

RIFERIMENTO RAPIDO CURL

Richieste HTTP, header, autenticazione, form, debugging

Utilizzo Base

Richieste Semplici

```
curl https://example.com # richiesta GET
curl -o file.html https://url # salva su file
curl -O https://url/file.tar.gz # salva con nome remoto
curl -L https://url # sequi redirect
```

Flag Comuni

```
-s Modalità silenziosa (nessun progresso)
-S Mostra errori in modalità silenziosa
-f Fallisce silenziosamente su errori HTTP
-L Segui i redirect
-o file Scrivi output su file
-O Salva con nome file remoto
-C Riprendi download interrotto
--max-time 30 Timeout dopo 30 secondi
```

Metodi HTTP

GET e HEAD

```
curl https://api.example.com/users
curl -I https://example.com # HEAD (solo header)
curl -i https://example.com # include header risposta
```

POST

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

PUT, PATCH e 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
```

Header

Impostare gli Header

```
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
```

Header di Risposta

```
-i Include header di risposta nell'output
-I Scarica solo gli header (HEAD)
-D file Salva header di risposta su file
-w '%{http_code}' Stampa il codice di stato HTTP
```

Autenticazione

Auth Basic e 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
```

Metodi di Autenticazione

```
--u user:pass Autenticazione Basic
--digest HTTP Digest auth
--negotiate Autenticazione Kerberos/SPNEGO
--ntlm Autenticazione NTLM
-n Usa credenziali ~/.netrc
```

Dati e Form

Invio Dati

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

Upload File

```
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 Scorciatoia: imposta Content-Type + Accept a JSON
-T file Carica file via PUT
```

SSL/TLS

Opzioni Certificato

```
curl --cacert ca.pem URL # bundle CA personalizzato
curl --cert client.pem URL # certificato client
curl --cert client.pem --key key.pem URL
curl -k URL # salta verifica TLS (solo sviluppo)
```

Flag TLS

```
-k / --insecure Salta la verifica del certificato TLS
--cacert file Usa certificato CA personalizzato
--cert file Certificato client
--key file Chiave privata client
--tlsv1.2 Forza TLS minimo 1.2
--tlsv1.3 Forza TLS minimo 1.3
```

Output e Debugging

Verbose e Trace

```
curl -v URL # output verbose
curl --trace dump.txt URL # trace completa su file
curl --trace-ascii - URL # trace su stdout
curl -w "%n(%{http_code})\n" URL # formato output personalizzato
```

Variabili Write-Out

```
%{http_code} Codice di stato risposta HTTP
%{time_total} Tempo totale in secondi
%{time_connect} Tempo per stabilire la connessione
%{size_download} Byte scaricati
%{speed_download} Velocità media di download
```

```
%{redirect_url} URL di redirect (se presente)
%{ssl_verify_result} Risultato verifica SSL (0 = OK)
```

Esempio Write-Out

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

Pattern Comuni

Workflow API

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

Pattern di Download

```
# Download con barra progresso
curl -# -O https://releases.example.com/v2.tar.gz
# Riprendi download interrotto
curl -C - -O https://releases.example.com/v2.tar.gz
# Download di file multipli
curl -O https://url/file1 -O https://url/file2
```

Helper per Scripting

```
# Verifica se l'URL è raggiungibile
curl -sf -o /dev/null https://example.com && echo OK
# Salva e riusa cookie
curl -c cookies.txt -b cookies.txt URL
# Limita la velocità della richiesta
curl --limit-rate 100k URL
```

Proxy e Rete

Impostazioni 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 e Resolve

```
--resolve host:port:addr Forza risoluzione DNS a addr
--dns-servers 8.8.8.8 Usa server DNS personalizzato
--interface eth0 Usa interfaccia di rete specifica
-4 / -6 Forza IPv4 / IPv6
```

Config e Avanzato

File di Configurazione

```
# ~/.curlrc - opzioni predefinite
--silent
--location
--max-time 30
# Usa file di configurazione esplicitamente
curl -K myconfig.txt URL
```

Flag Utili

```
--retry 3 Riprova in caso di errori transitori
--retry-delay 2 Attesa tra i tentativi (secondi)
--compressed Richiede e decomprime gzip/br
--limit-rate 100k Limita la velocità di trasferimento
-Z Trasferimenti paralleli (curl 7.66+)
--create-dirs Crea directory del percorso per-o
```