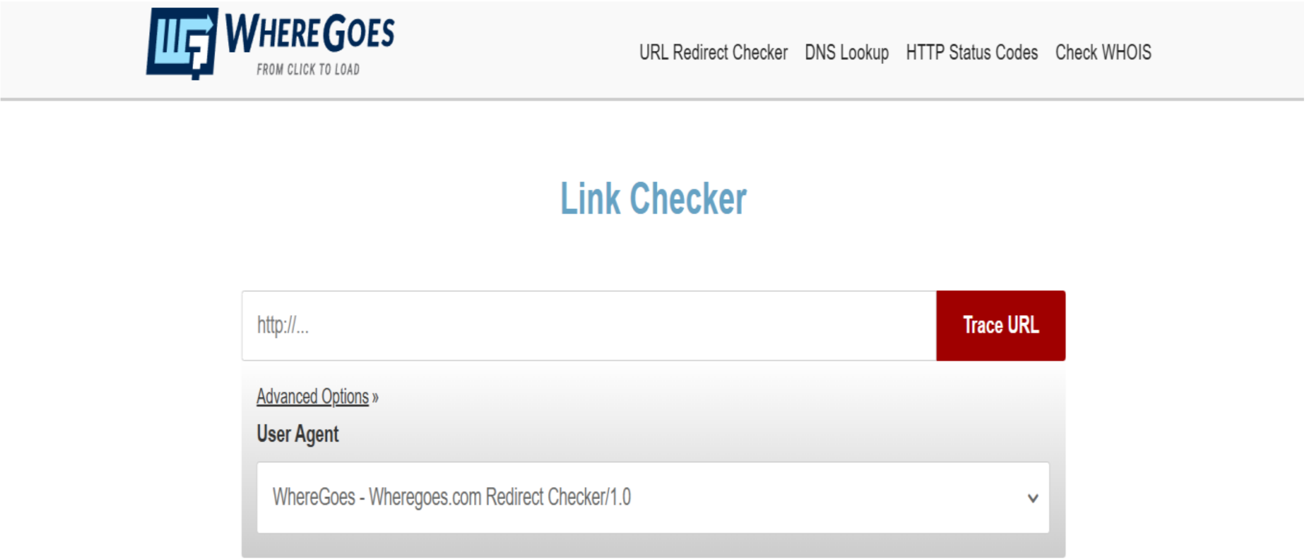
WhereGoes.com

- WhereGoes.com is an online tool designed to trace and debug URL redirects.
- Launched in 2008 by a Toronto developer for affiliate/redirect tracing, since updated and used by SEO/advertisement professionals; mentioned by reputable industry sources like Search Engine Journal and AHREFS.

What does it do?

- **URL Redirect Checker** – It follows any URL (including shortened links or affiliate redirects) step-by-step, revealing the full redirection path, HTTP status codes (like 301, 302), and intermediate hops.
- **Why it's useful** – Great for SEO experts ensuring backlinks aren't lost in redirect chains, affiliate marketers checking that links point correctly, or developers troubleshooting redirect loops and HTTP errors.
- **Bonus tools** – The site also includes utilities like DNS lookups, WHOIS checks, HTTP status code explanations, and bot behaviour info.



Link Checker

User Agent: WhereGoes - Wheregoes.com Redirect Checker/1.0

Now why should we use this site?

1. Uncover URL Redirect Chains

- It shows each step in a redirect (like 301, 302, 307 redirects).
- Useful when a link seems to change multiple times before reaching its destination.
- Great for spotting infinite loops or broken redirections.

2. Check Shortened or Affiliate Links

- Expands links from services like bit.ly, tinyurl, or tracking links used in emails or affiliate programs.
- Lets you see the final destination URL safely, before clicking.

3. Troubleshoot SEO Issues

- Redirection affects PageRank, crawling, and indexing.
- Detect unnecessary redirect hops that slow down site performance.
- Helps verify that 301 (permanent) or 302 (temporary) redirects are properly implemented.

4. Diagnose HTTP Status Codes

- You can identify if a URL returns 404 (Not Found), 403 (Forbidden), or 500 (Server Error).
- Helps find out where in the redirect path the problem lies.

