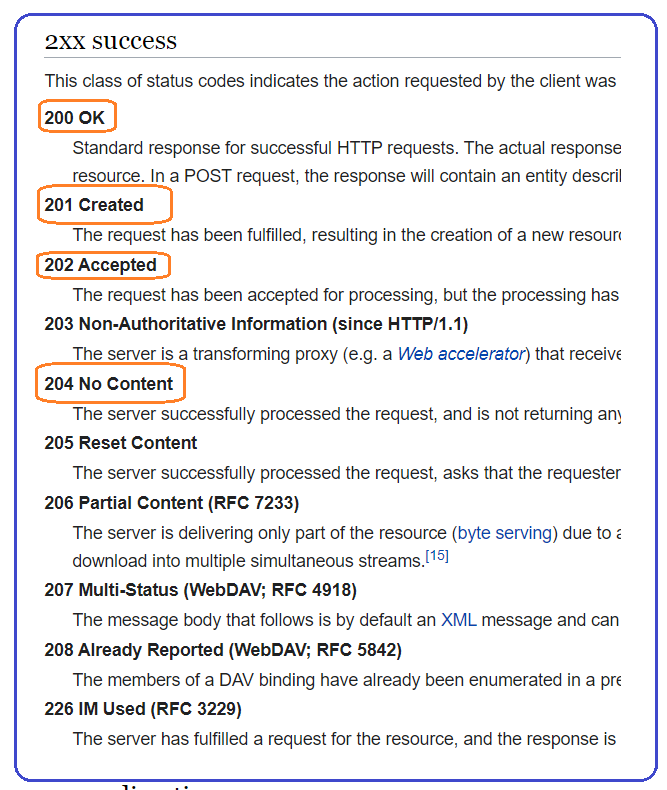
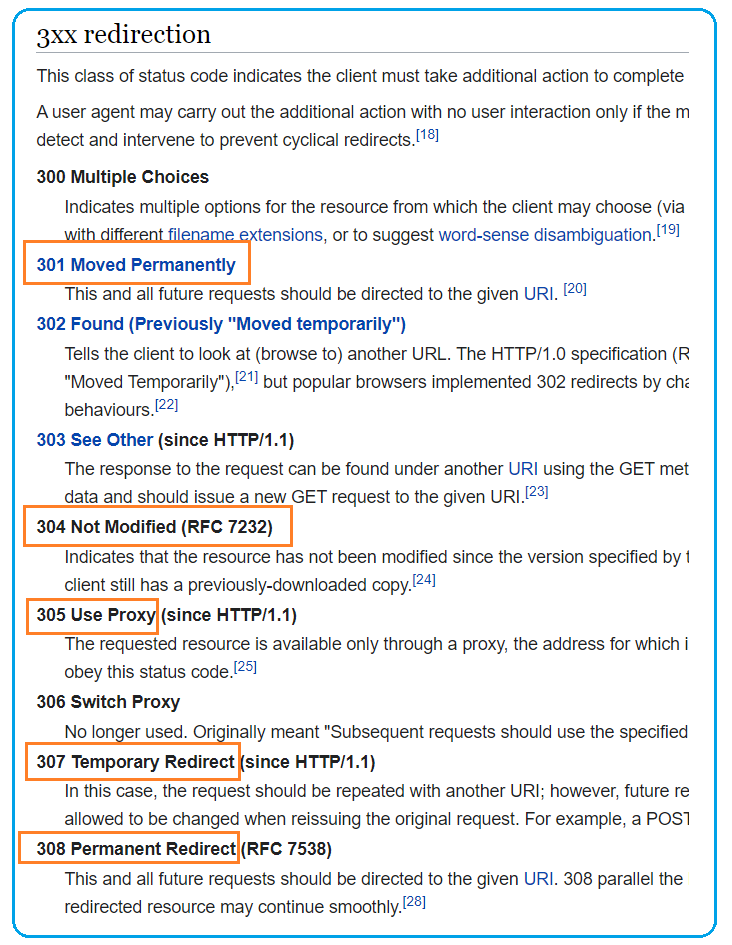
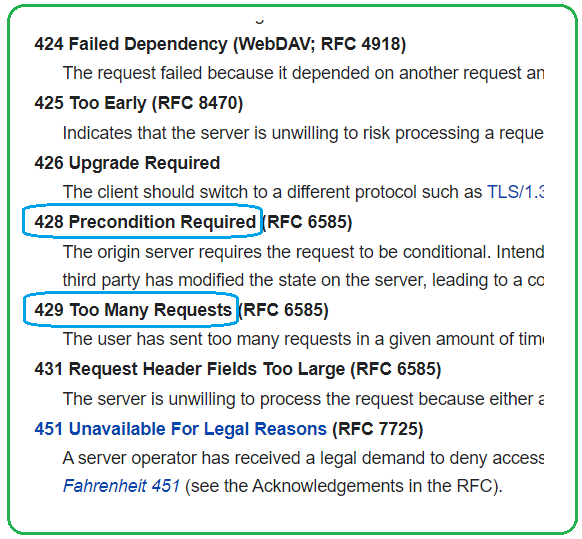
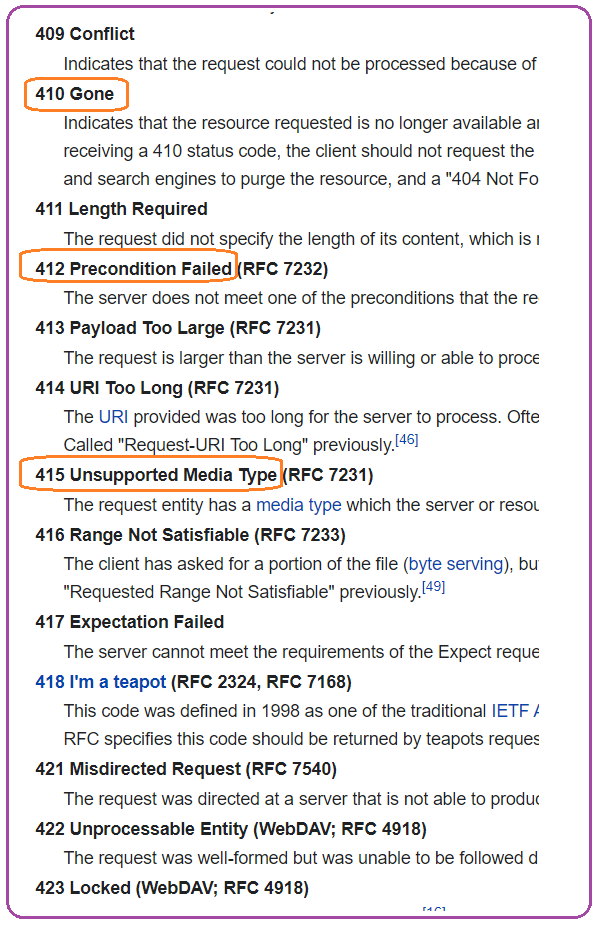
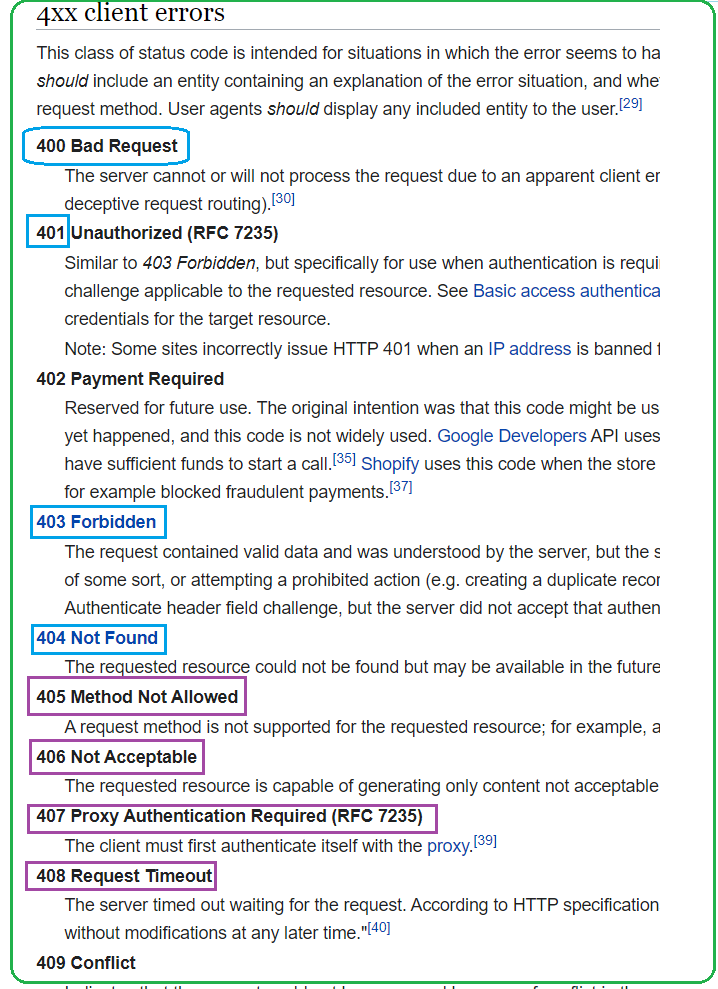
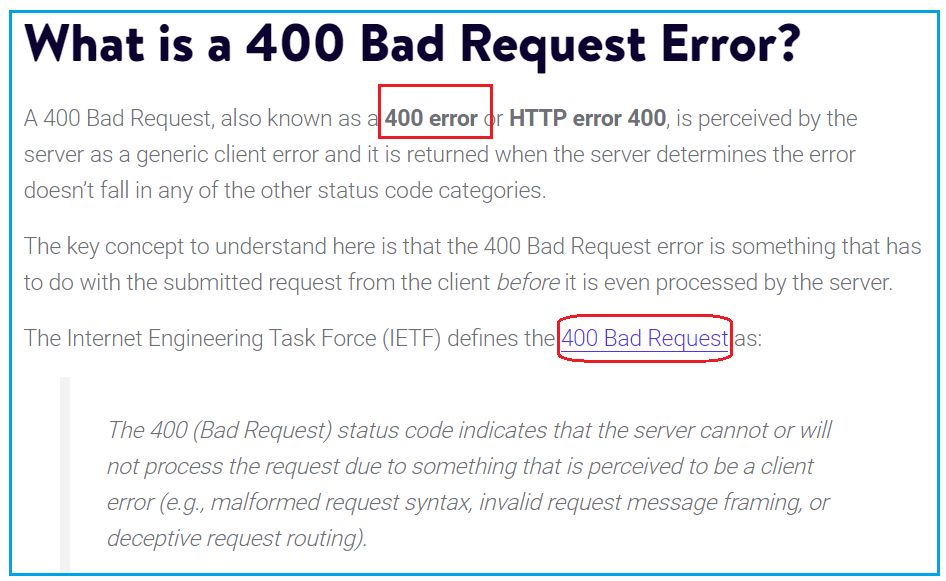
2022

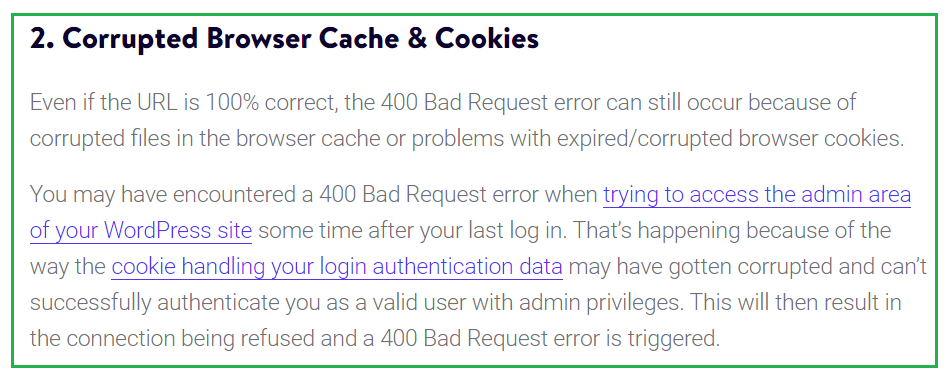
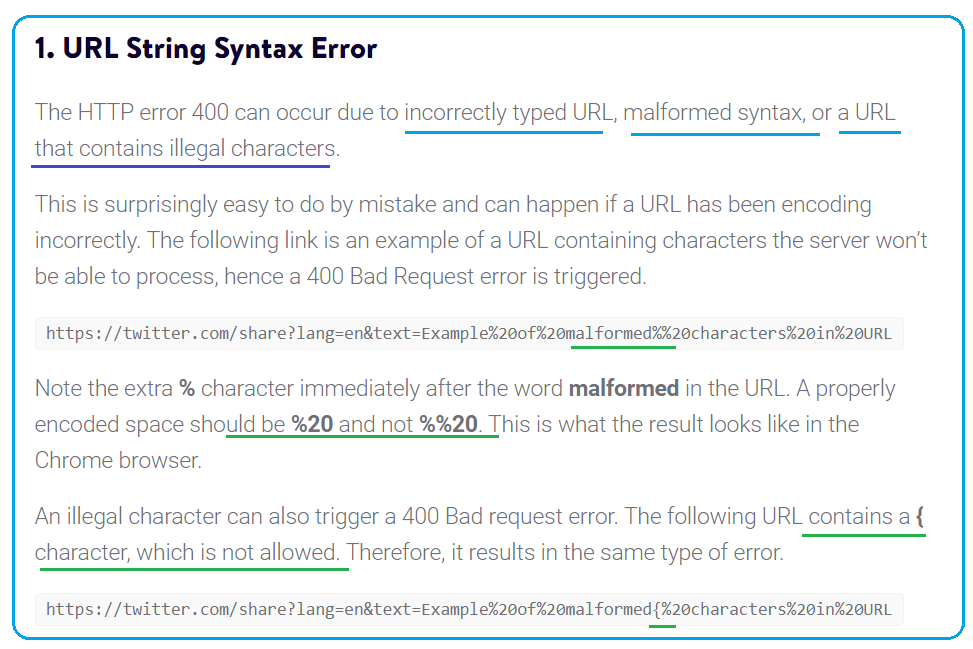
1/13/22 🡪 1/15/22

