

# Référence rapide des codes HTTP

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

## Informatif 1xx

### Codes 1xx

<b>100</b>	Continue — le serveur a reçu les en-têtes, le client doit envoyer le corps
<b>101</b>	Switching Protocols — passage à WebSocket ou HTTP/2
<b>102</b>	Processing — requête reçue, traitement en cours (WebDAV)
<b>103</b>	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 @large.json URL
# 101 : passage à WebSocket
Connection: Upgrade / Upgrade: websocket
```

## Succès 2xx

### Codes 2xx

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

### Utilisation API REST

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

## Redirection 3xx

### Codes 3xx

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

### Comportement des redirections

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

## Erreur client 4xx

### Erreurs client courantes

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

### Autres erreurs client

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

## Erreur serveur 5xx

### Codes 5xx

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

### Stratégie de réessai

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

## Codes courants

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

<b>200</b>	OK — tout a fonctionné
<b>201</b>	Created — nouvelle ressource créée
<b>204</b>	No Content — succès, corps vide
<b>301</b>	Moved Permanently — mettre à jour l'URL
<b>304</b>	Not Modified — utiliser le cache
<b>400</b>	Bad Request — corriger la requête
<b>401</b>	Unauthorized — se connecter d'abord
<b>403</b>	Forbidden — permissions insuffisantes
<b>404</b>	Not Found — mauvaise URL ou supprimé
<b>422</b>	Unprocessable — erreurs de validation
<b>429</b>	Too Many Requests — ralentir
<b>500</b>	Server Error — pas votre faute
<b>502</b>	Bad Gateway — défaillance proxy/amont
<b>503</b>	Unavailable — réessayer plus tard

## Référence des en-têtes

### En-têtes de requête

<b>Accept</b>	Types de médias de réponse souhaités (ex. application/json)
<b>Authorization</b>	Identifiants (token Bearer, Basic base64)
<b>Content-Type</b>	Type de média du corps de la requête
<b>If-None-Match</b>	Conditionnel : ETag pour la validation du cache
<b>If-Modified-Since</b>	Conditionnel : date pour la validation du cache
<b>Cache-Control</b>	Directives de mise en cache (no-cache, max-age)
<b>User-Agent</b>	Chaîne d'identification du client

### En-têtes de réponse

<b>Content-Type</b>	Type de média du corps de la réponse
<b>Location</b>	URL cible de redirection ou de la ressource créée
<b>ETag</b>	Balise d'entité pour la validation du cache
<b>Cache-Control</b>	Directives de mise en cache (max-age, no-store)
<b>Retry-After</b>	Temps d'attente avant réessai (429/503)
<b>WWW-Authenticate</b>	Schéma d'authentification requis (envoyé avec 401)
<b>Set-Cookie</b>	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
```