

RÉFÉRENCE RAPIDE DES CODES HTTP

Codes de statut, en-têtes et modèles de réponse courants

Informatif 1xx

Codes 1xx

- 100** Continue — le serveur a reçu les en-têtes, le client doit envoyer le corps
- 101** Switching Protocols — passage à WebSocket ou HTTP/2
- 102** Processing — requête reçue, traitement en cours (WebDAV)
- 103** Early Hints — précharger les ressources avant la réponse finale

Note d'utilisation

```
# 100 Continue : le client envoie l'en-tête Expect, attend 100
curl -H "Expect: 100-continue" -d @target.json URL
# 101 : passage à WebSocket
Connection: Upgrade / Upgrade: websocket
```

Succès 2xx

Codes 2xx

- 200** OK — réponse de succès standard
- 201** Created — ressource créée avec succès (POST/PUT)
- 202** Accepted — requête reçue, traitement asynchrone
- 203** Non-Authoritative Info — transformée par un proxy
- 204** No Content — succès sans corps de réponse (DELETE)
- 205** Reset Content — succès, le client doit réinitialiser le formulaire
- 206** Partial Content — requête de plage satisfaite
- 207** Multi-Status — codes de statut multiples (WebDAV)

Utilisation API REST

- GET** → **200** Retourner la ressource avec le corps
- POST** → **201** Ressource créée, inclure l'en-tête Location
- PUT** → **200/204** Ressource mise à jour (avec/sans corps)
- DELETE** → **204** Supprimé, aucun corps retourné
- PATCH** → **200** Mise à jour partielle, retourner la ressource modifiée

Redirection 3xx

Codes 3xx

- 300** Multiple Choices — plusieurs représentations disponibles
- 301** Moved Permanently — ressource déplacée, mettre à jour les favoris
- 302** Found — redirection temporaire (souvent mal utilisé comme 303)
- 303** See Other — redirection avec GET après POST
- 304** Not Modified — utiliser la version en cache (ETag/If-Modified)
- 307** Temporary Redirect — même méthode, emplacement temporaire
- 308** Permanent Redirect — même méthode, emplacement permanent

Comportement des redirections

- 301/308** Permanent — les moteurs de recherche mettent à jour l'index
- 302/307** Temporaire — l'URL d'origine reste canonique
- 301/302** Peut changer la méthode en GET lors de la redirection
- 307/308** Doit conserver la méthode HTTP d'origine

Erreur client 4xx

Erreurs client courantes

- 400** Bad Request — syntaxe malformée ou paramètres invalides
- 401** Unauthorized — authentification requise ou échouée
- 403** Forbidden — authentifié mais non autorisé
- 404** Not Found — la ressource n'existe pas
- 405** Method Not Allowed — méthode HTTP non supportée
- 406** Not Acceptable — impossible de satisfaire l'en-tête Accept
- 408** Request Timeout — le client est trop lent pour envoyer la requête
- 409** Conflict — la requête est en conflit avec l'état actuel

Autres erreurs client

- 410** Gone — ressource définitivement supprimée (pas juste absente)
- 411** Length Required — en-tête Content-Length manquant
- 412** Precondition Failed — If-Match/If-Unmodified échoué
- 413** Content Too Large — corps de requête dépasse la limite
- 414** URI Too Long — l'URL dépasse la limite du serveur
- 415** Unsupported Media Type — Content-Type non accepté
- 422** Unprocessable Content — syntaxe valide, erreurs sémantiques
- 429** Too Many Requests — limite de taux dépassée

Erreur serveur 5xx

Codes 5xx

- 500** Internal Server Error — exception non gérée sur le serveur
- 501** Not Implemented — le serveur ne supporte pas la méthode
- 502** Bad Gateway — le serveur amont a envoyé une réponse invalide
- 503** Service Unavailable — surchargé ou en maintenance
- 504** Gateway Timeout — le serveur amont n'a pas répondu à temps
- 505** HTTP Version Not Supported — version non gérée
- 507** Insufficient Storage — le serveur ne peut pas stocker la requête (WebDAV)
- 511** Network Auth Required — connexion au portail captif requise

Stratégie de réessai

- 500** Réessayer avec backoff ; peut être transitoire
- 502/504** Réessayer — le problème amont peut se résoudre
- 503** Vérifier l'en-tête Retry-After avant de réessayer
- 501/505** Ne pas réessayer — corriger la requête client

Codes courants

Codes les plus utilisés (en un coup d'œil)

- 200** OK — tout a fonctionné
- 201** Created — nouvelle ressource créée
- 204** No Content — succès, corps vide
- 301** Moved Permanently — mettre à jour l'URL
- 304** Not Modified — utiliser le cache
- 400** Bad Request — corriger la requête
- 401** Unauthorized — se connecter d'abord
- 403** Forbidden — permissions insuffisantes
- 404** Not Found — mauvaise URL ou supprimé
- 422** Unprocessable — erreurs de validation

- 429** Too Many Requests — ralentir
- 500** Server Error — pas votre faute
- 502** Bad Gateway — défaillance proxy/amont
- 503** Unavailable — réessayer plus tard

Référence des en-têtes

En-têtes de requête

Accept Types de médias de réponse souhaités (ex. application/json)

Authorization Identifiants (token Bearer, Basic base64)

Content-Type Type de média du corps de la requête

If-None-Match Conditionnel : ETag pour la validation du cache

If-Modified-Since Conditionnel : date pour la validation du cache

Cache-Control Directives de mise en cache (no-cache, max-age)

User-Agent Chaîne d'identification du client

En-têtes de réponse

Content-Type Type de média du corps de la réponse

Location URL cible de redirection ou de la ressource créée

ETag Balise d'entité pour la validation du cache

Cache-Control Directives de mise en cache (max-age, no-store)

Retry-After Temps d'attente avant réessai (429/503)

WWW-Authenticate Schéma d'authentification requis (envoyé avec 401)

Set-Cookie Définir un cookie sur le client

Modèles courants

Flux de mise en cache

```
# Première requête — le serveur retourne un ETag
GET /api/data → 200, ETag: "abc123"
# Requête suivante — conditionnelle
GET /api/data, If-None-Match: "abc123"
→ 304 Not Modified (utiliser le cache)
```

Flux d'authentification

```
# Requête non authentifiée
GET /api/secret → 401, WWW-Authenticate: Bearer
# Avec token
GET /api/secret, Authorization: Bearer <token>
→ 200 OK
```

Limitation de débit

```
# Réponse de limitation de débit
429 Too Many Requests
Retry-After: 60
X-RateLimit-Remaining: 0
X-RateLimit-Reset: 1700000000
```

Négociation de contenu

```
# Le client préfère JSON, accepte XML
Accept: application/json, application/xml;q=0.9
# Le serveur ne peut pas satisfaire → 406 Not Acceptable
# Le serveur retourne la meilleure correspondance → 200 + Content-Type
```