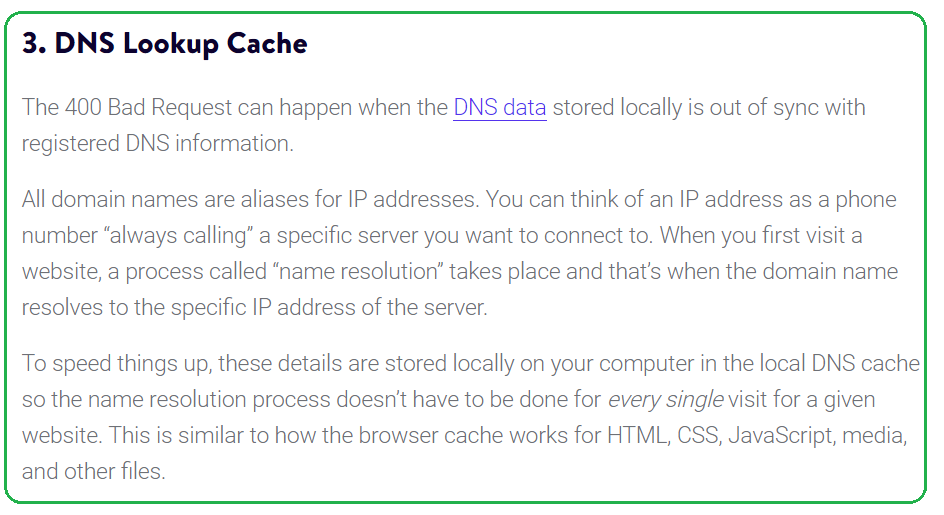


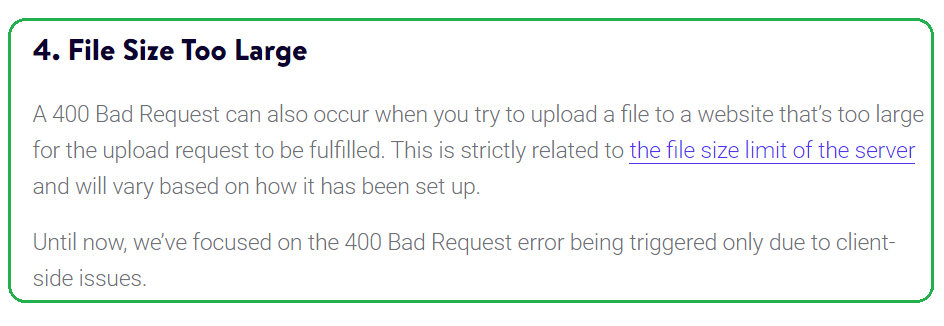
s



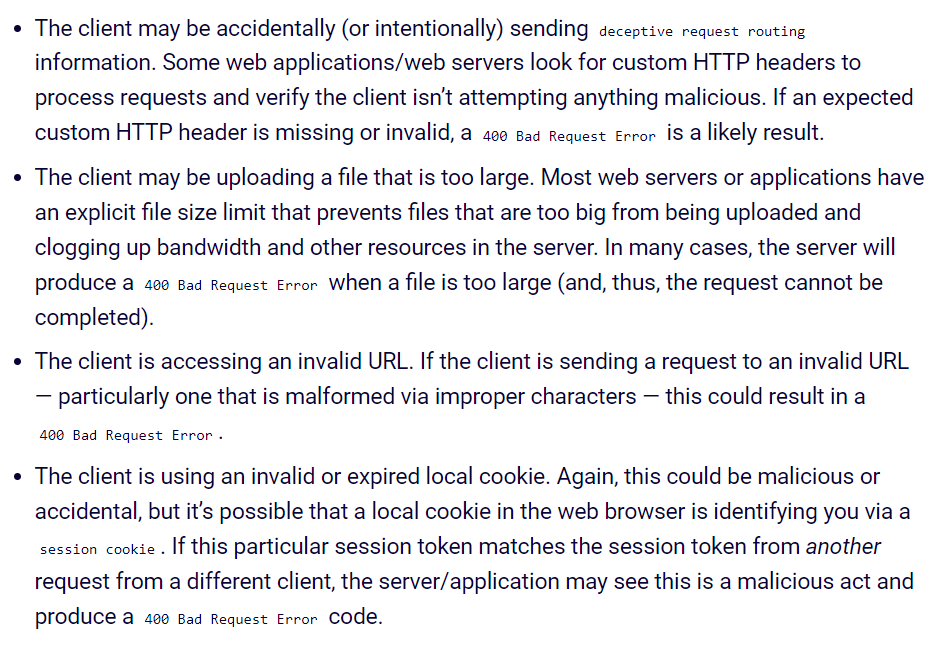


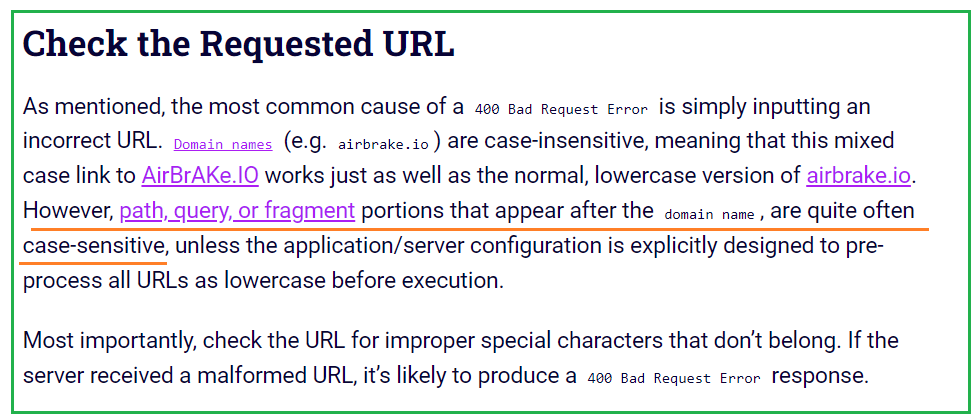


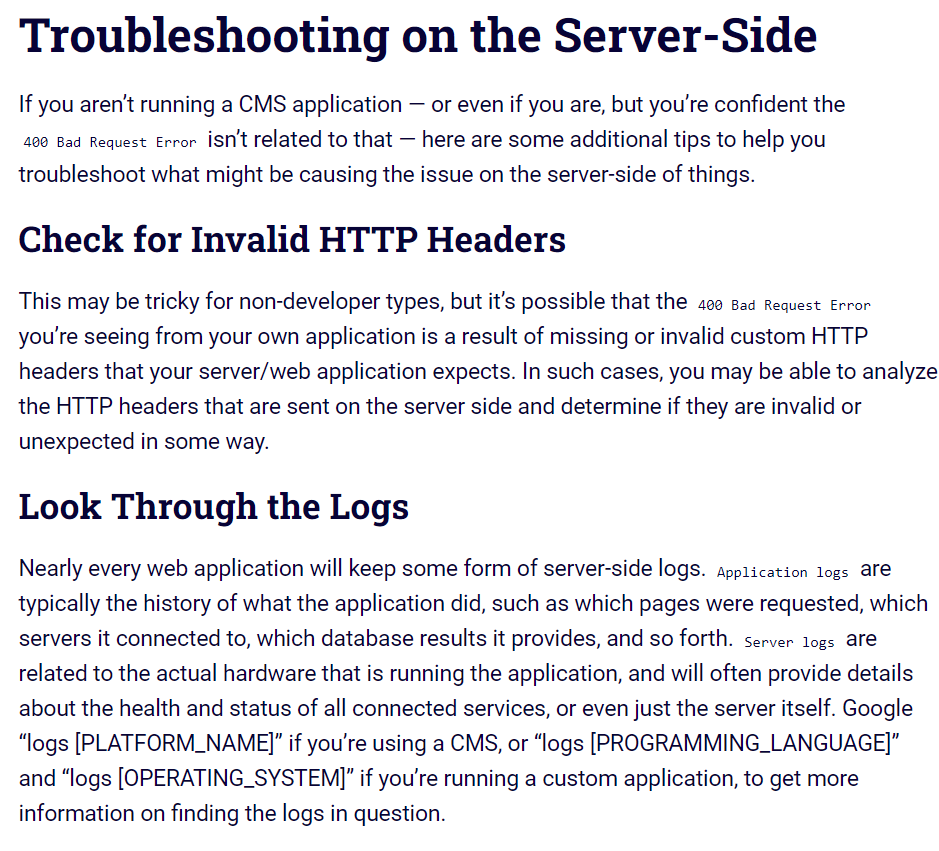
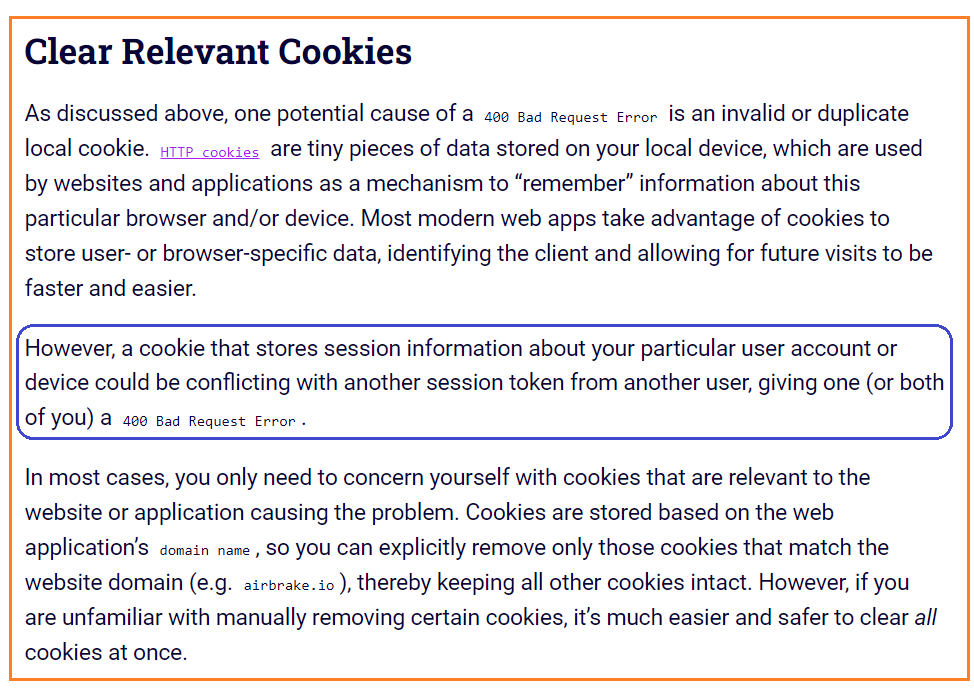


