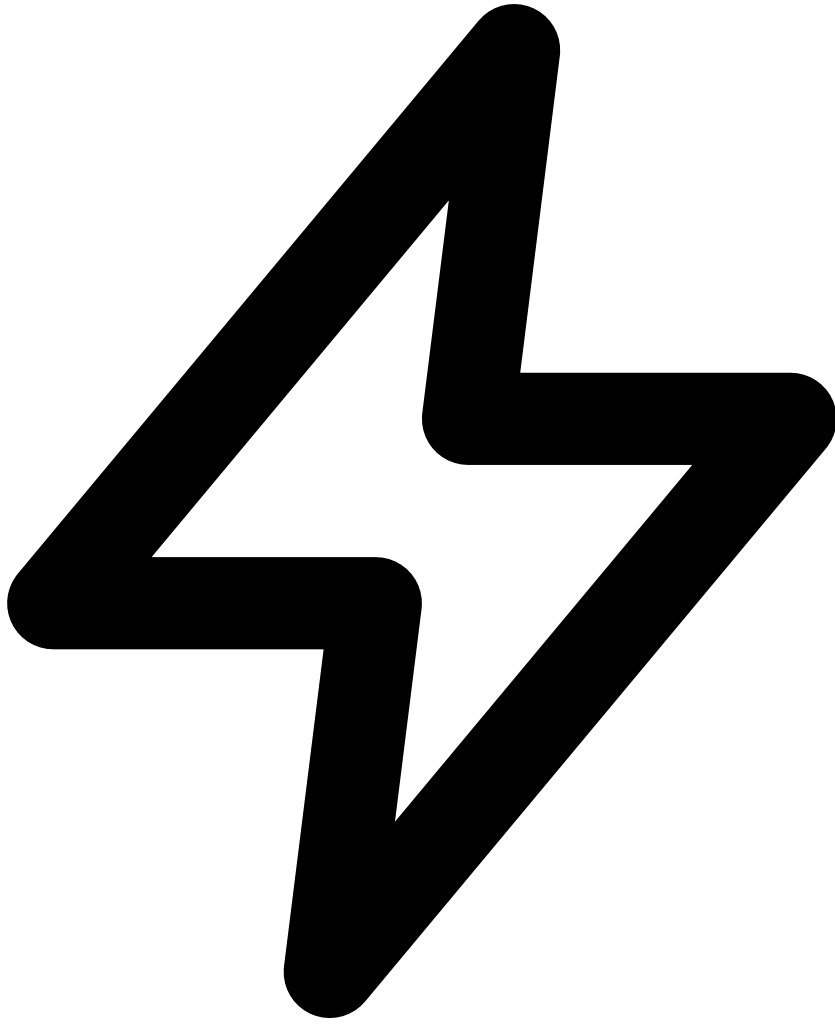
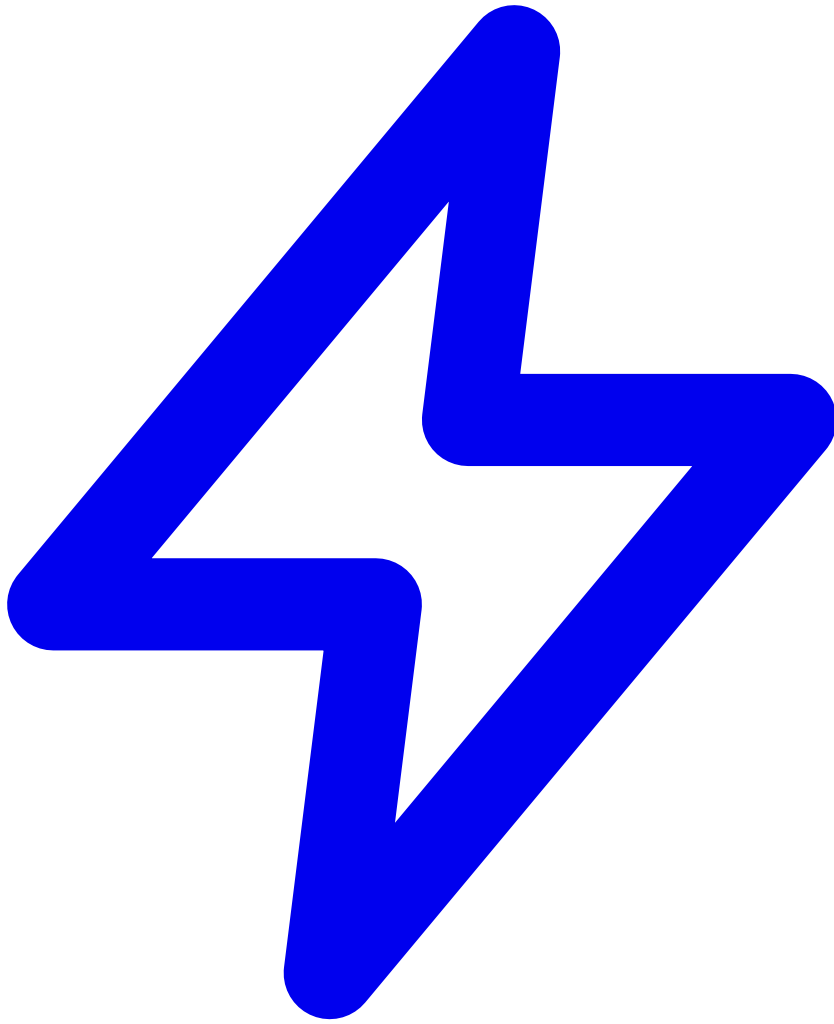