5. Analyze Tracking or Marketing Campaigns

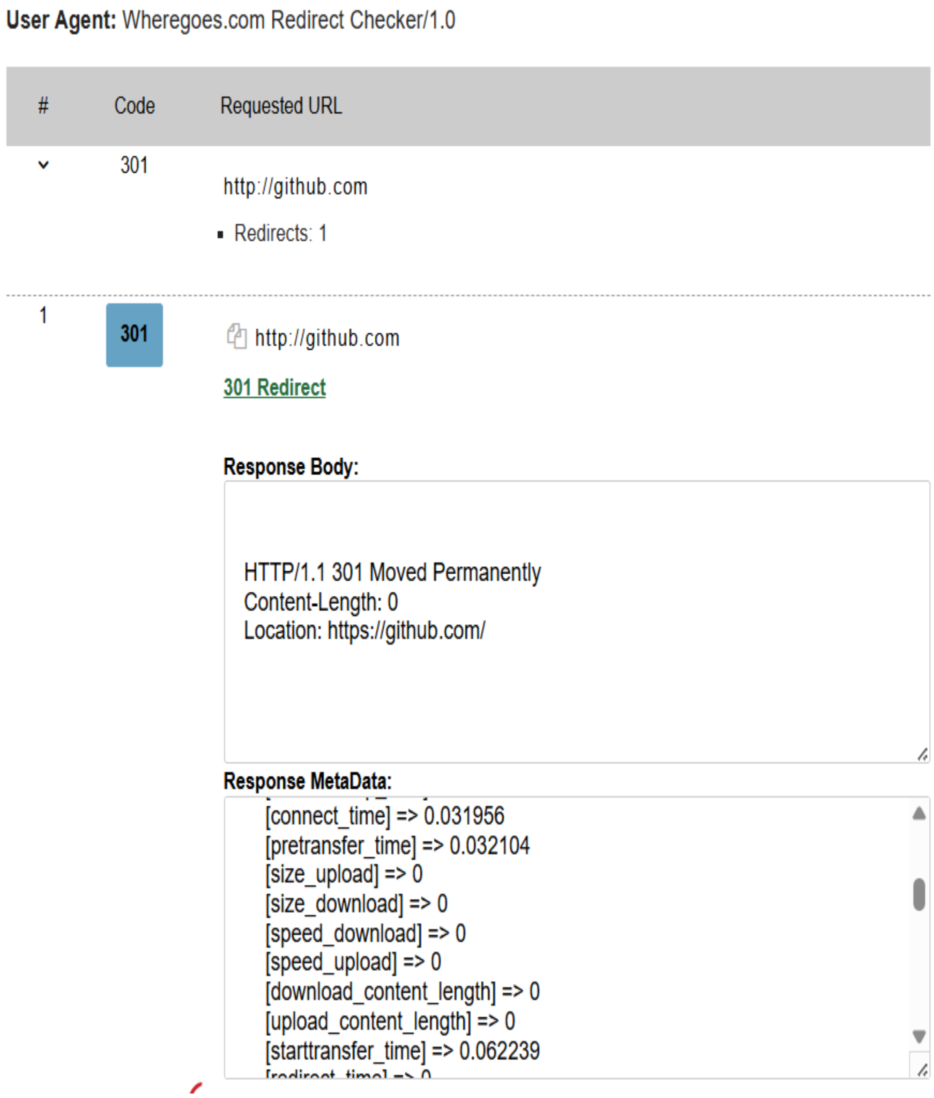
- Marketers often use redirect links in email or ad campaigns. This tool helps you:
  - Verify that UTM parameters or tracking codes are intact.
  - Confirm that no analytics data is lost during redirects.

6. Security & Transparency

- Safely inspect suspicious or unknown links without clicking them.
- Useful for identifying malicious redirects or phishing attempts.

Let me give an example:

- You clicked on a Bit.ly link but want to know where it really goes → Paste into WhereGoes.
- You changed page URLs and want to ensure proper redirection to new URLs → Use WhereGoes to confirm 301 status.
- You're doing SEO for a website and want to check redirect behavior from HTTP → HTTPS → final URL.
- You're an affiliate marketer checking if your unique tracking ID remains in the final redirected URL.