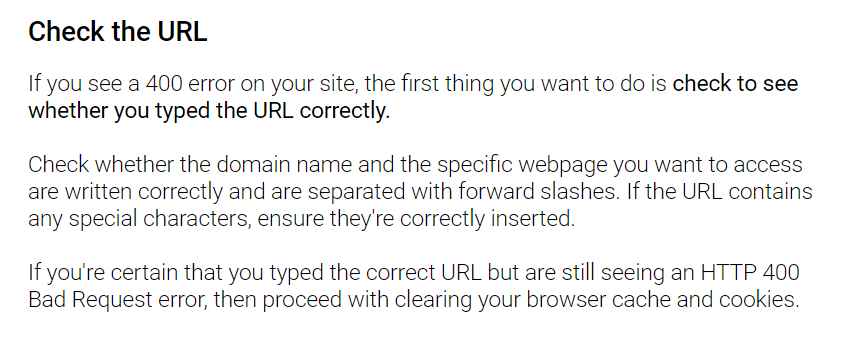


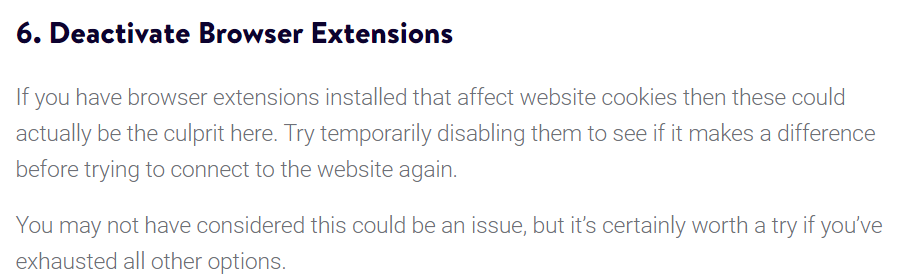
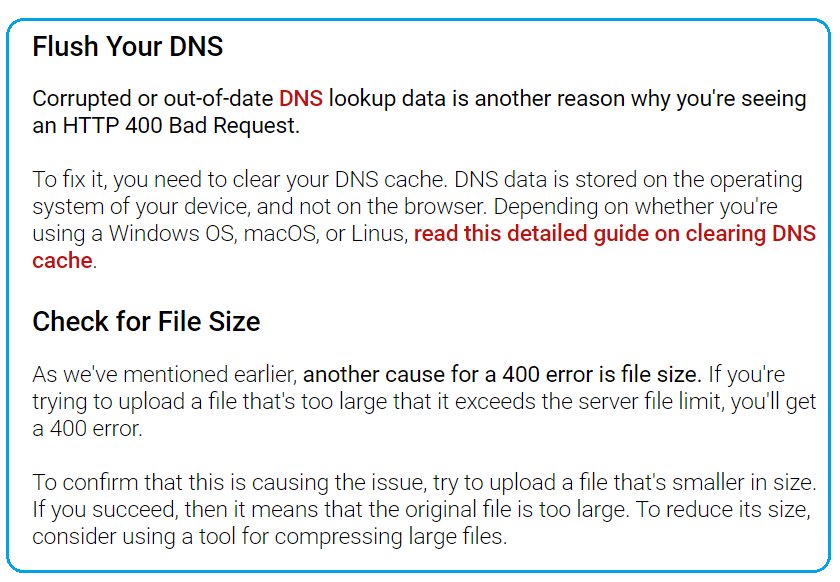
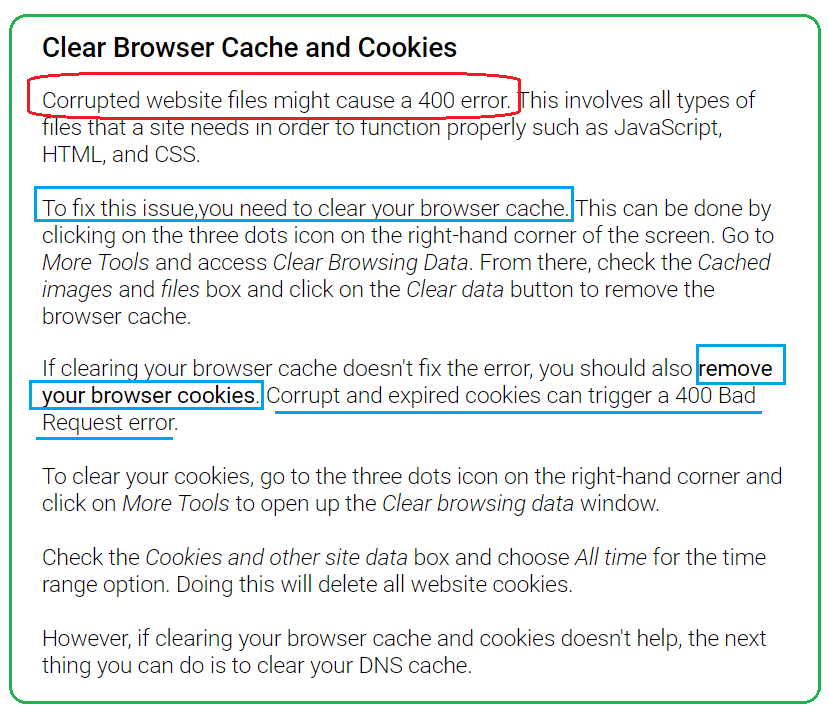
A [400 Bad Request Error](http://httpwg.org/specs/rfc7231.html#status.400) indicates that the server (remote computer) is unable (or refuses) to process the request sent by the client (web browser), due to an issue that is perceived by the server to be a client problem. There are a wide variety of scenarios in which a 400 Bad Request Error could appear in an application, but below are some of the most likely causes:











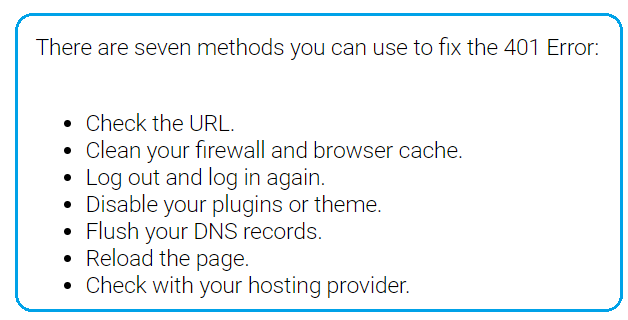
# **401 Unauthorized Error**

A 401 Unauthorized Error is an [HTTP](https://www.exai.com/blog/hypertext-transfer-protocol) status code that indicates that the server received an unverified request.

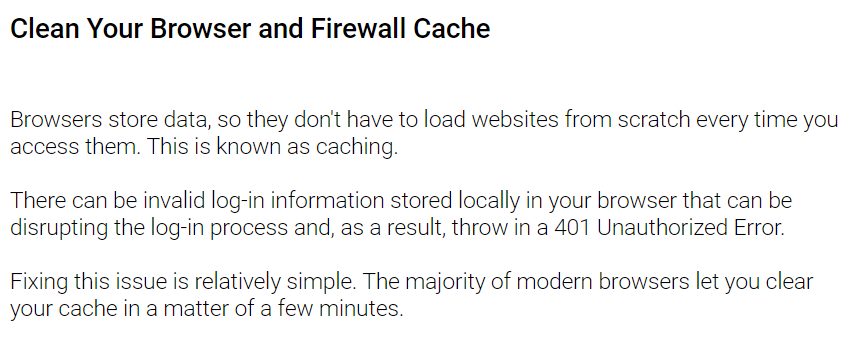
The**401 Unauthorized Error** is an HTTP status code error that represented the request sent by the client to the server that lacks **valid authentication** credentials.

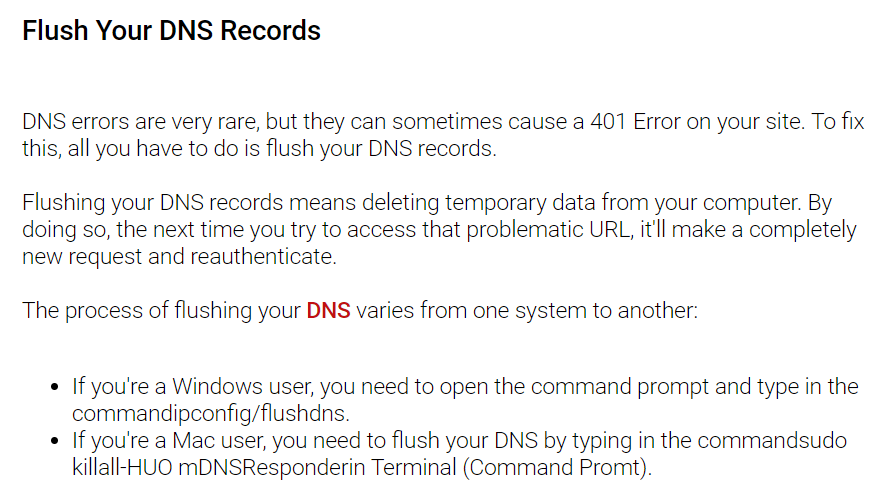
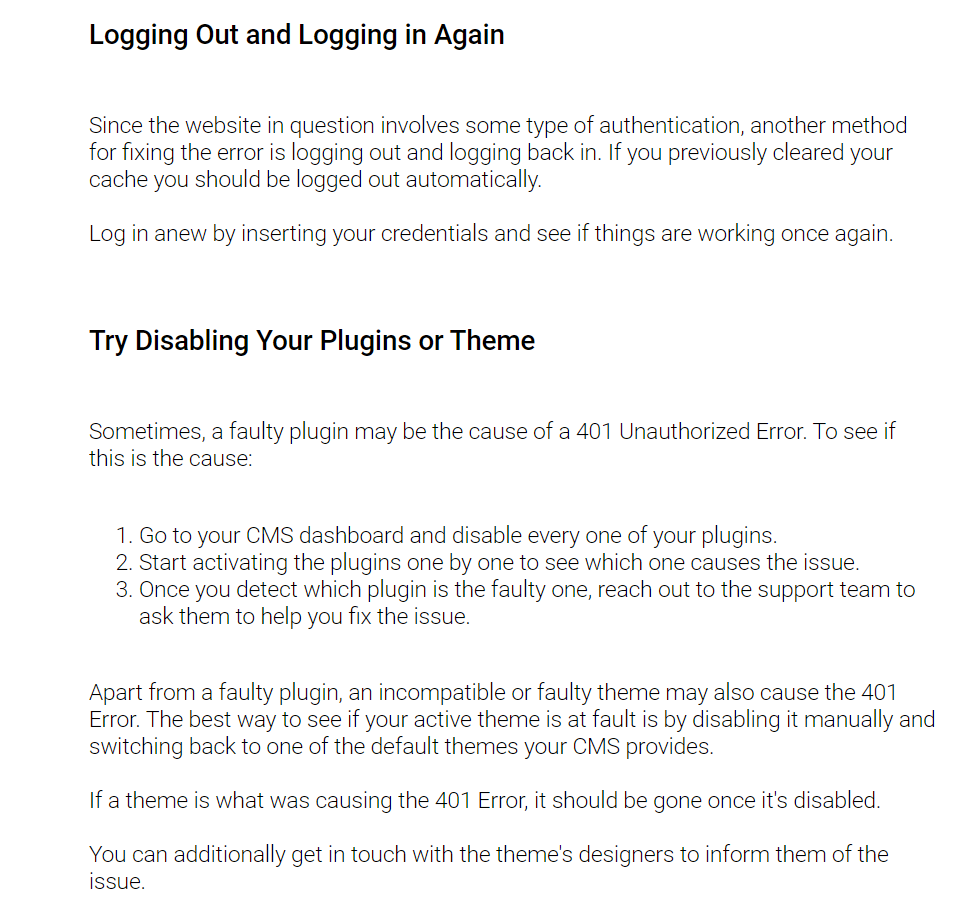
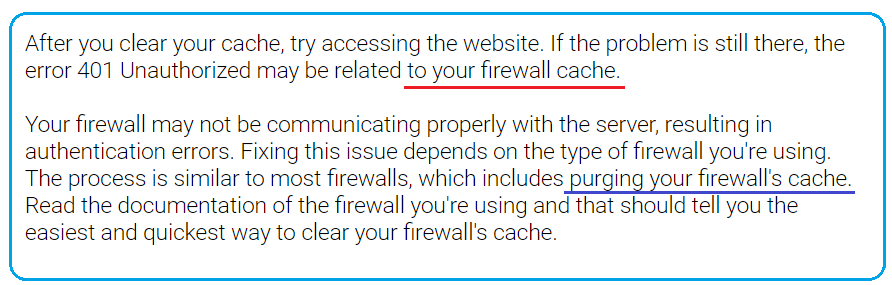
In human terms, this means that the website you're trying to access won't load until you log-in with a valid user ID and password. This type of client error occurs on restricted resources, such as password-protected pages of your site that require authentication credentials.

Consequently, instead of gaining access to the web page you requested, the web browser will show you an error message. 401 Unauthorized Error messages can appear in any browser, so the messages appearing may vary from one browser to another



#### **Clear Relevant Cookies**





**401 Unauthorized Error Occur:** This error may occur due to the reasons described below:

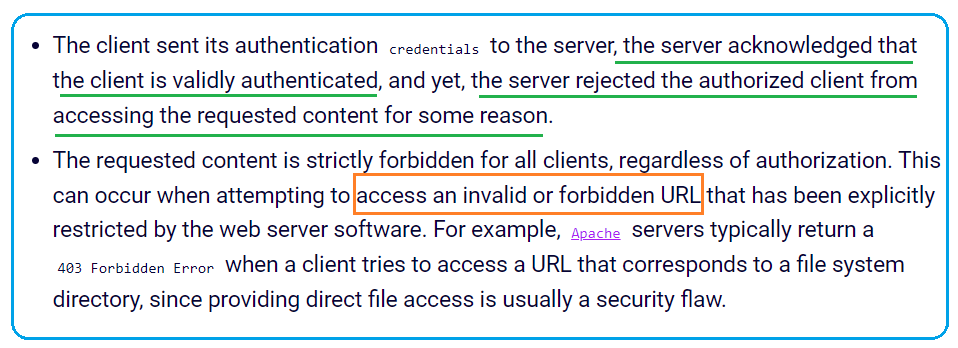
* It may occur client **does not provide the proper authentication** credentials to the server within the request time.
* **It may occur when the server rejects the request** of the client for some reason even though the client provides proper authentication credentials.
* When the client is **banned for some reason by the server**.

