:53

16

!!! DEPR: The answer reflects what used to be common practice, up until 2014 !!!

TL;DR

- 401: A refusal that has to do with authentication
- 403: A refusal that has NOTHING to do with authentication

Practical Examples

If **apache** *requires authentication* (via `.htaccess`), and you hit `Cancel`, it will respond with a `401 Authorization Required`

If **nginx** finds a file, but has no *access rights* (user/group) to read/access it, it will respond with `403 Forbidden`

RFC (2616 Section 10)

401 Unauthorized (10.4.2)

Meaning 1: **Need to authenticate**

> The request requires user authentication. ...

Meaning 2: **Authentication insufficient**

> ... If the request already included Authorization credentials, then the 401 response indicates that authorization has been refused for those credentials. ...

403 Forbidden (10.4.4)

Meaning: **Unrelated to authentication**

> ... Authorization will not help ...

More details:

> The server understood the request, but is refusing to fulfill it.

> It SHOULD describe the reason for the refusal in the entity

> The status code 404 (Not Found) can be used instead

(If the server wants to keep this information from client)

Share Improve this answer edited Mar 9, 2021 at 13:38

Follow

answered Feb 25, 2015 at 9:03

Levite
17.6k ● 8 ● 53 ● 51

Your "Authorization will not help" quote is from a spec that's been obsolete since June 2014. tools.ietf.org/html/rfc7231 replaced it and says the opposite - that *"The client MAY repeat the request with new or different credentials."* As such, it's now definitely okay to use a 403 response in "Need to authenticate" and "Authentication insufficient" scenarios. – Mark Amery Mar 8, 2021 at 21:40

Thank you! If you want you can edit the answer. For now I put a deprecation warning at the top. – Levite Mar 9, 2021 at 13:43

16

`401` response code means one of the following:

1. An access token is missing.
2. An access token is either expired, revoked, malformed, or invalid.

A `403` response code on the other hand means that the access token is indeed valid, but that the user does not have appropriate privileges to perform the requested action.

Share Improve this answer

edited Aug 8, 2023 at 19:22

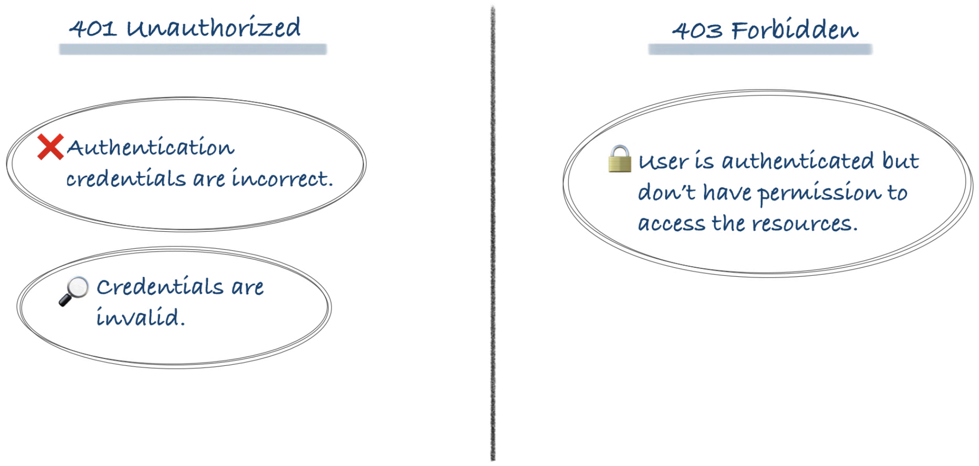
Follow

answered Feb 17, 2022 at 11:16

Ran Turner
17.8k ● 8 ● 56 ● 59

14

I have created a simple note for you which will make it clear.


401 Unauthorized
❌ Authentication credentials are incorrect.
🔍 Credentials are invalid.
403 Forbidden
🔒 User is authenticated but don't have permission to access the resources.

Share Improve this answer Follow

answered Nov 11, 2021 at 12:19

Pratham
537 ● 4 ● 7

13

> they are not logged in or do not belong to the proper user group

You have stated two different cases; each case should have a different response:

1. If they are not logged in at all you should return **401 Unauthorized**

2. If they are logged in but don't belong to the proper user group, you should return **403 Forbidden**

Note on the RFC based on comments received to this answer:

If the user is not logged in they are un-authenticated, the HTTP equivalent of which is 401 and is misleadingly called Unauthorized in the RFC. As section 10.4.2 states for **401 Unauthorized**:

> "The request requires user *authentication*."

If you're unauthenticated, 401 is the correct response. However if you're unauthorized, in the semantically correct sense, 403 is the correct response.

