

# HTTP STATUS CODES QUICK REFERENCE

Status codes, headers, and common response patterns

## Informational 1xx

### 1xx Codes

- 100** Continue — server received headers, client should send body
- 101** Switching Protocols — upgrading to WebSocket or HTTP/2
- 102** Processing — server received request, still working (WebDAV)
- 103** Early Hints — preload resources before final response

### Usage Note

```
# 100 Continue: client sends Expect header, waits for 100
curl -H "Expect: 100-continue" -d @large.json URL
# 101: upgrade to WebSocket
Connection: Upgrade / Upgrade: websocket
```

## Success 2xx

### 2xx Codes

- 200** OK — standard success response
- 201** Created — resource successfully created (POST/PUT)
- 202** Accepted — request received, processing async
- 203** Non-Authoritative Info — transformed by proxy
- 204** No Content — success with no response body (DELETE)
- 205** Reset Content — success, client should reset form
- 206** Partial Content — range request fulfilled
- 207** Multi-Status — multiple status codes (WebDAV)

### REST API Usage

- GET → 200** Return resource with body
- POST → 201** Resource created, include Location header
- PUT → 200/204** Updated resource (with/without body)
- DELETE → 204** Deleted, no body returned
- PATCH → 200** Partial update, return modified resource

## Redirection 3xx

### 3xx Codes

- 300** Multiple Choices — multiple representations available
- 301** Moved Permanently — resource moved, update bookmarks
- 302** Found — temporary redirect (often misused as 303)
- 303** See Other — redirect with GET after POST
- 304** Not Modified — use cached version (ETag/If-Modified)
- 307** Temporary Redirect — same method, temporary location
- 308** Permanent Redirect — same method, permanent location

### Redirect Behavior

- 301/308** Permanent — search engines update index
- 302/307** Temporary — original URL stays canonical
- 301/302** May change method to GET on redirect
- 307/308** Must preserve original HTTP method

## Client Error 4xx

### Common Client Errors

- 400** Bad Request — malformed syntax or invalid parameters
- 401** Unauthorized — authentication required or failed
- 403** Forbidden — authenticated but not permitted
- 404** Not Found — resource does not exist
- 405** Method Not Allowed — HTTP method not supported
- 406** Not Acceptable — can't satisfy Accept header
- 408** Request Timeout — client too slow to send request
- 409** Conflict — request conflicts with current state

### More Client Errors

- 410** Gone — resource permanently deleted (not just missing)
- 411** Length Required — Content-Length header missing
- 412** Precondition Failed — If-Match/If-Unmodified failed
- 413** Content Too Large — request body exceeds limit
- 414** URI Too Long — URL exceeds server limit
- 415** Unsupported Media Type — Content-Type not accepted
- 422** Unprocessable Content — valid syntax, semantic errors
- 429** Too Many Requests — rate limit exceeded

## Server Error 5xx

### 5xx Codes

- 500** Internal Server Error — unhandled exception on server
- 501** Not Implemented — server doesn't support the method
- 502** Bad Gateway — upstream server sent invalid response
- 503** Service Unavailable — overloaded or in maintenance
- 504** Gateway Timeout — upstream server didn't respond in time
- 505** HTTP Version Not Supported — version not handled
- 507** Insufficient Storage — server can't store request (WebDAV)
- 511** Network Auth Required — captive portal login needed

### Retry Strategy

- 500** Retry with backoff; may be transient
- 502/504** Retry — upstream issue may resolve
- 503** Check Retry-After header before retrying
- 501/505** Do not retry — fix client request

## Common Codes

### Most-Used Codes (at a glance)

- 200** OK — everything worked
- 201** Created — new resource made
- 204** No Content — success, empty body
- 301** Moved Permanently — update URL
- 304** Not Modified — use cache
- 400** Bad Request — fix your request
- 401** Unauthorized — log in first
- 403** Forbidden — insufficient permissions
- 404** Not Found — wrong URL or deleted
- 422** Unprocessable — validation errors
- 429** Too Many Requests — slow down
- 500** Server Error — not your fault
- 502** Bad Gateway — proxy/upstream failure

- 503** Unavailable — try again later

## Headers Reference

### Request Headers

- Accept** Desired response media types (e.g. application/json)
- Authorization** Credentials (Bearer token, Basic base64)
- Content-Type** Media type of request body
- If-None-Match** Conditional: ETag for cache validation
- If-Modified-Since** Conditional: date for cache validation
- Cache-Control** Caching directives (no-cache, max-age)
- User-Agent** Client identification string

### Response Headers

- Content-Type** Media type of response body
- Location** Redirect target or created resource URL
- ETag** Entity tag for cache validation
- Cache-Control** Caching directives (max-age, no-store)
- Retry-After** Wait time before retrying (429/503)
- WWW-Authenticate** Auth scheme required (sent with 401)
- Set-Cookie** Set cookie on client

## Common Patterns

### Caching Flow

```
# First request - server returns ETag
GET /api/data → 200, ETag: "abc123"
# Subsequent request - conditional
GET /api/data, If-None-Match: "abc123"
→ 304 Not Modified (use cache)
```

### Auth Flow

```
# Unauthenticated request
GET /api/secret → 401, WWW-Authenticate: Bearer
# With token
GET /api/secret, Authorization: Bearer <token>
→ 200 OK
```

### Rate Limiting

```
# Rate limited response
429 Too Many Requests
Retry-After: 60
X-RateLimit-Remaining: 0
X-RateLimit-Reset: 1700000000
```

### Content Negotiation

```
# Client prefers JSON, accepts XML
Accept: application/json, application/xml;q=0.9
# Server can't satisfy → 406 Not Acceptable
# Server returns best match → 200 + Content-Type
```