**Methods to rectify the error:** The 401 Unauthorized error can be fixed by using any of the following ways:

* **Check The URL:** Due to manual errors in typing the URL, the 401 unauthorized error may occur. Hence, checking the URL and rectifying the mistakes in it will fix the 401 error status.
* **Flush the DNS:** Errors in DNS also creates 401 error status sometimes. Therefore, clearing the DNS will also rectify this error. In Windows, the DNS can be flushed by typing **ipconfig/flushdns** in the command prompt and clicking on **ENTER.**
* **Clear Browser Cookie:** In some situations, the cookies may not work smoothly leading to improper server authentication. Thus, by clearing the cookies, the error can be rectified.
* **Logging out and Logging in again:** This error may also occur during the maintenance time of the websites. Therefore, visiting the website and logging in again by providing the credentials may also rectify this error.
* **Website mistake:** A few times all the above things are good or accurate but still you will get the 401 Unauthorized Error, which is a mistake of the website. That time you need to contact the webmaster of that website and inform that the server is down. You can email them at webmaster@webmaster.com replace the webmaster.com with the website, or you can see the contact us option on any website through that you can inform them.

# **403 Forbidden Error**

The **403 Forbidden Error** is an [HTTP response status code](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status), which indicates that the identified client does not have proper authorization to access to the requested content.



**Check the Requested URL**

#### **Clear Relevant Cookies**

#### **Clear the Cache**

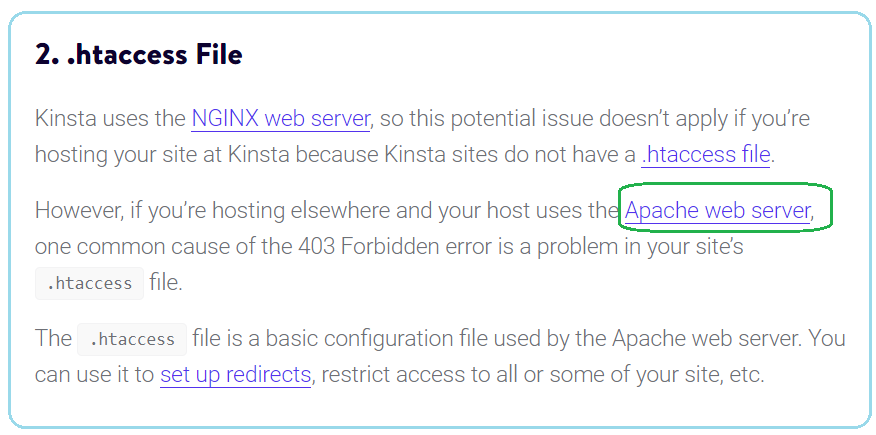
#### **Log Out and Log In**

**. File Permissions**

Each folder and file on your WordPress site’s server has its own unique file permissions that control who can:

* **Read** – see the data in the file/view the contents of a folder.
* **Write** – modify the file/add or delete files inside a folder
* **Execute** – run the file and/or execute it as a script/access a folder and perform functions and commands.

These permissions are indicated by a 3-digit number, with each digit indicating the level of permission for each of the 3 categories above.



## What causes the 403 Forbidden error?

In a nutshell, you see the 403 Forbidden error when you try to access a page or file on a website that you don’t have permission to access. The website’s server is functioning – there’s just some type of permissions issue getting in the way of you seeing the content, which is why the server responds with the 403 [HTTP error code](https://themeisle.com/blog/what-are-http-error-codes/). You might also see this error displayed as an HTTP 403, Error 403, or simply Forbidden

This problem typically results from issues with file permissions or [your site’s .htaccess file](https://themeisle.com/blog/htaccess-file-in-wordpress/). On WordPress sites, the error also might result from an issue with a plugin that you’re using

List of all 4xx status codes[[edit](https://en.ryte.com/wiki/index.php?title=Status_Code_400&action=edit&section=4)]

Errors with the status code 4xx are listed below:

**\* 400 bad request**: All errors with the status code 4xx indicate an invalid request from a client to a server.

**\* 401 unauthorized**: This request to the server requires the client to authorize. This is usually done by logging in. If a user still wants to access the password-protected resource, the status code 401 (unauthorized) appears with a note on what to do.

**\* 402 payment required**: This status code is for future use. It will indicate that you have to pay a fee in order to view the [content,](https://en.ryte.com/wiki/Content) and that it will only be visible after payment.

**\* 403 forbidden**: Access is denied even with valid login data. This can happen for example when a site is requested with [HTTP](https://en.ryte.com/wiki/HTTP), but is configured with HTTPS.

**\* 404 not found**: One of the most frequently displayed status codes is the 404-error. It is used to indicate that a requested link does not exist. If the error page displays a different HTTP status code to 404, this results in a [soft 404 error](https://en.ryte.com/wiki/Soft_404_Error).

**\* 405 method not allowed**: The request was made using the wrong request method. The required method, such as GET or POST is explained by the response within the error code.

**\* 406 not acceptable**: In this case, the format requested by the client cannot be issued by the server. The content type is available in the server response.

**\* 407 proxy authentication required**: Similar to status code 401, the server requests authentication by the client, in this case in relation to the [proxy server](https://en.ryte.com/wiki/Proxy_Server) being used.

**\* 408 request time-out**: This code is displayed if the client could not send a complete request in the time period defined by the server.

**\* 409 conflict**: This request by the client is rejected by the server because it was submitted under a false assumption. This status code may get output if the resource has changed.

**\* 410 gone**: If the user receives this status code, it means that the resource is no longer available and/or has been deleted.

**\* 411 length required**: If this code appears, the content length needs to be specified in the header to process the client request.

**\* 412 precondition failed**: In this case, the prerequisite has been defined in the request, which does not apply. (for example, an if match)

**\* 413 request entity too large**: This indicates that the request was too big to be processed by the server in question. The server response may include the instruction to try again later.

**\* 414 request-URI too long**: The server cannot respond because the URI is too long. This is usually caused by too many redirects.

**\* 415 unsupported media type**: This request cannot be answered because the media type is not available.

**\* 416 request range not satisfiable**: This error code indicates that a portion of the requested resource is no longer available or invalid.

**\* 417 expectation failed**: This code will be output if the “expect” field of the [header](https://en.ryte.com/wiki/Head) specifies a particular request that the server that cannot fulfill.

**\* 422 unprocessable entity**: This code indicates that the request cannot be processed. This could be caused by semantic errors, but not media type errors as is the case with 415.

**\* 423 locked**: The requested resource is temporarily locked and not accessible.

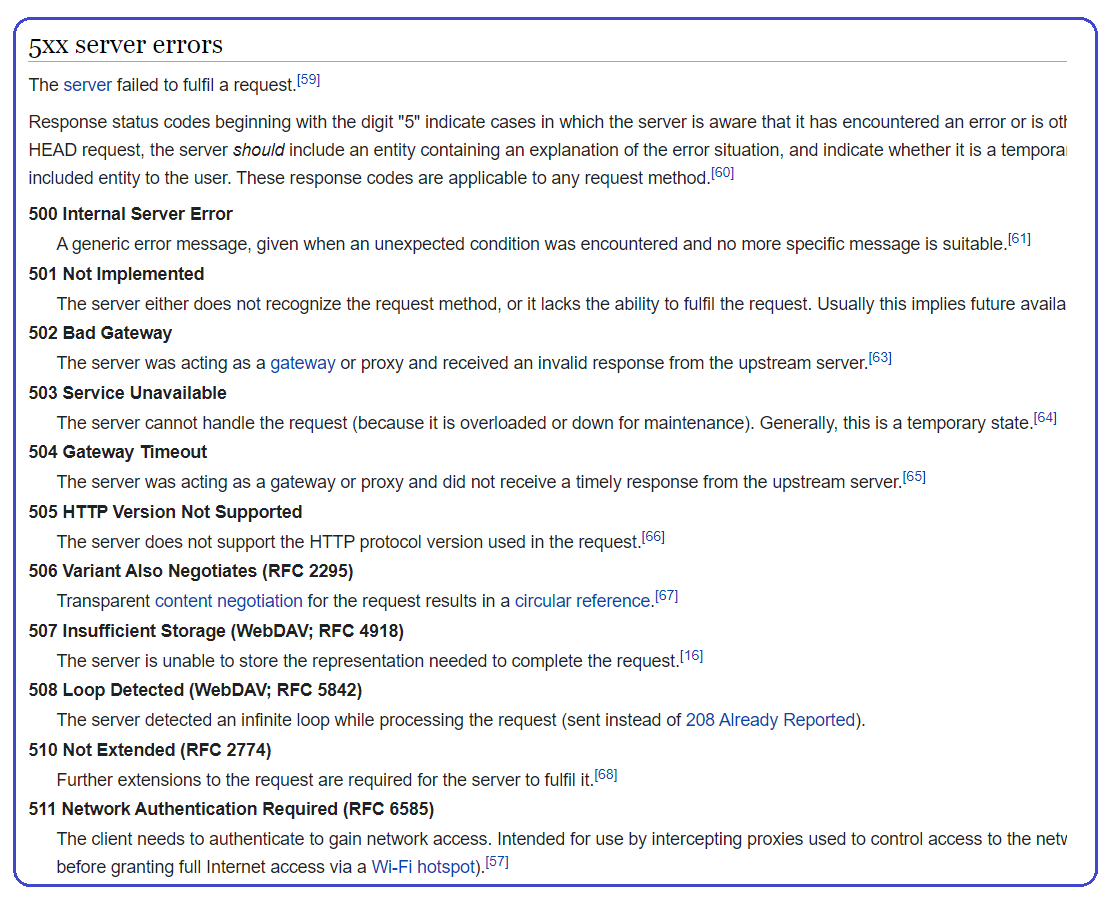
**\* 424 failed dependency**: If this status code is issued, two requests were made. The second request depended on the first, but was unsuccessful.

**\* 426 upgrade required**: In order for the server to handle this request, the client must use [TLS](https://en.ryte.com/wiki/Transport_Layer_Security_(TLS)) 1.0.

**\* 428 precondition required**: Preconditions must be fulfilled In order to execute this request successfully.

**\* 429 too many requests**: This code is issued by the server if it received too many requests from a client within a certain period of time. This may occur, for example, if an [SEO tool](https://en.ryte.com/wiki/SEO_Software) queries too many keywords on Google within a short period of time.

**\* 431 request header fields too large**: If the length of the header field or the entire header has been exceeded, this status code appears.



# **504 Gateway Timeout Error**

A **504 Gateway Timeout Error** is an [HTTP response status code](https://developer.mozilla.org/en-US/docs/Web/HTTP/Status) indicating that a **server, which is currently acting as a gateway or proxy**, **did not receive** **a timely response from another server further upstream (origin Server)**. In another word, Gateway or proxy server did not received a timely response from origin server; so this error is issued by Gateway or proxy server.

To that extent, the HTTP 504 error means that the server running as a gateway is out of time to wait for a server response. The code is returned when there are two servers involved in processing the request, and the response time from the second server has expired.

 In most HTTP communications, a client will connect to a server via a third-party [gateway](https://en.wikipedia.org/wiki/Gateway_(telecommunications)) computer. The gateway acts as, well, a gateway, by which messages from the client can be securely sent to the server, and vice-versa. A gateway acts as a node within the larger network web, connecting and routing communications between multiple client, server, and other node types within the (virtual) vicinity.

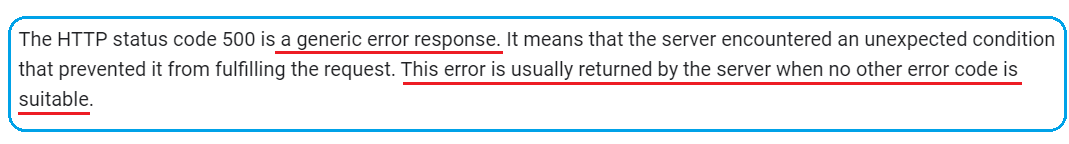
Believe it or not, most homes with Internet access even have an active gateway. Your local home network, which is likely setup through a router (or router+modem hybrid), typically assigns IP addresses to all the devices on your network using the base address of 192.168.1.\*, where the asterisk changes depending on the device. In most cases, communication from one such local network address to another local network address is allowed, but when your computer attempts to connect to an IP address outside of this base range, your router’s gateway will intercept it and perform the communication between your computer and the remote server on your behalf.