[Autonomiq](#)

[TextDeveloperConvertersGeneratorsBlog](#)

Search tools ⌘K



[TextDeveloperConvertersGeneratorsBlogAboutContact](#)

Search for a tool...

[Home](#) / [Blog](#) / HTTP Status Codes Explained: What 200, 301, 404 and 500 Really Mean

## HTTP Status Codes Explained: What 200, 301, 404 and 500 Really Mean

By the [AutonomiqTech Editorial Team](#) · Reviewed against our [editorial standards](#) · 7 min read ·  
Last reviewed 2026

## HTTP Status Codes Explained: What 200, 301, 404 and 500 Really Mean

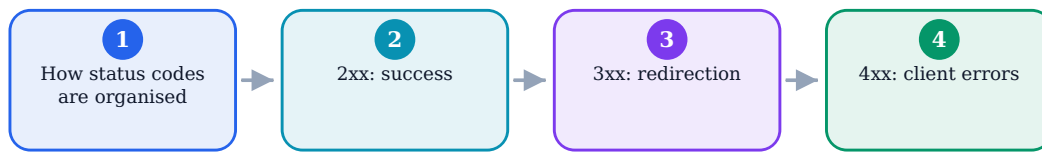


Figure: HTTP Status Codes Explained: What 200, 301, 404 and 500 Really Mean

Every time your browser loads a page, the server replies with a three-digit **HTTP status code** that says how the request went. Most of the time you never see it — but when you hit a “404 Not Found” or a “500 Server Error”, the code suddenly matters.

This practical reference explains how status codes are grouped and what the most important ones mean, so you can read them confidently whether you're building sites or just curious.

Advertisement

### On this page

1. [How status codes are organised](#)
2. [2xx: success](#)
3. [3xx: redirection](#)
4. [4xx: client errors](#)
5. [5xx: server errors](#)
6. [Using status codes well](#)
7. [Status codes you'll meet most](#)
8. [What each category tells you](#)
9. [Using status codes well as a developer](#)
10. [A few codes worth recognising beyond the basics](#)

## How status codes are organised

Every HTTP response includes a three-digit code, and the **first digit** tells you the general category: **1xx** informational, **2xx** success, **3xx** redirection, **4xx** client errors, and **5xx** server errors. Learn the families first, and individual codes become much easier to interpret.

### 2xx: success

The **2xx** family means the request succeeded. **200 OK** is the one you see constantly — the page or data loaded fine. **201 Created** often appears after successfully creating something (like submitting a form that makes a new record). If you're seeing 2xx codes, things are working.

## 3xx: redirection

The **3xx** family means the resource is somewhere else. The two most important are **301 Moved Permanently** and **302 Found** (temporary). Use a **301** when a page has permanently moved — it tells browsers and search engines to update to the new URL, preserving link value. Use a **302** for temporary redirects. Getting this distinction right matters a lot for SEO.

## 4xx: client errors

The **4xx** family means something was wrong with the request. **404 Not Found** — the page doesn't exist — is the most famous. **403 Forbidden** means you're not allowed, and **401 Unauthorized** means you need to authenticate. These point to problems on the requesting side (a bad link, missing login, or wrong permissions).

## 5xx: server errors

The **5xx** family means the server failed to fulfil a valid request. **500 Internal Server Error** is a generic 'something broke on our end'. **503 Service Unavailable** often means the server is overloaded or down for maintenance. Unlike 4xx codes, these signal that the fix is on the server side, not the client's.

## Using status codes well

For site owners, status codes are a diagnostic goldmine. Regular 404s reveal broken links to fix; proper 301s keep moved pages from losing traffic; and clusters of 5xx errors flag server problems needing urgent attention. Reading them fluently turns cryptic errors into clear, actionable information.

## Status codes you'll meet most

You don't need to memorise every code, but a handful appear constantly. Knowing these covers most situations:

| Code | Meaning               | In plain terms                |
|------|-----------------------|-------------------------------|
| 200  | OK                    | It worked                     |
| 301  | Moved Permanently     | This has a new address        |
| 404  | Not Found             | That page doesn't exist       |
| 403  | Forbidden             | You're not allowed in         |
| 500  | Internal Server Error | Something broke on the server |

Remembering the first digit — 2 success, 3 redirect, 4 your error, 5 server error — lets you interpret almost any code at a glance.