User Agent: Wheregoes.com Redirect Checker/1.0

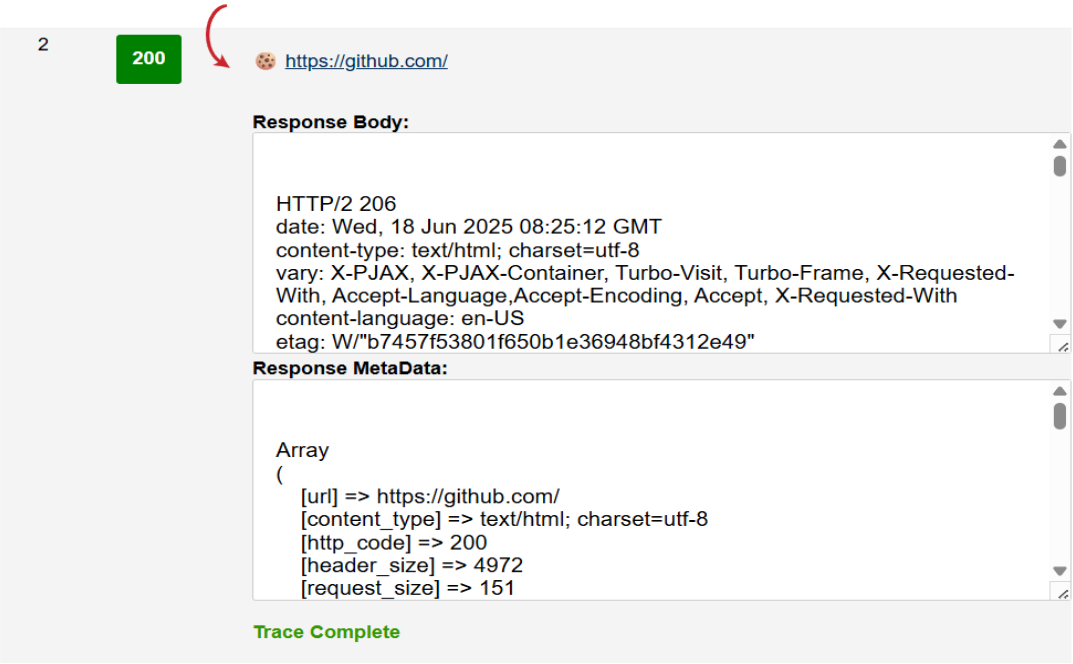
#	Code	Requested URL
˅	301	http://github.com ▪ Redirects: 1
1	301	http://github.com 301 Redirect

Response Body:

```
HTTP/1.1 301 Moved Permanently
Content-Length: 0
Location: https://github.com/
```

Response MetaData:

```
    [connect_time] => 0.031956
    [pretransfer_time] => 0.032104
    [size_upload] => 0
    [size_download] => 0
    [speed_download] => 0
    [speed_upload] => 0
    [download_content_length] => 0
    [upload_content_length] => 0
    [starttransfer_time] => 0.062239
    [redirect_time] => 0
```



2 | 200 | https://github.com/

Response Body:

```
HTTP/2 206
date: Wed, 18 Jun 2025 08:25:12 GMT
content-type: text/html; charset=utf-8
vary: X-PJAX, X-PJAX-Container, Turbo-Visit, Turbo-Frame, X-Requested-
With, Accept-Language,Accept-Encoding, Accept, X-Requested-With
content-language: en-US
etag: W/"b7457f53801f650b1e36948bf4312e49"
```