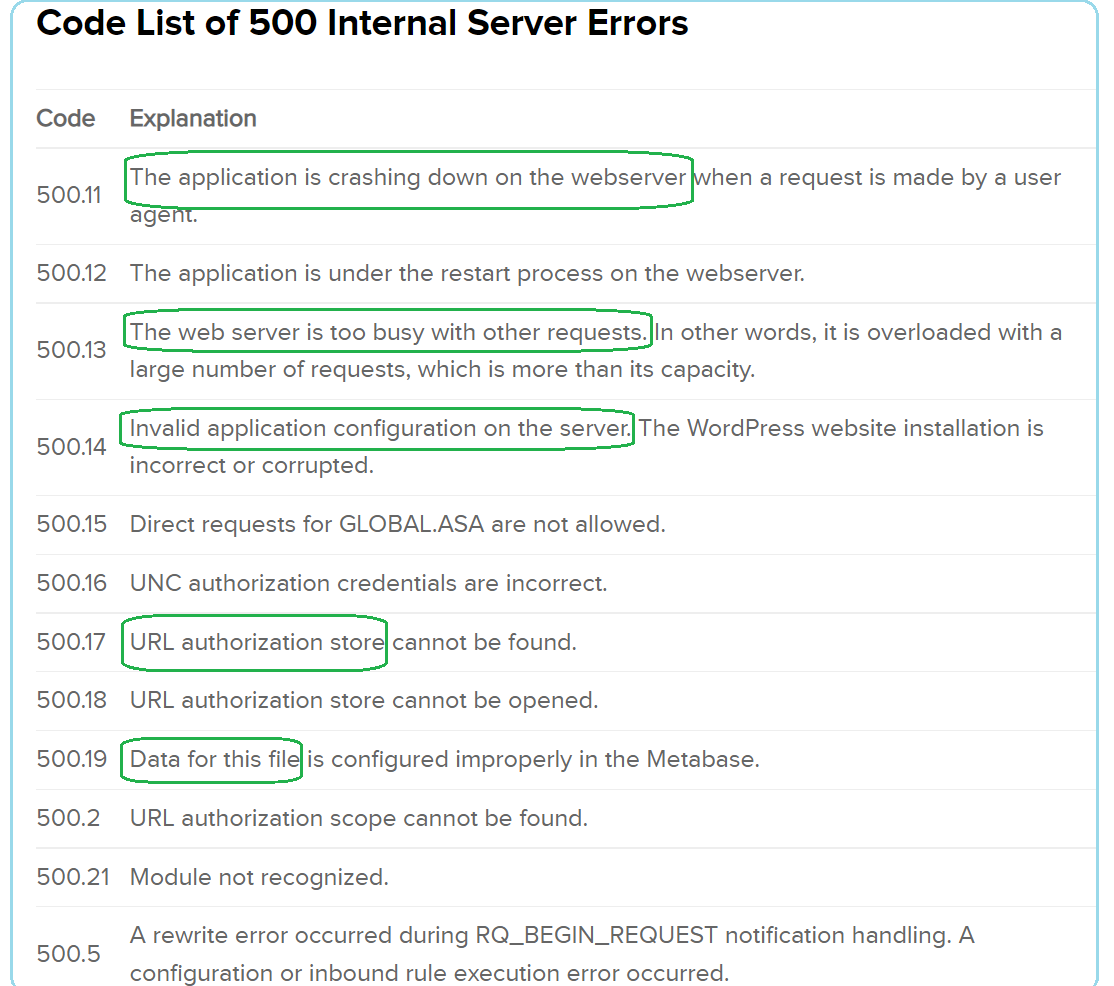
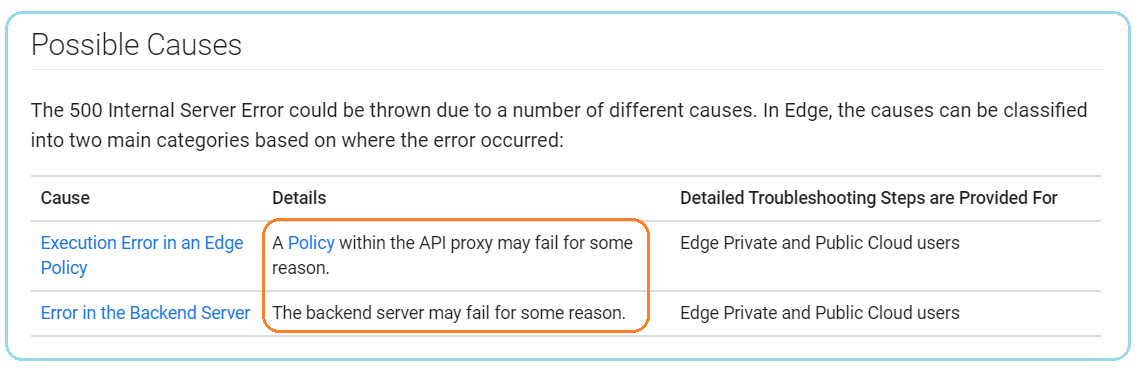
In some situations, the web server running your application *may* be the cause of the problem. This is particularly true when **your server is running either a combination frontend+backend server setup (such as Nginx and Apache)**, or **the web server is relying on third-party services**, **which are typically located elsewhere on additional upstream servers**. Any of the upstream servers that your client (web browser) is connecting through may be down or experiencing issues at this time, which could cause a delay in processing and lead to the 504 Gateway Timeout Error you’re seeing.

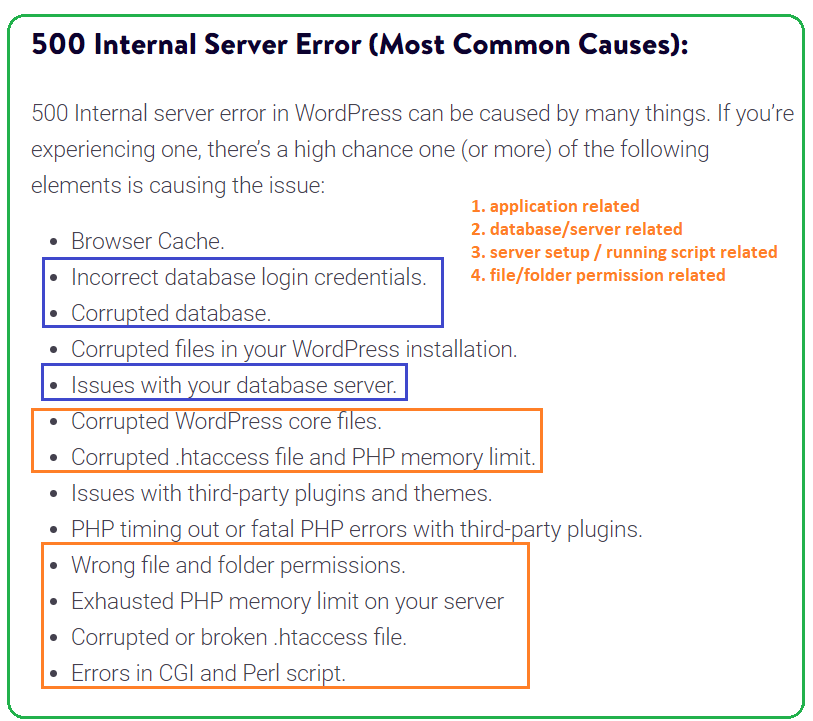
Here are some additional tips to help you troubleshoot what might be causing the 504 Gateway Timeout Error to appear on the server-side of things:

* **Recent DNS Changes** – The Domain Name System (DNS) is a decentralized naming system for devices connected through a network (even a massive network, such as the Internet itself). In short, the DNS associates domain names (e.g. airbrake.io) to specific IP addresses, and stores that association in a series of [authoritative name servers](https://www.whatsmydns.net/) spread around the world. Thus, when you ask your computer to connect to airbrake.io, your computer actually checks with a nearby DNS name server to find out what the specific IP address (internet resource) is that it should connect to. From your perspective it’s going directly to airbrake.io, but behind the scenes the traffic is routed to an IP address (52.203.232.56, in this case). Consequently, your application may present a 504 Gateway Timeout Error if your site has made recent changes to its DNS server, which is a result of changing host servers or moving the site to a different IP address. Such DNS changes, known as DNS propagation, are not instant and can sometimes take a few hours to propagate throughout all the authoritative name servers.
* **Server Connectivity Issues** – While it may sound simple, it’s entirely possible that a 504 Gateway Timeout Error simply indicates that a server somewhere in the chain is down or unreachable for whatever reason. Most modern applications don’t reside on a single server, but may, instead, be spread over multiple systems, or even rely on many third-party services to function. If any one of these servers are down for maintenance or otherwise inaccessible, this could result in an error that *appears* to be from your own application.
* **Improper Firewall Configuration** – A firewall is a basic security device that monitors network traffic and acts as a gatekeeper, deciding which traffic is safe and which could be malicious. In most cases, all potentially harmful traffic is stopped (and may be logged for network admin use). In some situations, it’s entirely possible for a firewall configured somewhere on the network in which your application is running to be preventing some form of critical traffic from getting through. This is particularly true for applications that rely on content delivery networks (CDNs), which act as a third-party host for “heavy” content like images or videos, hosting that content on behalf of your application, so your application can maintain its speed and efficiency. However, automatic firewall services can sometimes perform false positives, mistaking perfectly safe and valid content from CDNs or elsewhere as malicious, thereby shutting off that stream of content in an instant, which could lead to a 504 Gateway Timeout Error.
* **Check the Logs** – Nearly **every web application will keep some form of server-side logs**. Application logs are typically the history of what the application did, such as which pages were requested, which servers it connected to, which database results it provides, and so forth. **Server logs are related to the actual hardware that is running the application**, and will often provide details about the health and status of all connected services, or even just the server itself. Google “logs [PLATFORM\_NAME]” if you’re using a CMS, or “logs [PROGRAMMING\_LANGUAGE]” and “logs [OPERATING\_SYSTEM]” if you’re running a custom application, to get more information on finding the logs in question.
* Application Code or Script Bugs – If all else fails, it may be that a problem in some custom code within your application is causing the issue. Try to diagnose where the issue may be coming from through manually debugging your application, along with parsing through application and server logs. Ideally, make a copy of the entire application to a local development machine and perform a step-by-step debug process, which will allow you to recreate the exact scenario in which the 504 Gateway Timeout Error occurred and view the application code at the moment something goes wrong.
* Slow server. It is possible that the server where you host your WordPress website is responding too slowly, and therefore it generates gateway errors.
* Insufficient PHP workers. PHP workers are used to running code on your WordPress site. Demanding sites can make all PHP workers busy. In that case, they form the queue. If the queue and the backlog are full, old requests are ignored. You can ask your hosting to increase the number of PHP workers. Additional PHP workers for a site allow executing several requests simultaneously.
* Problems with the firewall. The firewall on your server may contain errors due to incorrect configuration or rules blocking the connection.
* Network connection. If there is a problem with the network connection between the proxy server and the web server, this can lead to delays. Also, there may be network problems with the load balancer, if it is used.