## What each category tells you

The leading digit groups codes by meaning, which is the fastest way to understand one:

- **2xx**: the request succeeded.
- **3xx**: further action (usually a redirect) is needed.
- **4xx**: the client's request had a problem — bad URL, no permission, etc.
- **5xx**: the server failed to fulfil a valid request.

## Using status codes well as a developer

Status codes aren't just diagnostic messages to read; when you build services, using them correctly is an important part of communicating clearly with the clients and browsers that consume your API or site. Returning the right code lets other software respond intelligently — a well-behaved client can retry on certain server errors, follow a redirect automatically, or show a helpful message on a not-found rather than treating every outcome the same. Misusing codes, such as returning a 200 'success' when something actually failed, is a common and damaging mistake, because it hides errors from anything relying on the status to know what happened, leading to bugs that are hard to trace. Similarly, distinguishing between a client error and a server error matters: a 4xx tells the caller to fix their request, while a 5xx tells them the problem is on your side, and confusing the two sends people looking in the wrong place. Using redirects with the appropriate permanent or temporary code helps search engines and browsers handle moved content correctly. Providing a clear, structured error response alongside the status code — explaining what went wrong — makes your service far easier to work with. For consumers of APIs and websites, the same knowledge is valuable in reverse: reading the status code first tells you immediately whether to look at your own request, wait and retry, or report a server-side fault. Treating status codes as a precise, shared language rather than an afterthought is what makes web systems predictable, debuggable and pleasant to build against.

## A few codes worth recognising beyond the basics

While the common codes cover most situations, a handful of less famous ones are worth recognising because they explain confusing situations you will eventually meet. A 429 tells you that you have made too many requests in too short a time and are being rate-limited, which is the correct response to see if you hammer an API and the polite fix is simply to slow down and try again after a pause. A 401 versus a 403 is a distinction that trips people up: a 401 means you are not authenticated, essentially that the service does not know who you are and you need to log in or supply valid credentials, whereas a 403 means it knows exactly who you are but you are not permitted to do this particular thing. A 302 signals a temporary redirect, telling browsers and search engines that a page has moved for now but the original address should still be treated as the real one, in contrast to the 301 that announces a permanent move. A 503 indicates the server is temporarily unavailable, often because it is overloaded or down for maintenance, which suggests waiting and retrying rather than assuming your request was wrong. Recognising these extra codes means that when something behaves oddly, you can often diagnose it instantly from the status alone — knowing whether to authenticate, wait, back off, or report a fault — instead of guessing, which is the whole practical value of understanding status codes as a language rather than a set of opaque numbers.

## Printable checklist

Print this page or save the PDF to keep these steps handy.

- ☐ How status codes are organised
- ☐ 2xx: success
- ☐ 3xx: redirection
- ☐ 4xx: client errors
- ☐ 5xx: server errors
- ☐ Using status codes well
- ☐ Status codes you'll meet most
- ☐ What each category tells you

## Summary

HTTP status codes are three-digit numbers a server returns to describe the outcome of a request. They're grouped by first digit: 2xx success, 3xx redirection, 4xx client errors, 5xx server errors. Knowing the key ones — 200, 301, 302, 404, 500 and a few others — helps you diagnose problems, set up redirects correctly, and keep a site healthy.

## Key Takeaways

- Status codes are grouped by their first digit into five families.
- 2xx means success; 200 OK is the everyday 'it worked'.
- 3xx means redirection; 301 is permanent, 302 is temporary.
- 4xx means a client-side problem; 404 Not Found is the most famous.
- 5xx means a server-side error; 500 is a generic server failure.

## Frequently Asked Questions

### What does a 404 error mean?

404 Not Found means the server couldn't find the requested page. It's usually caused by a broken link, a mistyped URL, or a page that was moved or deleted without a redirect.

### What's the difference between a 301 and a 302 redirect?

A 301 is a permanent redirect — use it when a page has moved for good, since it passes on link value. A 302 is temporary, telling browsers the original URL will return later.

### Is a 500 error my fault or the website's?

A 500 Internal Server Error is a server-side problem, so it's on the website's end, not yours. Refreshing later sometimes helps, but the fix has to come from the site's operators.

## Related Guides

- [What Is an API?](#)
- [What Is a URL Slug?](#)
- [What Is a JSON Web Token?](#)
- [Regular Expressions Basics](#)

## Related articles

- [What Is Base64 Actually Used For? Real-World Examples](#)
- [What Is a URL Slug? Best Practices for Clean, Readable URLs](#)
- [UTF-8 vs ASCII: Character Encoding Explained for Beginners](#)

---

Source: <https://autonomiqtech.com/http-status-codes-explained.html> — Educational content. Verify time-sensitive details against current sources.