

CODICI DI STATO HTTP RIFERIMENTO RAPIDO

Codici di stato, header e pattern di risposta comuni

Informativi 1xx

Codici 1xx

- 100** Continue — il server ha ricevuto gli header, il client può inviare il corpo
- 101** Switching Protocols — upgrade a WebSocket o HTTP/2
- 102** Processing — il server ha ricevuto la richiesta, sta elaborando (WebDAV)
- 103** Early Hints — precarica risorse prima della risposta finale

Nota d'Uso

```
# 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
```

Successo 2xx

Codici 2xx

- 200** OK — risposta di successo standard
- 201** Created — risorsa creata con successo (POST/PUT)
- 202** Accepted — richiesta ricevuta, elaborazione asincrona
- 203** Non-Authoritative Info — trasformato da proxy
- 204** No Content — successo senza corpo della risposta (DELETE)
- 205** Reset Content — successo, il client deve azzerare il form
- 206** Partial Content — richiesta di intervallo soddisfatta
- 207** Multi-Status — più codici di stato (WebDAV)

Uso nelle API REST

- GET** → **200** Restituisce la risorsa con il corpo
- POST** → **201** Risorsa creata, includi l'header Location
- PUT** → **200/204** Risorsa aggiornata (con/senza corpo)
- DELETE** → **204** Eliminato, nessun corpo restituito
- PATCH** → **200** Aggiornamento parziale, restituisce la risorsa modificata

Reindirizzamento 3xx

Codici 3xx

- 300** Multiple Choices — sono disponibili più rappresentazioni
- 301** Moved Permanently — risorsa spostata, aggiorna i segnalibri
- 302** Found — reindirizzamento temporaneo (spesso usato al posto di 303)
- 303** See Other — reindirizzamento con GET dopo POST
- 304** Not Modified — usa la versione in cache (ETag/If-Modified)
- 307** Temporary Redirect — stesso metodo, posizione temporanea
- 308** Permanent Redirect — stesso metodo, posizione permanente

Comportamento dei Reindirizzamenti

- 301/308** Permanente — i motori di ricerca aggiornano l'indice
- 302/307** Temporaneo — l'URL originale rimane canonico
- 301/302** Possono cambiare il metodo in GET al reindirizzamento
- 307/308** Devono preservare il metodo HTTP originale

Errore Client 4xx

Errori Client Comuni

- 400** Bad Request — sintassi malformata o parametri non validi
- 401** Unauthorized — autenticazione richiesta o fallita
- 403** Forbidden — autenticato ma non autorizzato
- 404** Not Found — la risorsa non esiste
- 405** Method Not Allowed — metodo HTTP non supportato
- 406** Not Acceptable — impossibile soddisfare l'header Accept
- 408** Request Timeout — il client è troppo lento nell'inviare la richiesta
- 409** Conflict — la richiesta è in conflitto con lo stato corrente

Altri Errori Client

- 410** Gone — risorsa eliminata definitivamente (non solo mancante)
- 411** Length Required — header Content-Length mancante
- 412** Precondition Failed — If-Match/If-Unmodified fallito
- 413** Content Too Large — il corpo della richiesta supera il limite
- 414** URI Too Long — l'URL supera il limite del server
- 415** Unsupported Media Type — Content-Type non accettato
- 422** Unprocessable Content — sintassi valida, errori semantici
- 429** Too Many Requests — limite di frequenza superato

Errore Server 5xx

Codici 5xx

- 500** Internal Server Error — eccezione non gestita sul server
- 501** Not Implemented — il server non supporta il metodo
- 502** Bad Gateway — il server upstream ha inviato una risposta non valida
- 503** Service Unavailable — sovraccarico o in manutenzione
- 504** Gateway Timeout — il server upstream non ha risposto in tempo
- 505** HTTP Version Not Supported — versione non gestita
- 507** Insufficient Storage — il server non può archiviare la richiesta (WebDAV)
- 511** Network Auth Required — richiesto login al portale captive

Strategia di Retry

- 500** Riprova con backoff, potrebbe essere transitorio
- 502/504** Riprova — il problema upstream potrebbe risolversi
- 503** Controlla l'header Retry-After prima di riprovare
- 501/505** Non riprovare — correggere la richiesta del client

Codici Comuni

Codici più Usati (a colpo d'occhio)

- 200** OK — tutto funziona
- 201** Created — nuova risorsa creata
- 204** No Content — successo, corpo vuoto
- 301** Moved Permanently — aggiorna l'URL
- 304** Not Modified — usa la cache
- 400** Bad Request — correggi la richiesta
- 401** Unauthorized — effettua prima il login
- 403** Forbidden — permessi insufficienti

404 Not Found — URL errato o eliminato

422 Unprocessable — errori di validazione

429 Too Many Requests — rallenta

500 Server Error — non è colpa tua

502 Bad Gateway — errore proxy/upstream

503 Unavailable — riprova più tardi

Riferimento Header

Header di Richiesta

Accept Tipi di media desiderati nella risposta (es. application/json)

Authorization Credenziali (Bearer token, Basic base64)

Content-Type Tipo di media del corpo della richiesta

If-None-Match Condizionale: ETag per la validazione della cache

If-Modified-Since Condizionale: data per la validazione della cache

Cache-Control Direttive di caching (no-cache, max-age)

User-Agent Stringa di identificazione del client

Header di Risposta

Content-Type Tipo di media del corpo della risposta

Location URL di reindirizzamento o della risorsa creata

ETag Tag entità per la validazione della cache

Cache-Control Direttive di caching (max-age, no-store)

Retry-After Tempo di attesa prima di riprovare (429/503)

WWW-Authenticate Schema di autenticazione richiesto (inviato con 401)

Set-Cookie Imposta cookie sul client

Pattern Comuni

Flusso di Caching

```
# 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)
```

Flusso di Autenticazione

```
# 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
```

Negoziazione del Contenuto

```
# 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
```