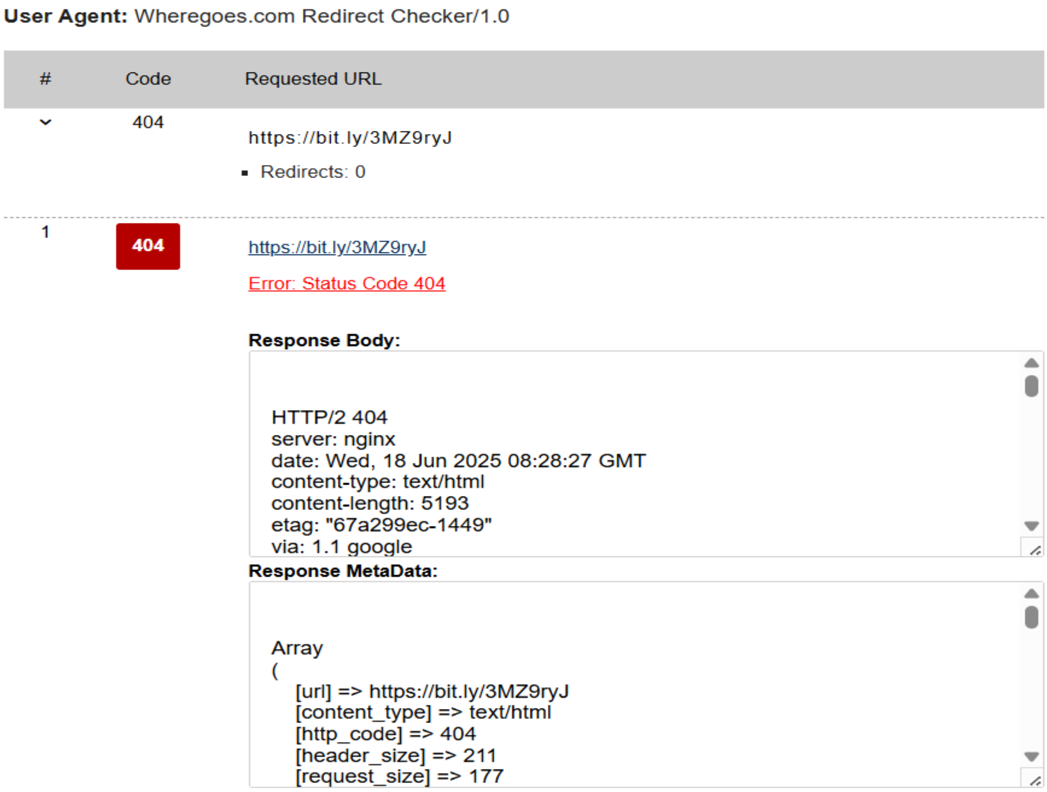
Response MetaData:

```
Array
(
    [url] => https://github.com/
    [content_type] => text/html; charset=utf-8
    [http_code] => 200
    [header_size] => 4972
    [request_size] => 151
```

Trace Complete

Here the trace is completed.

Example 2:



User Agent: Wheregoes.com Redirect Checker/1.0

#	Code	Requested URL
˅	404	https://bit.ly/3MZ9ryJ ▪ Redirects: 0
1	404	https://bit.ly/3MZ9ryJ Error: Status Code 404

Response Body:

```
HTTP/2 404
server: nginx
date: Wed, 18 Jun 2025 08:28:27 GMT
content-type: text/html
content-length: 5193
etag: "67a299ec-1449"
via: 1.1 google
```

Response MetaData:

```
Array
(
    [url] => https://bit.ly/3MZ9ryJ
    [content_type] => text/html
    [http_code] => 404
    [header_size] => 211
    [request_size] => 177
```

This url is created using a Bitly link.

Now to know the meaning of these codes we need to follow the status code sheet.

HTTP Status Codes Cheat Sheet

Table of Contents

When you attempt to access a URL, you will receive an HTTP (Hypertext Transfer Protocol) status code in response. The status code is issued by the server to the client that made the request. Below is a quick cheat sheet of all the HTTP status codes with links to more information about the status code.

Types of HTTP status code responses

The first digit identifies the class of the the type of status code response. If there is no HTTP status code response, then there will not be a status code.

1XX Informational Responses

The 1xx (Informational) class of status code indicates an interim response for communicating connection status or request progress prior to completing the requested action and sending a final response. 1xx responses are terminated by the first empty line after the status-line (the empty line signaling the end of the header section). Since HTTP/1.0 did not define any 1xx status codes, a server MUST NOT send a 1xx response to an HTTP/1.0 client.

A client MUST be able to parse one or more 1xx responses received prior to a final response, even if the client does not expect one. A user agent MAY ignore unexpected 1xx responses.

A proxy MUST forward 1xx responses unless the proxy itself requested the generation of the 1xx response. For example, if a proxy adds an “Expect: 100-continue” field when it forwards a request, then it need not forward the corresponding 100 (Continue) response(s).