Share Improve this answer Follow

edited Nov 12, 2018 at 19:19

answered Oct 1, 2012 at 14:34

Zaid Masud

13.4k ● 9 ● 69 ● 88

5 This is not correct. Refer to RFC and to @Cumbayah's answer. – Davide R. Nov 24, 2012 at 10:40

8 @DavideR. the RFC uses *authentication* and *authorization* interchangeably. I believe it makes more sense when read with the *authentication* meaning. – Zaid Masud Nov 25, 2012 at 1:59

This answer is reversed. Unauthorized is not the same as Un-authenticated. @DavideR is right. Authentication and Authorization are NOT interchangeable – BozoJoe Oct 17, 2013 at 20:24

3 2616 should be burned. Several newer RFCs are much clearer that there is a need to differentiate between "I don't know you" and "I know you but you can't access this." There is *no* legitimate reason to acknowledge the existence of a resource that will never be fulfilled (or not fulfilled via http), which is what the 403-truthers are suggesting. – Michael Blackburn Aug 22, 2016 at 16:06

13

401: *Who are you again??* (programmer walks into a bar with no ID or invalid ID)

403: *Oh great, you again. I've got my eye on you. Go on, get outta here.* (programmer walks into a bar they are 86'd from)

Share Improve this answer Follow

answered Aug 11, 2022 at 23:10

emery

9,613 ● 11 ● 49 ● 53

9

In English:

401

> You are potentially allowed access but for some reason on this request you were denied. Such as

> a bad password? Try again, with the correct request you will get a success response instead.

403

> You are not, ever, allowed. Your name is not on the list, you won't ever get in, go away, don't send a re-try request, it will be refused, always. Go away.

Share Improve this answer Follow

answered Apr 8, 2020 at 14:23

James
4,742 ● 5 ● 38 ● 50

Just not true. The current spec's description of 403 states that *"The client MAY repeat the request with new or different credentials."*, which contradicts your description of 403 here. – Mark Amery Mar 8, 2021 at 21:46

1 @MarkAmery "repeat with new or different credentials" ok so my answer still stands because a new or different request is not a "re-try" is it? If you are logged in as your own user and get a 403, then try again you will get a 403. If you logout and back in with an Admin user and now get a 200 instead, that is not a retry request. It is a different request altogether with different credentials. So my answer still stands, "you" are not allowed, "your" name is not on the list, "you" wont ever get in, "don't send a re-try request". Using different credentials is not a "re-try" it's a new request. – James Mar 9, 2021 at 11:54

401: You need HTTP basic auth to see this.

8

If the user just needs to log in using you site's standard HTML login form, 401 would not be appropriate because it is specific to HTTP basic auth.

403: This resource exists but you are not authorized to see it, and HTTP basic auth won't help.

I don't recommend using 403 to deny access to things like `/includes`, because as far as the web is concerned, those resources don't exist at all and should therefore 404.

In other words, 403 means "this resource requires some form of auth other than HTTP basic auth (such as using the web site's standard HTML login form)".

https://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html#sec10.4.2

Share Improve this answer Follow

edited Jun 1, 2022 at 17:26

answered Sep 23, 2017 at 12:33

Val Kornea
4,697 ● 3 ● 41 ● 43

5

I think it is important to consider that, to a browser, 401 initiates an authentication dialog for the user to enter new credentials, while 403 does not. Browsers think that, if a 401 is returned, then the user should re-authenticate. So

401 stands for invalid authentication while 403 stands for a lack of permission.

Here are some cases under that logic where an error would be returned from authentication or authorization, with important phrases bolded.

- A resource requires authentication but **no credentials** were **specified**.

401: The client should specify credentials.

- The specified credentials are in an **invalid format**.

400: That's neither 401 nor 403, as syntax errors should always return 400.

- The specified credentials reference a **user** which **does not exist**.

401: The client should specify valid credentials.

- The specified **credentials** are **invalid** but specify a valid user (or don't specify a user if a specified user is not required).

401: Again, the client should specify valid credentials.

- The specified **credentials** have **expired**.

401: This is practically the same as having invalid credentials in general, so the client should specify valid credentials.

- The specified credentials are completely valid but do not **suffice** the particular **resource**, though it is possible that credentials with more permission could.

403: Specifying valid credentials would not grant access to the resource, as the current credentials are already valid but only do not have permission.

- The particular **resource** is **inaccessible** regardless of credentials.

403: This is regardless of credentials, so specifying valid credentials cannot help.

- The specified credentials are completely valid but the particular **client** is **blocked** from using them.

403: If the client is blocked, specifying new credentials will not do anything.

Share Improve this answer Follow

edited Dec 14, 2018 at 21:53

answered Jun 2, 2018 at 23:34

Grant Gryczan
1,595 ● 17 ● 26

1

Given the latest RFC's on the matter ([7231](7231) and [7235](7235)) the use-case seems quite clear (italics added):

- 401 is for unauthenticated ("lacks valid authentication"); i.e. 'I don't know who you are, or I don't trust you are who you say you are.'

