

Referencia Rápida de Códigos de Estado HTTP

Códigos de estado, cabeceras y patrones de respuesta comunes

Informativos 1xx

Códigos 1xx

100	Continue — el servidor recibió las cabeceras, el cliente debe enviar el cuerpo
101	Switching Protocols — actualizando a WebSocket o HTTP/2
102	Processing — el servidor recibió la solicitud, aún procesando (WebDAV)
103	Early Hints — precargar recursos antes de la respuesta final

Nota de 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
```

Éxito 2xx

Códigos 2xx

200	OK — respuesta de éxito estándar
201	Created — recurso creado exitosamente (POST/PUT)
202	Accepted — solicitud recibida, procesando de forma asíncrona
203	Non-Authoritative Info — transformada por un proxy
204	No Content — éxito sin cuerpo de respuesta (DELETE)
205	Reset Content — éxito, el cliente debe resetear el formulario
206	Partial Content — solicitud de rango satisfecha
207	Multi-Status — múltiples códigos de estado (WebDAV)

Uso en REST API

GET → 200	Devolver recurso con cuerpo
POST → 201	Recurso creado, incluir cabecera Location
PUT → 200/204	Recurso actualizado (con/sin cuerpo)
DELETE → 204	Eliminado, sin cuerpo devuelto
PATCH → 200	Actualización parcial, devolver recurso modificado

Redirección 3xx

Códigos 3xx

300	Multiple Choices — hay múltiples representaciones disponibles
301	Moved Permanently — recurso movido, actualizar marcadores
302	Found — redirección temporal (frecuentemente mal usado como 303)
303	See Other — redirigir con GET después de POST
304	Not Modified — usar versión en caché (ETag/If-Modified)
307	Temporary Redirect — mismo método, ubicación temporal
308	Permanent Redirect — mismo método, ubicación permanente

Comportamiento de redirección

301/308	Permanente — los motores de búsqueda actualizan el índice
302/307	Temporal — la URL original sigue siendo canónica
301/302	Puede cambiar el método a GET en la redirección
307/308	Debe preservar el método HTTP original

Error del cliente 4xx

Errores de cliente comunes

400	Bad Request — sintaxis mal formada o parámetros inválidos
401	Unauthorized — autenticación requerida o fallida
403	Forbidden — autenticado pero sin permiso
404	Not Found — el recurso no existe
405	Method Not Allowed — método HTTP no soportado
406	Not Acceptable — no puede satisfacer la cabecera Accept
408	Request Timeout — el cliente tardó demasiado en enviar la solicitud
409	Conflict — la solicitud entra en conflicto con el estado actual

Más errores de cliente

410	Gone — recurso eliminado permanentemente (no solo faltante)
411	Length Required — falta la cabecera Content-Length
412	Precondition Failed — If-Match/If-Unmodified fallido
413	Content Too Large — el cuerpo de la solicitud excede el límite
414	URI Too Long — la URL excede el límite del servidor
415	Unsupported Media Type — Content-Type no aceptado
422	Unprocessable Content — sintaxis válida, errores semánticos
429	Too Many Requests — límite de tasa excedido

Error del servidor 5xx

Códigos 5xx

500	Internal Server Error — excepción no manejada en el servidor
501	Not Implemented — el servidor no soporta el método
502	Bad Gateway — el servidor upstream envió una respuesta inválida
503	Service Unavailable — sobrecargado o en mantenimiento
504	Gateway Timeout — el servidor upstream no respondió a tiempo
505	HTTP Version Not Supported — versión no manejada
507	Insufficient Storage — el servidor no puede almacenar la solicitud (WebDAV)
511	Network Auth Required — se requiere login en portal cautivo

Estrategia de reintentos

500	Reintentar con backoff; puede ser transitorio
502/504	Reintentar — el problema upstream puede resolverse
503	Verificar cabecera Retry-After antes de reintentar
501/505	No reintentar — corregir la solicitud del cliente

Códigos comunes

Códigos más usados (de un vistazo)

200	OK — todo funcionó
201	Created — nuevo recurso creado
204	No Content — éxito, cuerpo vacío
301	Moved Permanently — actualizar URL
304	Not Modified — usar caché
400	Bad Request — corregir la solicitud
401	Unauthorized — iniciar sesión primero
403	Forbidden — permisos insuficientes
404	Not Found — URL incorrecta o recurso eliminado
422	Unprocessable — errores de validación
429	Too Many Requests — reducir el ritmo
500	Server Error — no es culpa del cliente
502	Bad Gateway — fallo en proxy/upstream
503	Unavailable — intentar más tarde

Referencia de cabeceras

Cabeceras de solicitud

Accept	Tipos de media de respuesta deseados (p. ej. application/json)
Authorization	Credenciales (Bearer token, Basic base64)
Content-Type	Tipo de media del cuerpo de la solicitud
If-None-Match	Condición: ETag para validación de caché
If-Modified-Since	Condición: fecha para validación de caché
Cache-Control	Directivas de caché (no-cache, max-age)
User-Agent	Cadena de identificación del cliente

Cabeceras de respuesta

Content-Type	Tipo de media del cuerpo de la respuesta
Location	URL destino de redirección o recurso creado
ETag	Etiqueta de entidad para validación de caché
Cache-Control	Directivas de caché (max-age, no-store)
Retry-After	Tiempo de espera antes de reintentar (429/503)
WWW-Authenticate	Esquema de autenticación requerido (enviado con 401)
Set-Cookie	Establecer cookie en el cliente

Patrones comunes

Flujo de caché

```
# 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)
```

Flujo de autenticación

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

Límite de tasa

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

Negociación de contenido

```
# 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
```