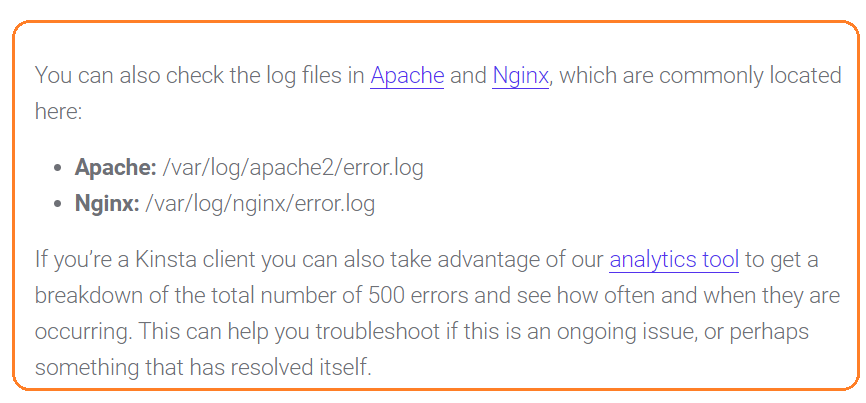
# **500 Internal Server Errors**

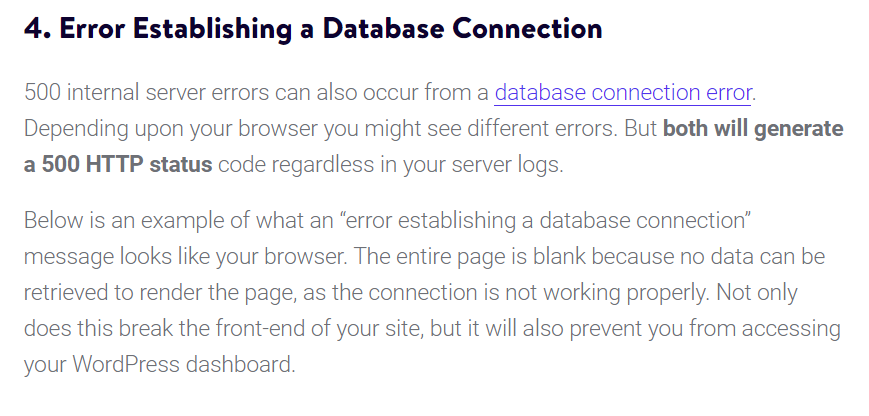


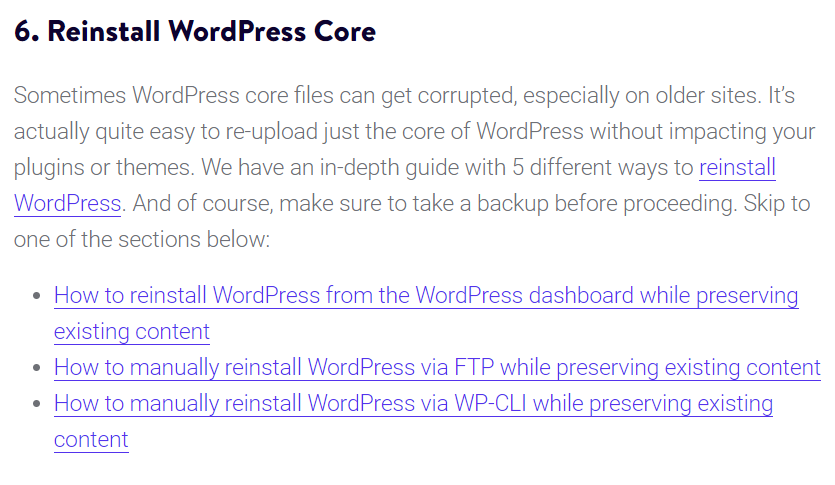
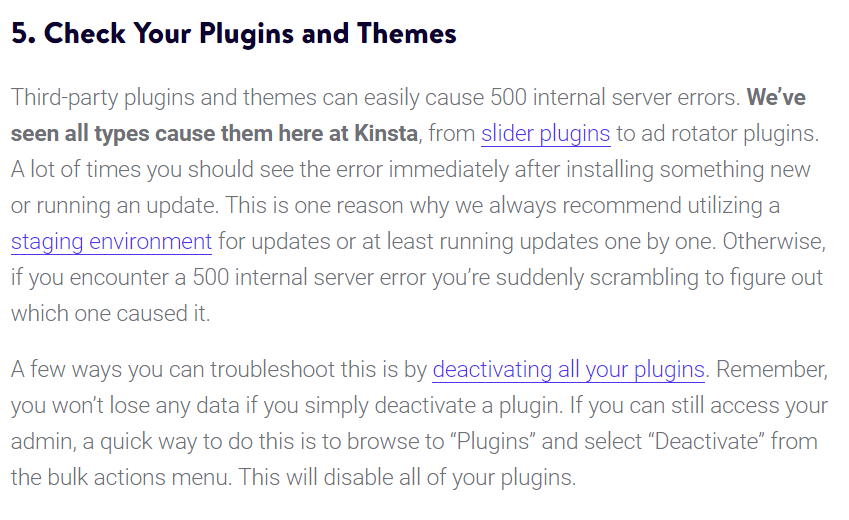


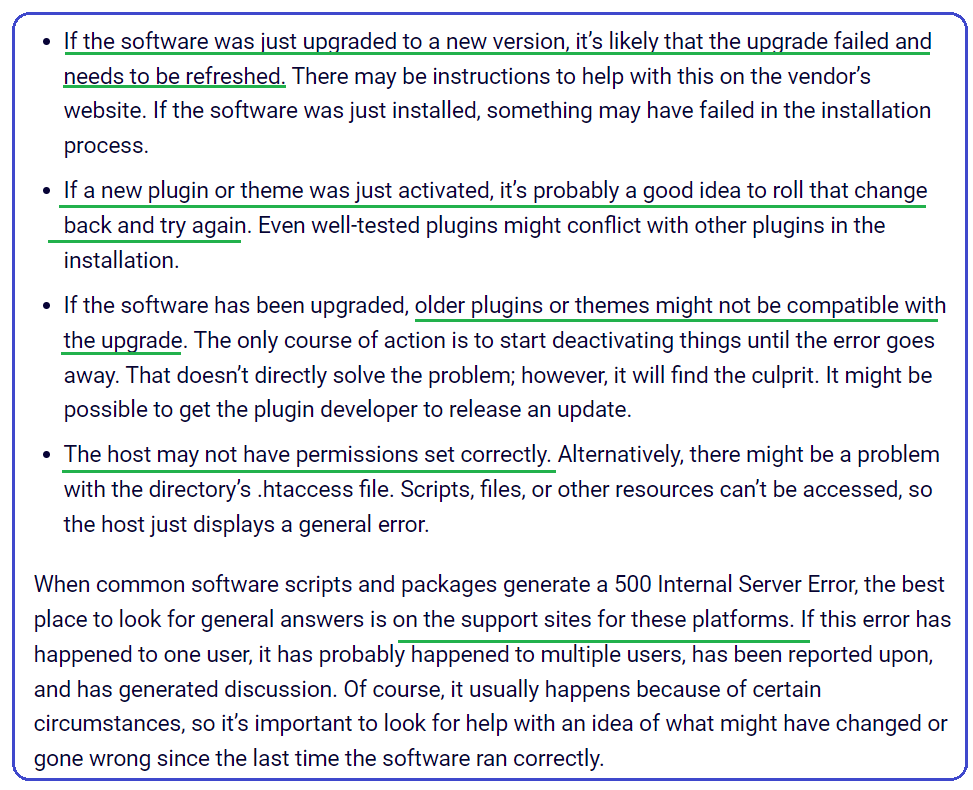
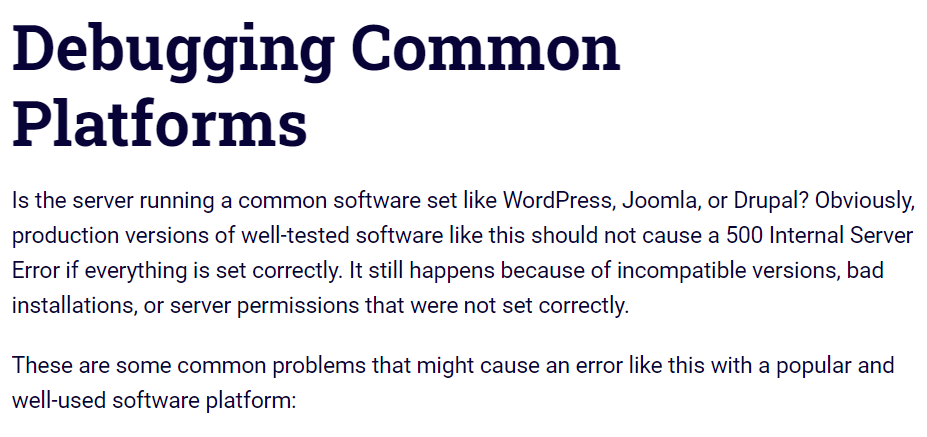
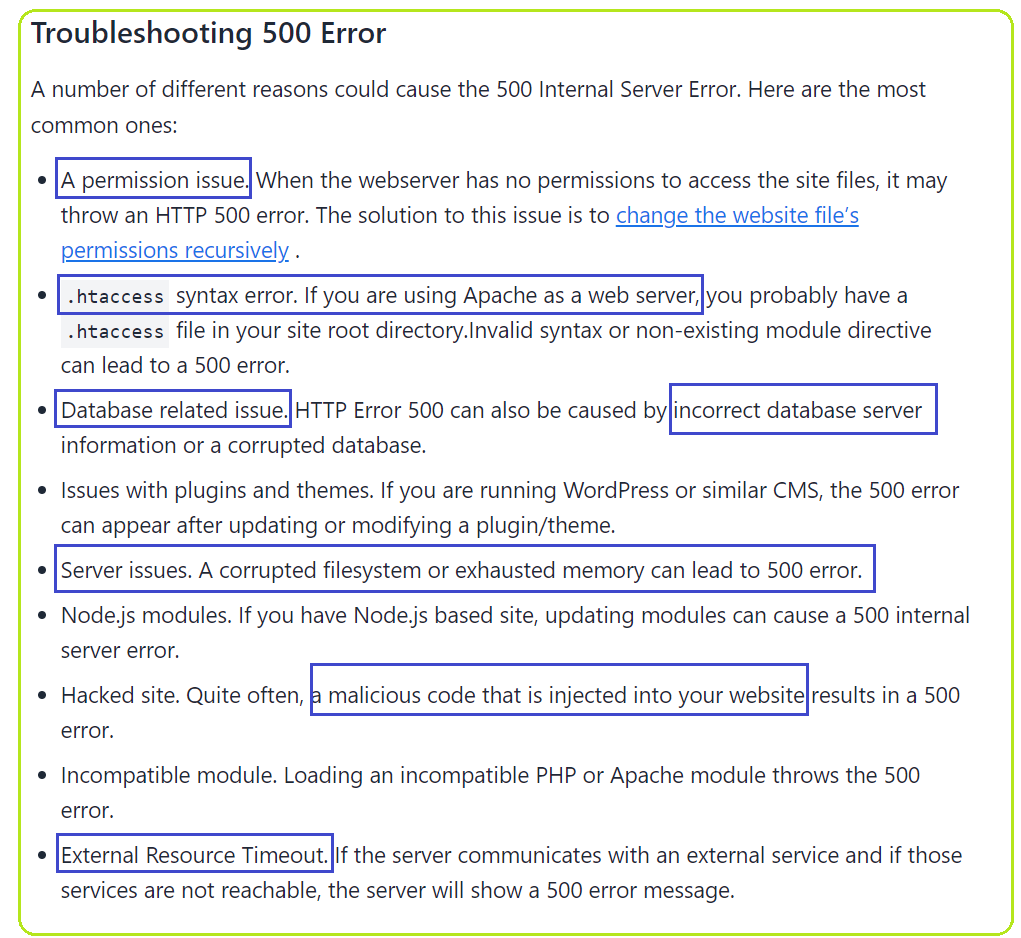
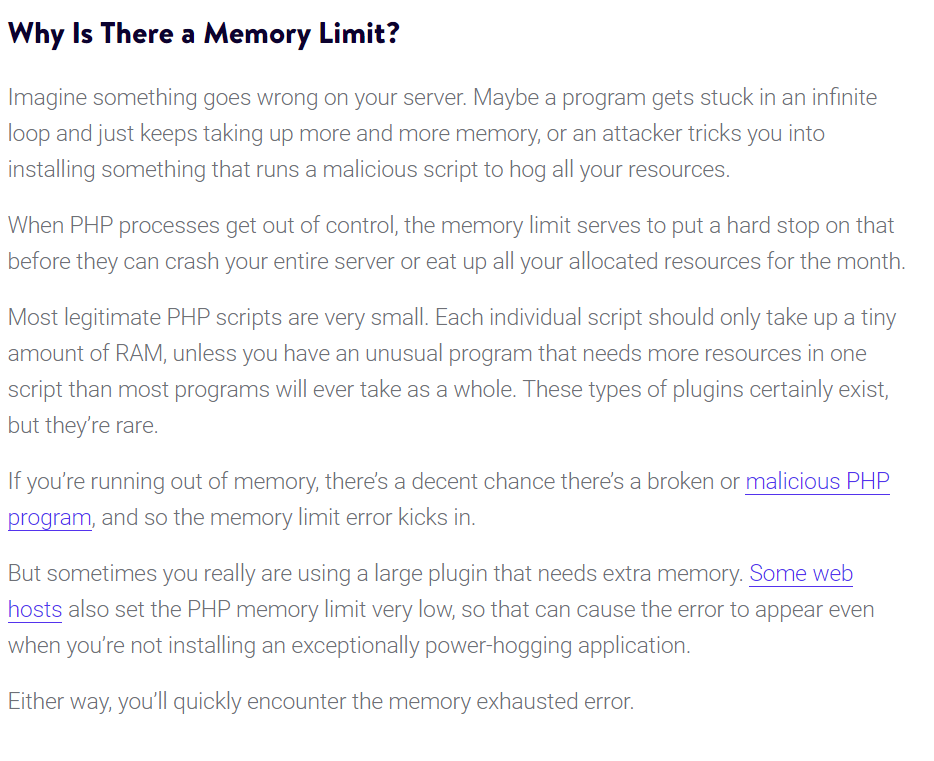
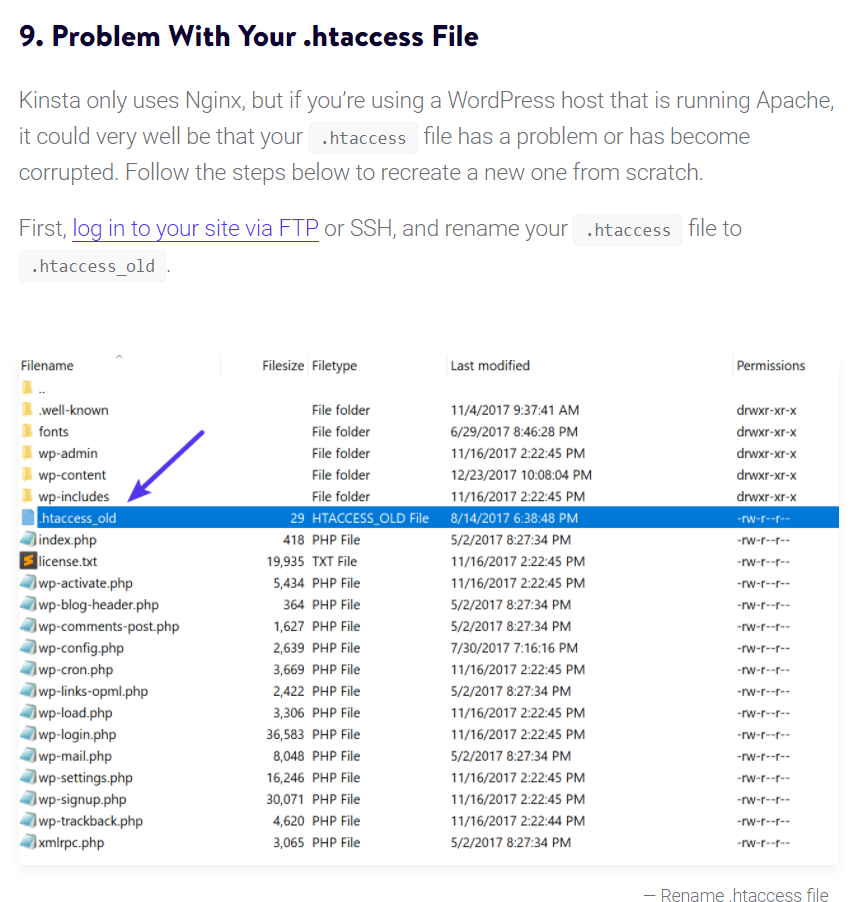
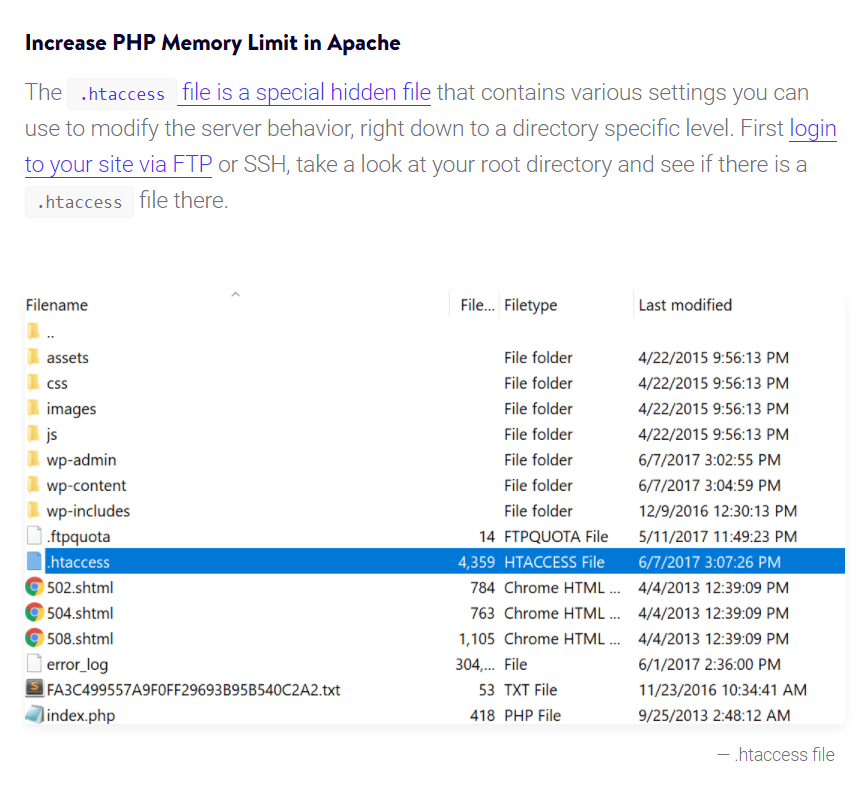
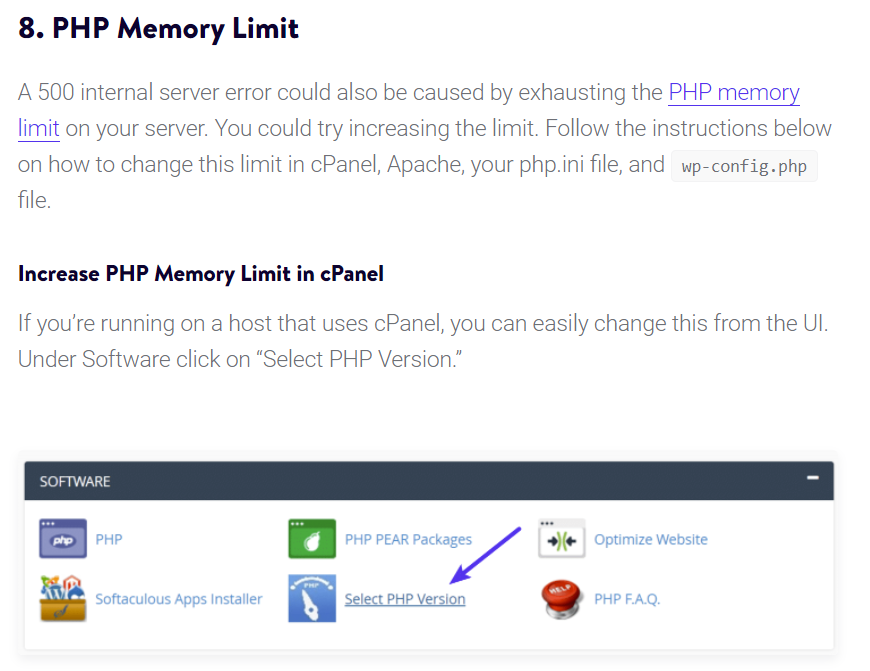
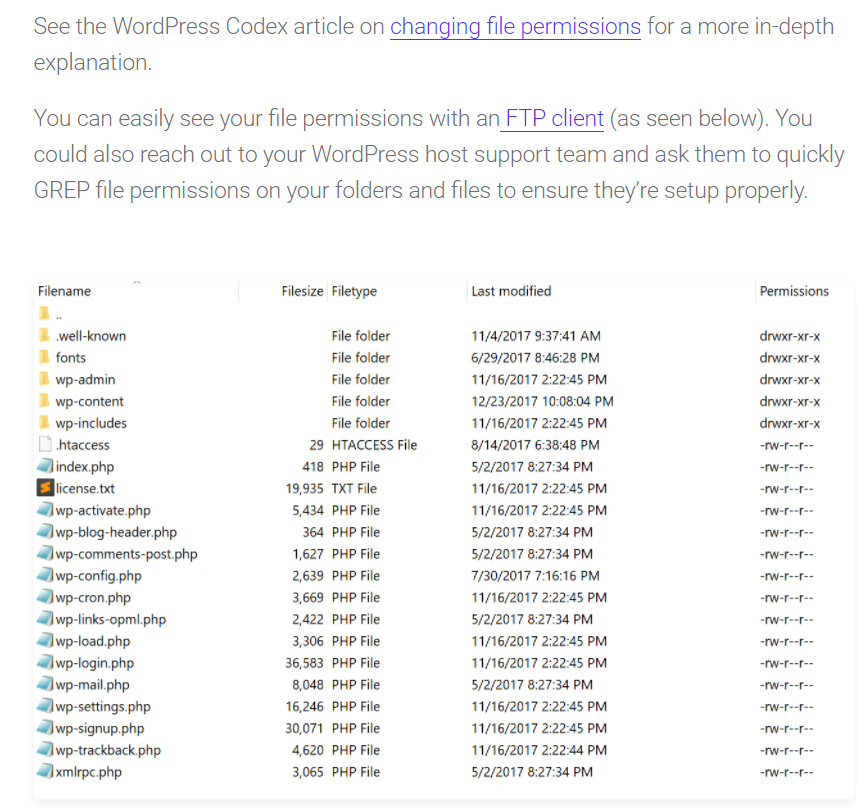
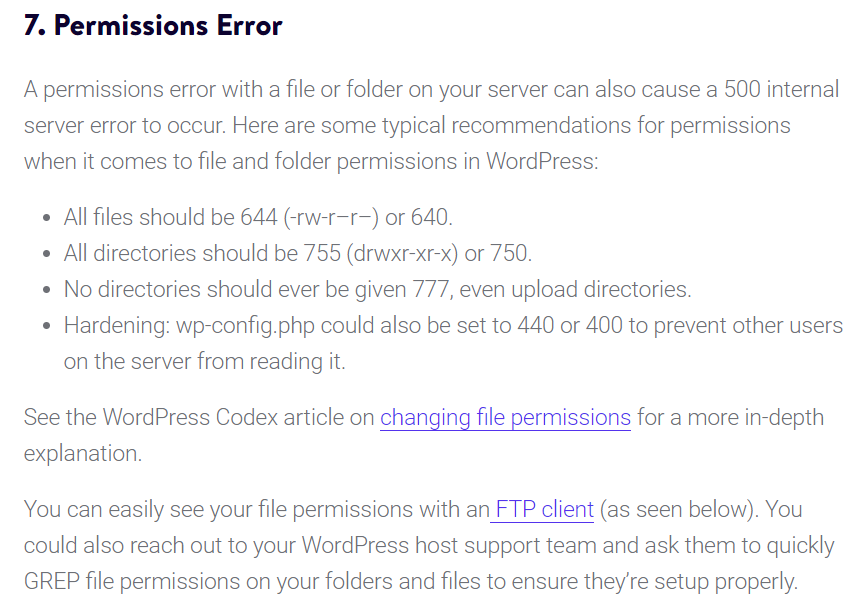