- 100 Continue
- 101 Switching Protocol
- 102 Processing (WebDAV)
- 103 Early Hints

2XX Successful Responses

The 2xx Successful class of status code indicates that the client’s request was successfully received, understood, and accepted.

- 200 OK
- 201 Created
- 202 Accepted
- 203 Non-Authoritative Information
- 204 No Content
- 205 Reset Content
- 206 Partial Content
- 207 Multi-Status (WebDAV)
- 208 Already Reported (WebDAV)
- 226 IM Used (HTTP Delta encoding)

3XX Redirects

- 300 Multiple Choice
- 301 Moved Permanently
- 302 Found
- 303 See Other
- 304 Not Modified
- 305 Use Proxy
- 306 Unused
- 307 Temporary Redirect
- 308 Permanent Redirect

4XX Client Errors

- 400 Bad Request
- 401 Unauthorized
- 402 Payment Required
- 403 Forbidden
- 404 Not Found
- 405 Method Not Allowed
- 406 Not Acceptable
- 407 Proxy Authentication Required
- 408 Request Timeout

- 409 Conflict
- 410 Gone
- 411 Length Required
- 412 Precondition Failed
- 413 Payload Too Large
- 414 URI Too Long
- 415 Unsupported Media Type
- 416 Range Not Satisfiable
- 417 Expectation Failed
- 418 I’m a teapot
- 421 Misdirected Request
- 422 Unprocessable Entity (WebDAV)
- 423 Locked (WebDAV)
- 424 Failed Dependency (WebDAV)
- 425 Too Early
- 426 Upgrade Required
- 428 Precondition Required
- 429 Too Many Requests
- 431 Request Header Fields Too Large
- 451 Unavailable For Legal Reasons
- 499 Client Closed Request

5XX Server Errors

- 500 Internal Server Error
- 501 Not Implemented
- 502 Bad Gateway
- 503 Service Unavailable
- 504 Gateway Timeout
- 505 HTTP Version Not Supported
- 506 Variant Also Negotiates
- 507 Insufficient Storage (WebDAV)
- 508 Loop Detected (WebDAV)
- 510 Not Extended
- 511 Network Authentication Required

We got code as 404 error now what does that mean?

What Does a 404 Error Mean?

Ah, you've clicked on a link or used the WhereGoes redirect checker and received an http error 404 not found. Don't worry this is quite a common issue. This is the error provided when the web browser cannot find the requested website address. The "404" is the designated HTTP status code for the "Not Found" error. The meaning can differ slightly due to context, but basically, the 404 means the web server cannot find the requested resource whether it is a page or file.

What is a broken link?

Broken links, also known as *dead links*, are links on a website that point to a page or file that no longer exists resulting in a 404 error code. Dead links are often a result of linkrot. Linkrot or "link rot" is the web document half-life where over time URLs no longer point to their original resource due to them being relocated or removed entirely.

What Causes a 404 Error?

404 errors can happen for various reasons:

1. The page or file was deleted and no longer exists. Broken links are often the result of linkrot.
2. The web page's address or filename changed and the owner of the website did not create a 301 redirect to forward you to the new address.
3. If the URL was typed in manually, you have misspelled the real address and thus the destination doesn't exist.
4. If a website is using WordPress, there may be an issue with the permalinks (pretty links) in the database.
5. If a link redirects to another link, but the destination URL address no longer exists.
6. The domain name system (DNS) cannot resolve to an IP address.
7. The domain name itself doesn't exist any longer.

How to Find Broken Links on Website

It is best to find your broken links and 404 images before a search engine does. This will help ensure that your search rankings do not suffer as a result.

Scan Site for Broken Links

Scan your site with tools like SEMRush or Ahrefs periodically. If you are actively making changes to the website, then it is inevitable that you will break links at some point. These scheduled SEO audit scans will inform you when you have unknowingly introduced a dead link or referencing external websites that have recently moved or taken down their content resulting in a broken link.

Monitor 404 Errors

It is important to log all of your 404 errors that occur on your site. These logs will help you identify pages that users or bots are accessing but cannot find.

