

CURL 빠른 참조

HTTP 요청, 헤더, 인증, 폼, 디버깅

기본 사용법

간단한 요청

```
curl https://example.com # GET request
curl -o file.html https://url # save to file
curl -O https://url/file.tar.gz # save with remote name
curl -L https://url # follow redirects
```

주요 플래그

- s** 조용한 모드 (진행 표시 없음)
- S** 조용한 모드에서 오류 표시
- i** HTTP 오류 시 조용히 실패
- L** 리다이렉트 따라가기
- o file** 파일로 출력 저장
- O** 원격 파일명으로 저장
- C -** 중단된 다운로드 재개
- max-time 30** 30초 후 타임아웃

HTTP 메서드

GET & HEAD

```
curl https://api.example.com/users
curl -I https://example.com # HEAD (headers only)
curl -I https://example.com # include response headers
```

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
```

헤더

헤더 설정

```
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
```

응답 헤더

- i** 출력에 응답 헤더 포함
- I** 헤더만 가져오기 (HEAD)
- D file** 응답 헤더를 파일로 덤프
- w '%{http_code}'** HTTP 상태 코드 출력

인증

Basic & Token 인증

```
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
```

인증 방법

- u user:pass** Basic 인증
- digest** HTTP Digest 인증
- negotiate** Kerberos/SPNEGO 인증
- ntlm** NTLM 인증
- n** ~/.netrc 자격증명 사용

데이터 & 폼

데이터 전송

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

파일 업로드

```
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

- d** application/x-www-form-urlencoded
- F** multipart/form-data
- json** 단축; Content-Type + Accept를 JSON으로 설정
- T file** PUT으로 파일 업로드

SSL/TLS

인증서 옵션

```
curl --cacert ca.pem URL # custom CA bundle
curl --cert client.pem URL # client certificate
curl --cert client.pem --key key.pem URL
curl -k URL # skip TLS verify (dev only)
```

TLS 플래그

- k / --insecure** TLS 인증서 검증 건너뛸
- cacert file** 커스텀 CA 인증서 사용
- cert file** 클라이언트 인증서
- key file** 클라이언트 개인키
- tlsv1.2** 최소 TLS 1.2 강제
- tlsv1.3** 최소 TLS 1.3 강제

출력 & 디버깅

상세 & 추적

```
curl -v URL # verbose output
curl --trace dump.txt URL # full trace to file
curl --trace-ascii - URL # trace to stdout
curl -w "%{http_code}\n" URL # custom output format
```

Write-Out 변수

- %{http_code}** HTTP 응답 상태 코드
- %{time_total}** 총 소요 시간 (초)
- %{time_connect}** 연결 수립 시간
- %{size_download}** 다운로드된 바이트

- %{speed_download}** 평균 다운로드 속도
- %{redirect_url}** 리다이렉트 URL (있는 경우)
- %{ssl_verify_result}** SSL 검증 결과 (0 = 정상)

Write-Out 예시

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

일반 패턴

API 워크플로우

```
# GET JSON and pipe to jq
curl -s https://api.example.com/data | jq '.items[]'
# POST JSON with auth
curl -s -H "Authorization: Bearer $TOKEN" \
--json '{"key": "val"}' https://api.example.com
```

다운로드 패턴

```
# Download with progress bar
curl -# -O https://releases.example.com/v2.tar.gz
# Resume interrupted download
curl -C - -O https://releases.example.com/v2.tar.gz
# Download multiple files
curl -O https://url/file1 -O https://url/file2
```

스크립팅 도구미

```
# Check if url is reachable
curl -sf -o /dev/null https://example.com && echo OK
# Save cookies and reuse
curl -c cookies.txt -b cookies.txt URL
# Rate limit request
curl --limit-rate 100k URL
```

프록시 & 네트워크

프록시 설정

```
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 & 해석

- resolve host:port:addr** DNS 해석을 addr로 강제
- dns-servers 8.8.8.8** 커스텀 DNS 서버 사용
- interface eth0** 특정 네트워크 인터페이스 사용
- 4 / -6** IPv4 / IPv6 강제

설정 & 고급

설정 파일

```
# ~/.curlrc - default options
--silent
--location
--max-time 30

# Use config file explicitly
curl -K myconfig.txt URL
```

유용한 플래그

- retry 3** 일시적 오류 시 재시도
- retry-delay 2** 재시도 간격 (초)
- compressed** gzip/br 요청 및 압축 해제
- limit-rate 100k** 전송 속도 제한
- Z** 병렬 전송 (curl 7.66+)
- create-dirs** -o를 위한 경로 디렉토리 생성