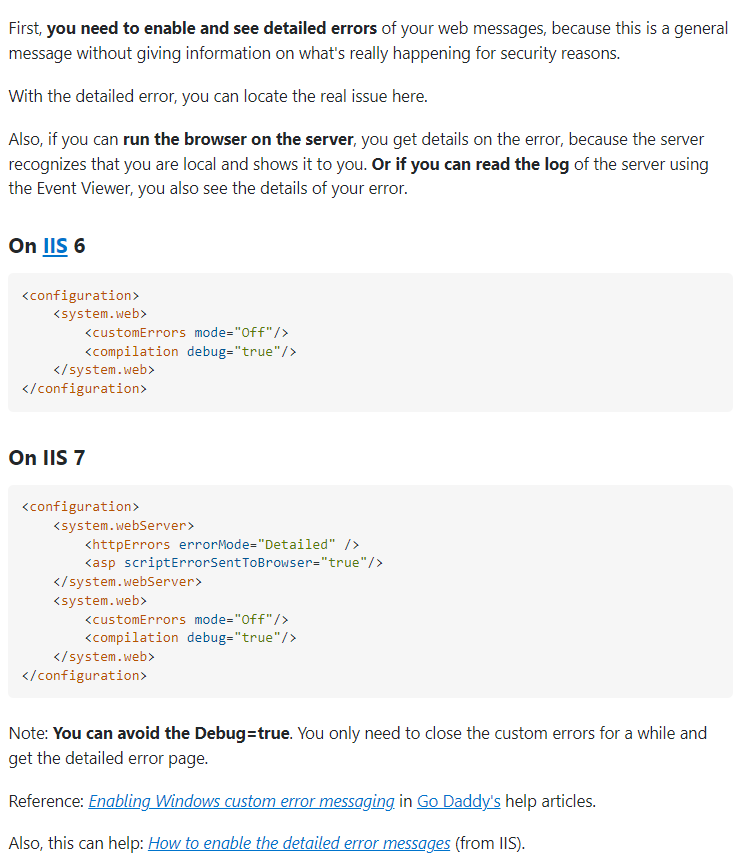
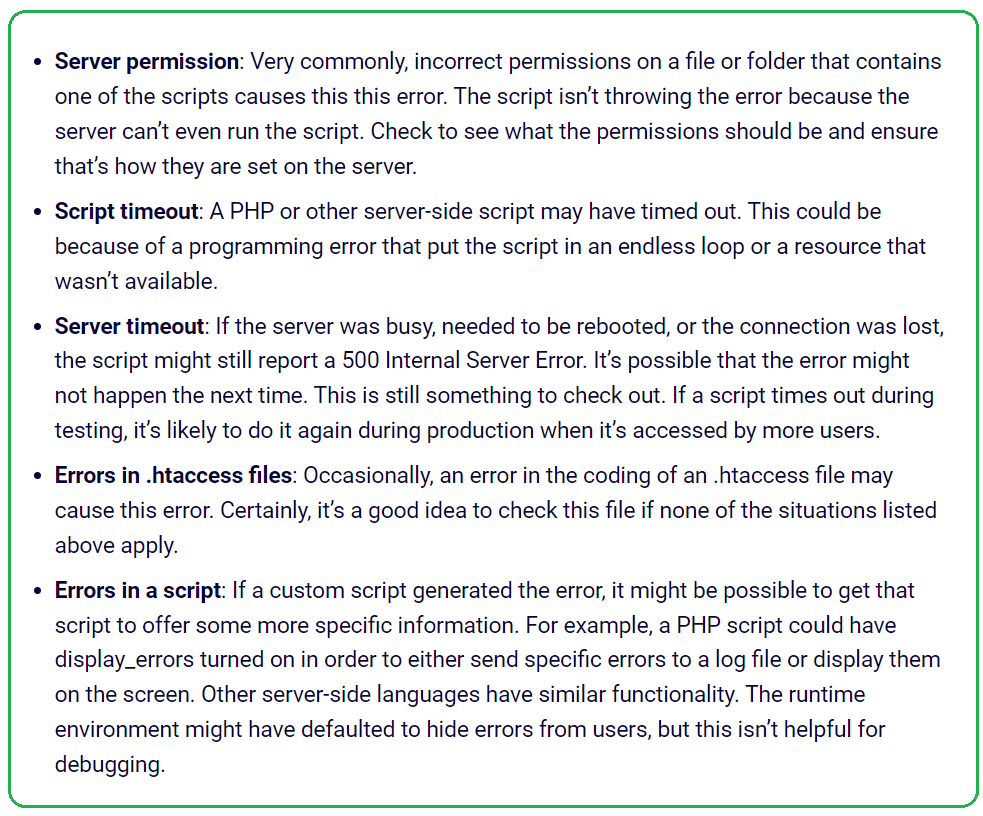
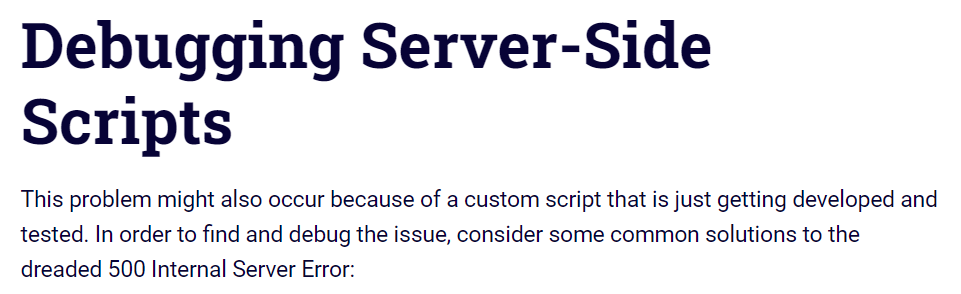
### Check Your Server Logs