Here's some great tools to monitor your 404 errors from your WordPress admin panel.

- WP Security Safe
- Redirection
- Rank Math SEO

Of course you can always review Google's free tools.

- Google Search Console
- Google Analytics

How to Fix 404 Errors?

Find New URL Location

If you were linking to an article on a website that now 404s or clicked on a broken link from someone else's website, then don't fret. The website owner probably changed the URL location and forgot to add in a 301 redirect.

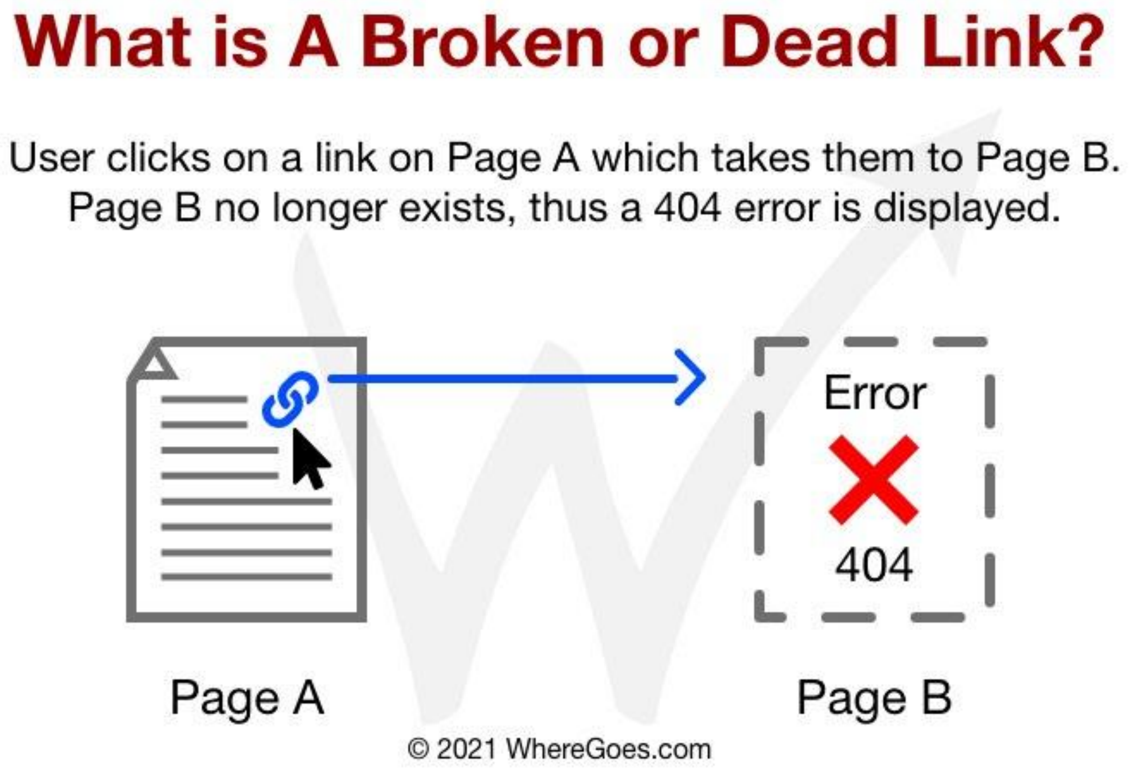
1. Simply go to google.com and search for "site:website.com" and then title of the article. It would look like this: "site:website.com {title of article}".
2. If you do not find the article initially, try the same method again but this time, use other keywords related to the topic.

How to Fix Broken Links

If you are the website owner, then here's what you do:

1. Find the location of the old content. You can use the method mentioned above, "site:website.com {keywords}" to find it on your own site.
2. Once you have found the location of the new content, you need to create a 301 redirect from the old location to the new location. Doing that will ensure that the next time Google crawls the old URL, it will get forward to the new URL.
3. You need to update all the content that is linking to the old webpage address to link to the new URL destination. This is important as it will

reduce additional work on the server to get your visitors or Google to the end destination.



What is A Broken or Dead Link?

User clicks on a link on Page A which takes them to Page B. Page B no longer exists, thus a 404 error is displayed.

© 2021 WhereGoes.com