

# curl Tham Khảo Nhanh

HTTP request, header, xác thực, form, debug

## Sử Dụng Cơ Bản

### Request Đơn Giản

```
curl https://example.com # GET request
curl -o file.html https://url # lưu vào file
curl -O https://url/file.tar.gz # lưu theo tên từ server
curl -L https://url # theo redirect
```

### Flag Thường Dùng

<b>-s</b>	Chế độ im lặng (không hiện tiến trình)
<b>-S</b>	Hiển thị lỗi trong chế độ im lặng
<b>-f</b>	Thất bại im lặng khi lỗi HTTP
<b>-L</b>	Theo redirect
<b>-o file</b>	Ghi đầu ra vào file
<b>-O</b>	Lưu theo tên file từ xa
<b>-C -</b>	Tiếp tục tải bị gián đoạn
<b>--max-time 30</b>	Timeout sau 30 giây

## HTTP Methods

### GET & HEAD

```
curl https://api.example.com/users
curl -I https://example.com # HEAD (chỉ header)
curl -i https://example.com # bao gồm response header
```

### POST

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

### PUT & PATCH & 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
```

## Headers

### Đặt Headers

```
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
```

### Response Headers

<b>-i</b>	Bao gồm response header trong đầu ra
<b>-I</b>	Chỉ lấy header (HEAD)
<b>-D file</b>	Dump response header vào file
<b>-w '%{http_code}'</b>	In HTTP status code

## Xác Thực

### Basic & Token Auth

```
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
```

### Các Phương Thức Auth

<b>-u user:pass</b>	Xác thực Basic
<b>--digest</b>	HTTP Digest auth
<b>--negotiate</b>	Kerberos/SPNEGO auth
<b>--ntlm</b>	NTLM authentication
<b>-n</b>	Dùng thông tin xác thực từ ~/.netrc

## Dữ Liệu & Form

### Gửi Dữ Liệu

```
curl -d "key=val&key2=val2" URL # form urlencoded
curl -d @data.json URL # dữ liệu từ file
curl --data-raw '{"raw":"json"}' URL
curl --data-urlencode "q=hello world" URL
```

### Tải File Lên

```
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>	Rút gọn: đặt Content-Type + Accept thành JSON
<b>-T file</b>	Tải file lên qua PUT

## SSL/TLS

### Tùy Chọn Certificate

```
curl --cacert ca.pem URL # CA bundle tùy chỉnh
curl --cert client.pem URL # client certificate
curl --cert client.pem --key key.pem URL
curl -k URL # bỏ qua xác minh TLS (chỉ dev)
```

### TLS Flags

<b>-k / --insecure</b>	Bỏ qua xác minh certificate TLS
<b>--cacert file</b>	Dùng CA certificate tùy chỉnh
<b>--cert file</b>	Client certificate
<b>--key file</b>	Private key của client
<b>--tlsv1.2</b>	Bắt buộc TLS tối thiểu 1.2
<b>--tlsv1.3</b>	Bắt buộc TLS tối thiểu 1.3

## Đầu Ra & Debug

### Verbose & Trace

```
curl -v URL # đầu ra verbose
curl --trace dump.txt URL # toàn bộ trace vào file
curl --trace-ascii - URL # trace ra stdout
curl -w "\n%{http_code}\n" URL # định dạng đầu ra tùy chỉnh
```

### Biến Write-Out

<b>%{http_code}</b>	HTTP response status code
<b>%{time_total}</b>	Tổng thời gian tính bằng giây
<b>%{time_connect}</b>	Thời gian thiết lập kết nối
<b>%{size_download}</b>	Byte đã tải
<b>%{speed_download}</b>	Tốc độ tải trung bình
<b>%{redirect_url}</b>	URL redirect (nếu có)
<b>%{ssl_verify_result}</b>	Kết quả xác minh SSL (0 = OK)

### Ví Dụ Write-Out

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

## Các Pattern Thường Gặp

### Workflow API

```
# GET JSON và pipe vào jq
curl -s https://api.example.com/data | jq '.items[]'
# POST JSON có xác thực
curl -s -H "Authorization: Bearer $TOKEN" \
--json '{"key":"val"}' https://api.example.com
```

## Các Pattern Tải

```
# Tải với progress bar
curl -# -O https://releases.example.com/v2.tar.gz
# Tiếp tục tải bị gián đoạn
curl -C - -O https://releases.example.com/v2.tar.gz
# Tải nhiều file
curl -O https://url/file1 -O https://url/file2
```

## Script Helper

```
# Kiểm tra URL có truy cập được không
curl -sf -o /dev/null https://example.com && echo OK
# Lưu cookie và tái sử dụng
curl -c cookies.txt -b cookies.txt URL
# Giới hạn tốc độ request
curl --limit-rate 100k URL
```

## Proxy & Mạng

### Cài Đặt 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 & Resolve

<b>--resolve host:port:addr</b>	Ép phân giải DNS thành addr
<b>--dns-servers 8.8.8.8</b>	Dùng DNS server tùy chỉnh
<b>--interface eth0</b>	Dùng network interface cụ thể
<b>-4 / -6</b>	Bắt buộc IPv4 / IPv6

## Config & Nâng Cao

### File Config

```
# ~/.curlrc - tùy chọn mặc định
--silent
--location
--max-time 30
```

```
# Dùng file config riêng mình
curl -K myconfig.txt URL
```

### Flag Hữu Ích

<b>--retry 3</b>	Thử lại khi gặp lỗi tạm thời
<b>--retry-delay 2</b>	Trễ giữa các lần thử lại (giây)
<b>--compressed</b>	Yêu cầu và giải nén gzip/br
<b>--limit-rate 100k</b>	Giới hạn tốc độ truyền
<b>-Z</b>	Truyền song song (curl 7.66+)
<b>--create-dirs</b>	Tạo thư mục đường dẫn cho -o