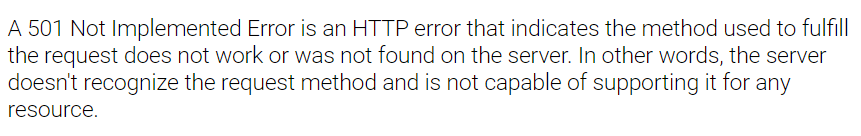


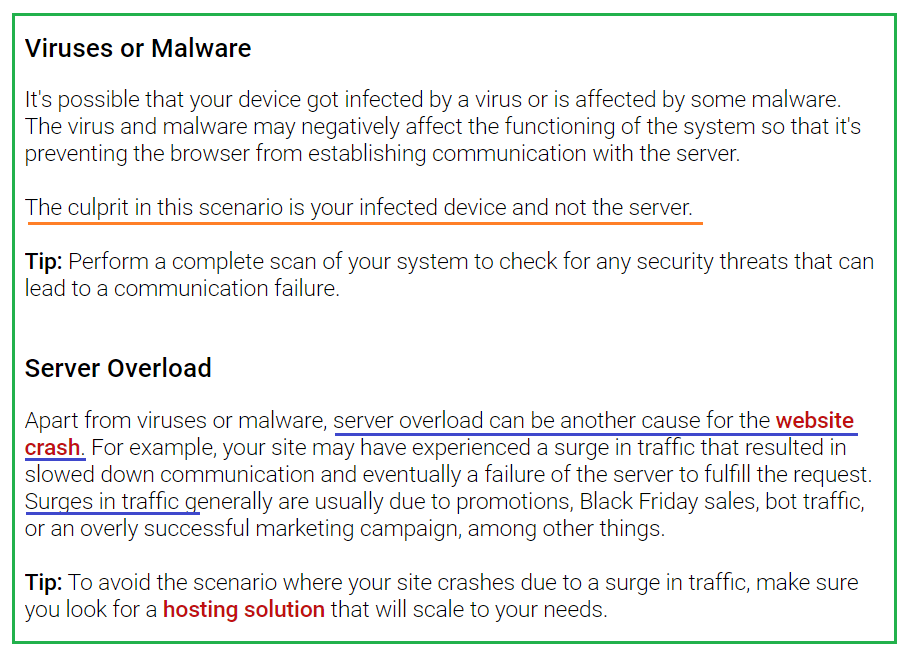






# **501 Not Implemented Error**

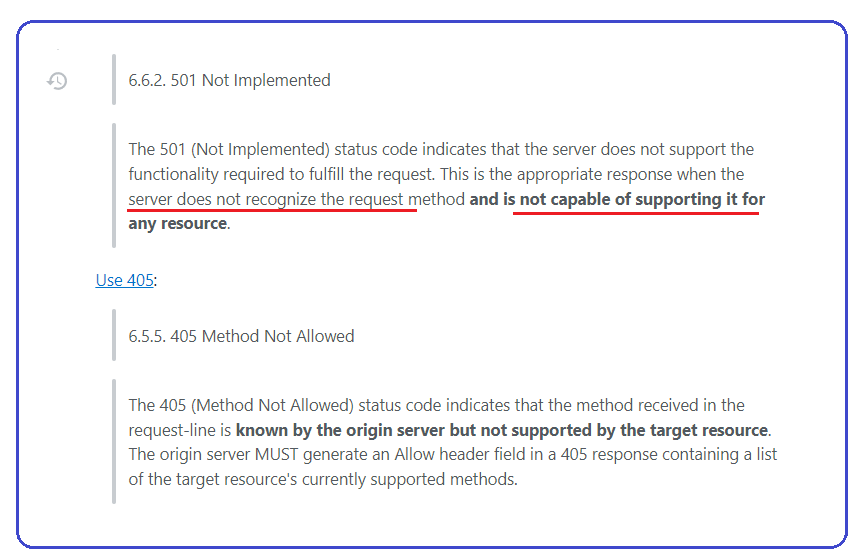


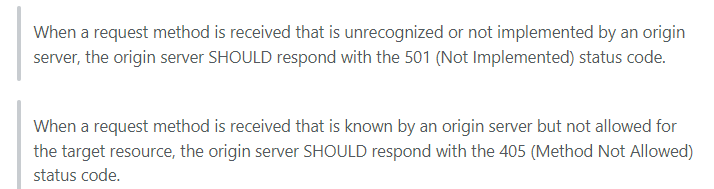


Not Implemented error is not a case of the content having not been implemented, but rather the **server not having the functionality to fulfill your request for that content**. That is what has not been implemented.

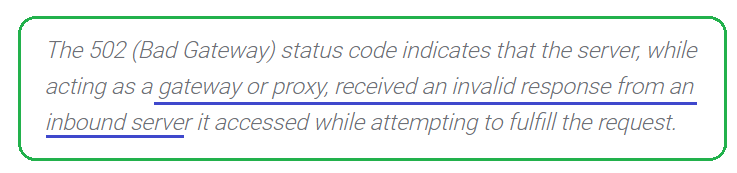
Or, in probably more common cases, the functionality that is not implemented is “being online.” That’s right: the HTTP 501 error is often given to users when the server hosting the website is completely offline and unavailable. The hosting service will return a 501 error because the server is legitimately unable to perform the requested function because, for one reason or another, the whole thing has crashed. In many cases, a crashed server will return a [500 error](https://www.elegantthemes.com/blog/tips-tricks/how-to-fix-the-500-internal-server-error-on-your-wordpress-website), but not always

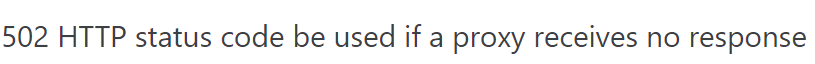
Keep in mind that anytime your server is down (or part of the functionality is offline), it can have a major effect on your search engine rankings. Google has a little grace on this front, meaning that if your server returns a 501 (or 500) error, it will be marked for revisitation. Generally, that’s enough to fix whatever happened. If not, however, Google may mark your website as being offline or inaccessible and de-index it.

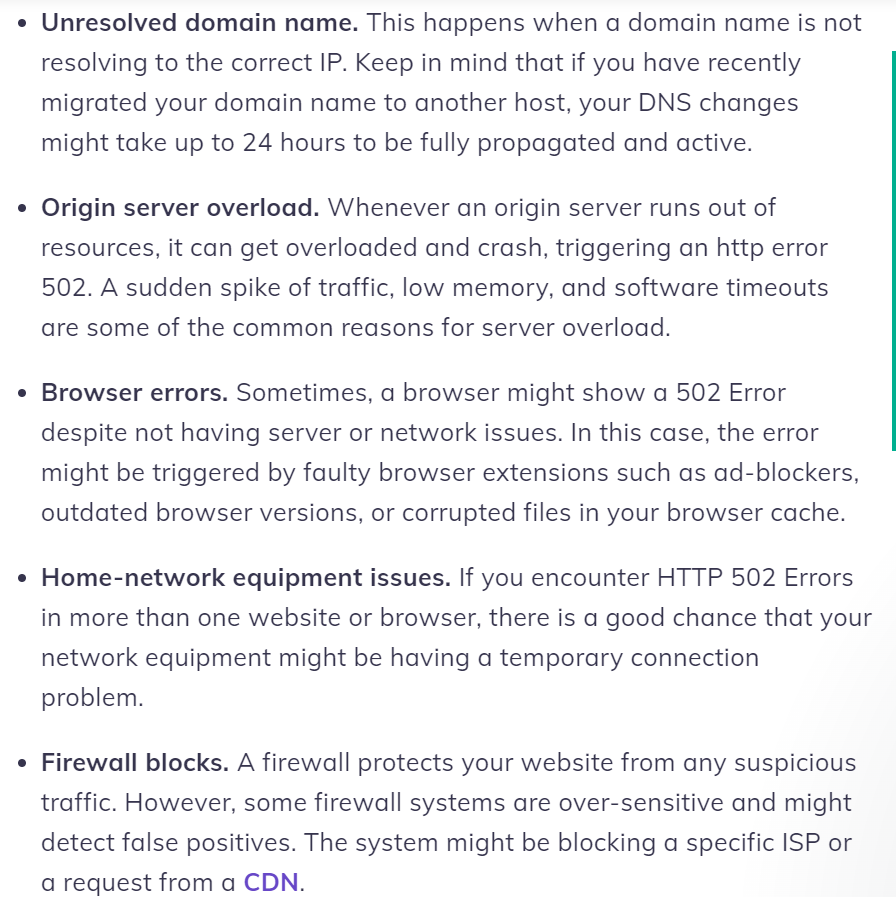
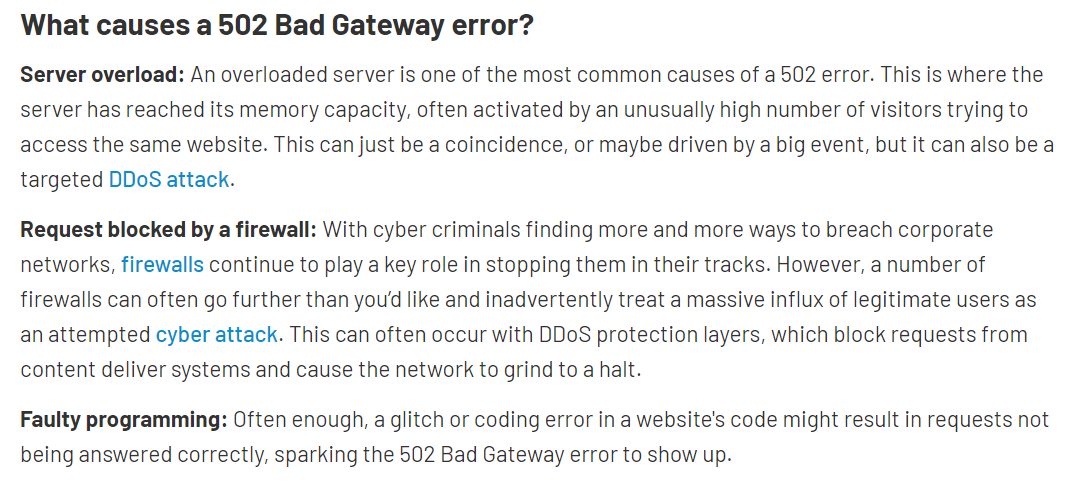




# **502 Bad Gateway Error**







# **503 Service Unavailable**

Specifically, the 503 Services Unavailable error indicates that everything is good with the request of the client, but the site's server is not available at the moment.



# **505 HTTP Version Not Supported**