> **401 Unauthorized**

> The 401 (Unauthorized) status code indicates that the request has not been applied because it *lacks valid authentication* credentials for the target resource. The server generating a 401 response MUST send a WWW-Authenticate header field (Section 4.1) containing at least one challenge applicable to the target resource.

> If the request included authentication credentials, then the 401 response indicates that authorization has been refused for those credentials. The user agent MAY repeat the request with a new or replaced Authorization header field (Section 4.2). If the 401 response contains the same challenge as the prior response, and the user agent has already attempted authentication at least once, then the user agent SHOULD present the enclosed representation to the user, since it usually contains relevant diagnostic information.

- 403 is for unauthorized ("refuses to authorize"); i.e. 'I know who you are, but you don't have permission to access this resource.'

> **403 Forbidden**

> The 403 (Forbidden) status code indicates that the server understood the request but *refuses to authorize* it. A server that wishes to make public why the request has been forbidden can describe that reason in the response payload (if any).

> If authentication credentials were provided in the request, the server considers them insufficient to grant access. The client SHOULD NOT automatically repeat the request with the same credentials. The client MAY repeat the request with new or different credentials. However, a request might be forbidden for reasons unrelated to the credentials.

> An origin server that wishes to "hide" the current existence of a forbidden target resource MAY instead respond with a status code of 404 (Not Found).

Share Improve this answer

Follow

edited Oct 7, 2021 at 10:58

Community Bot

1 ● 1

answered Jun 5, 2018 at 15:26

cjbarth

4,469 ● 6 ● 45 ● 65

2 -1; these passages have already been quoted in other answers here, and yours adds nothing new. I'd argue that it's patently *not* clear what the distinction is; you summarise the two codes as "lacks valid authentication" and "refuses to authorise" but I cannot conceive of any situation in which one of those short descriptions would apply where the other could not be interpreted to apply as well. – Mark Amery Jun 5, 2018 at 15:59

There are many answers here that cover many RFC's and are edited and updated muddying the waters. I included a link to explain what `authenticated` is and what `authorized` is and left off all outdated RFC's so that the application is clear. – cjbarth Jun 5, 2018 at 17:17

Your edit clarifies your interpretation of the two codes, which seems to match many other people's interpretation. However, I personally believe that interpretation makes little sense. The use of the phrase *"**If** authentication credentials were provided"* in the 403 description implies that a 403 can be appropriate even if no credentials were provided - i.e. the "unauthenticated" case. Meanwhile, to me the most natural interpretation of the phrase *"for the target resource"* being included in the 401 description is that a 401 can be used for a user who is authenticated but not authorized. – Mark Amery Jun 6, 2018 at 11:36

1

I have a slightly different take on it from the accepted answer.

It seems more semantic and logical to return a 403 when authentication fails and a 401 when authorisation fails.

Here is my reasoning for this:

When you are requesting to be authenticated, You are authorised to make that request. You need to otherwise no one would even be able to be authenticated in the first place.

If your authentication fails you are forbidden, that makes semantic sense.

On the other hand the forbidden can also apply for Authorisation, but Say you are authenticated and you are not authorised to access a particular endpoint. It seems more semantic to return a 401 Unauthorised.

Spring Boot's security returns 403 for a failed authentication attempt

Share Improve this answer Follow

answered Apr 6, 2022 at 22:44

user16422658

0

I think it's easier like this:

401 if the credentials you are using is not recognized by the system, for example if it's different realm or something.

if you managed to pass 401

403 if you are not allowed to access the resource, if you get this when you are not authenticated, chances are you won't be getting it even if you are authenticated, the system doesn't check if you have credentials or not.

Disclosure: I haven't read the RFCs.

Share Improve this answer Follow

answered Jul 10, 2023 at 20:47

fireh

21 ● 2

-7

In the case of 401 vs 403, this has been answered many times. This is essentially a 'HTTP request environment' debate, not an 'application' debate.

There seems to be a question on the roll-your-own-login issue (application).

In this case, simply not being logged in is not sufficient to send a 401 or a 403, unless you use HTTP Auth vs a login page (not tied to setting HTTP Auth). It sounds like you may be looking for a "201 Created", with a roll-your-own-login screen present (instead of the requested resource) for the application-level access to a file. This says:

"I heard you, it's here, but try this instead (you are not allowed to see it)"

Share Improve this answer Follow

answered Dec 12, 2014 at 19:01

Shawn

1

What exactly is being created? – Grant Gryczan Jun 9, 2018 at 1:25

The question states/asks "a user does not have sufficient privileges", there is no scenario I can think of where your "201" would be anything other than entirely wrong and utterly confusing for the client. Especially if my request is not related to "create", ie if I just want to login or GET something I'd expect a 200. – James Jan 19, 2022 at 11:11