

# Référence rapide curl

Requêtes HTTP, en-têtes, authentification, formulaires, débogage

## Utilisation de base

### Requêtes simples

```
curl https://example.com # requête GET
curl -o file.html https://url # sauvegarder dans un fichier
curl -O https://url/file.tar.gz # sauvegarder avec le nom distant
curl -L https://url # suivre les redirections
```

### Options courantes

<b>-s</b>	Mode silencieux (sans progression)
<b>-S</b>	Afficher les erreurs en mode silencieux
<b>-f</b>	Échouer silencieusement sur les erreurs HTTP
<b>-L</b>	Suivre les redirections
<b>-o file</b>	Écrire la sortie dans un fichier
<b>-O</b>	Sauvegarder avec le nom du fichier distant
<b>-C -</b>	Reprendre un téléchargement interrompu
<b>--max-time 30</b>	Expiration après 30 secondes

## Méthodes HTTP

### GET et HEAD

```
curl https://api.example.com/users
curl -I https://example.com # HEAD (en-têtes seulement)
curl -i https://example.com # inclure les en-têtes de réponse
```

### POST

```
curl -X POST https://api.example.com/users \
-H "Content-Type: application/json" \
-d '{"name":"Jo","email":"jo@ex.com"}'
```

### PUT, PATCH et DELETE

```
curl -X PUT https://api.example.com/users/1 \
-d '{"name":"Updated"}'
curl -X PATCH https://api.example.com/users/1 \
-d '{"email":"new@ex.com"}'
curl -X DELETE https://api.example.com/users/1
```

## En-têtes

### Définir des en-têtes

```
curl -H "Content-Type: application/json" URL
curl -H "Accept: text/html" URL
curl -H "X-Custom: value" URL
curl -H "Header1: v1" -H "Header2: v2" URL
```

### En-têtes de réponse

<b>-i</b>	Inclure les en-têtes de réponse dans la sortie
<b>-I</b>	Récupérer les en-têtes seulement (HEAD)
<b>-D file</b>	Vider les en-têtes de réponse dans un fichier
<b>-w '%{http_code}'</b>	Afficher le code de statut HTTP

## Authentification

### Auth basique et par token

```
curl -u user:pass https://api.example.com
curl -H "Authorization: Bearer TOKEN" URL
curl -u user:pass --digest URL
curl --negotiate -u : URL # Kerberos/SPNEGO
```

### Méthodes d'authentification

<b>-u user:pass</b>	Authentification basique
<b>--digest</b>	Auth HTTP Digest
<b>--negotiate</b>	Auth Kerberos/SPNEGO
<b>--ntlm</b>	Authentification NTLM
<b>-n</b>	Utiliser les identifiants ~/.netrc

## Données et formulaires

### Envoyer des données

```
curl -d "key=val&key2=val2" URL # form urlencodé
curl -d @data.json URL # données depuis un fichier
curl --data-raw '{"raw":"json"}' URL
curl --data-urlencode "q=hello world" URL
```

### Téléversement de fichiers

```
curl -F "file=@photo.jpg" URL
curl -F "file=@doc.pdf;type=application/pdf" URL
curl -F "field=value" -F "file=@img.png" URL
```

### Multipart vs URL-Encoded

<b>-d</b>	application/x-www-form-urlencoded
<b>-F</b>	multipart/form-data
<b>--json</b>	Raccourci : définit Content-Type + Accept à JSON
<b>-T file</b>	Téléverser un fichier via PUT

## SSL/TLS

### Options de certificat

```
curl --cacert ca.pem URL # bundle CA personnalisé
curl --cert client.pem URL # certificat client
curl --cert client.pem --key key.pem URL
curl -k URL # ignorer la vérification TLS (dev uniquement)
```

### Options TLS

<b>-k / --insecure</b>	Ignorer la vérification du certificat TLS
<b>--cacert file</b>	Utiliser un certificat CA personnalisé
<b>--cert file</b>	Certificat client
<b>--key file</b>	Clé privée client
<b>--tlsv1.2</b>	Forcer TLS 1.2 minimum
<b>--tlsv1.3</b>	Forcer TLS 1.3 minimum

## Sortie et débogage

### Mode verbeux et trace

```
curl -v URL # sortie verbeuse
curl --trace dump.txt URL # trace complète dans un fichier
curl --trace-ascii - URL # trace sur stdout
curl -w "\n%{http_code}\n" URL # format de sortie personnalisé
```

### Variables write-out

<b>%{http_code}</b>	Code de statut HTTP
<b>%{time_total}</b>	Temps total en secondes
<b>%{time_connect}</b>	Temps pour établir la connexion
<b>%{size_download}</b>	Octets téléchargés
<b>%{speed_download}</b>	Vitesse de téléchargement moyenne
<b>%{redirect_url}</b>	URL de redirection (si applicable)
<b>%{ssl_verify_result}</b>	Résultat de la vérification SSL (0 = OK)

### Exemple write-out

```
curl -s -o /dev/null -w \
"code: %{http_code}\ntime: %{time_total}s\n" \
https://example.com
```

## Patterns courants

### Workflow API

```
# Obtenir du JSON et envoyer vers jq
curl -s https://api.example.com/data | jq '.items[]'
# POST JSON avec auth
curl -s -H "Authorization: Bearer $TOKEN" \
--json '{"key":"val"}' https://api.example.com
```

## Patterns de téléchargement

```
# Télécharger avec barre de progression
curl -# -O https://releases.example.com/v2.tar.gz
# Reprendre un téléchargement interrompu
curl -C - -O https://releases.example.com/v2.tar.gz
# Télécharger plusieurs fichiers
curl -O https://url/file1 -O https://url/file2
```

## Aides pour les scripts

```
# Vérifier si une URL est accessible
curl -sf -o /dev/null https://example.com && echo OK
# Sauvegarder et réutiliser les cookies
curl -c cookies.txt -b cookies.txt URL
# Limiter le débit de la requête
curl --limit-rate 100k URL
```

## Proxy et réseau

### Paramètres proxy

```
curl -x http://proxy:8080 URL
curl -x socks5://proxy:1080 URL
curl --proxy-user user:pass -x http://proxy:8080 URL
curl --noproxy "*.local,localhost" URL
```

### DNS et résolution

<b>--resolve host:port:addr</b>	Forcer la résolution DNS vers addr
<b>--dns-servers 8.8.8.8</b>	Utiliser un serveur DNS personnalisé
<b>--interface eth0</b>	Utiliser une interface réseau spécifique
<b>-4 / -6</b>	Forcer IPv4 / IPv6

## Config et avancé

### Fichiers de config

```
# ~/.curlrc – options par défaut
--silent
--location
--max-time 30
```

```
# Utiliser un fichier de config explicitement
curl -K myconfig.txt URL
```

### Options utiles

<b>--retry 3</b>	Réessayer en cas d'erreurs transitoires
<b>--retry-delay 2</b>	Délai entre les tentatives (secondes)
<b>--compressed</b>	Demander et décompresser gzip/br
<b>--limit-rate 100k</b>	Limiter la vitesse de transfert
<b>-Z</b>	Transferts parallèles (curl 7.66+)
<b>--create-dirs</b>	Créer les répertoires du chemin pour -o