

# Referência Rápida curl

Requisições HTTP, headers, autenticação, formulários, depuração

## Uso Básico

### Requisições Simples

```
curl https://example.com # GET request
curl -o file.html https://url # salvar em arquivo
curl -O https://url/file.tar.gz # salvar com nome remoto
curl -L https://url # seguir redirecionamentos
```

### Flags Comuns

<b>-s</b>	Modo silencioso (sem progresso)
<b>-S</b>	Mostrar erros no modo silencioso
<b>-f</b>	Falhar silenciosamente em erros HTTP
<b>-L</b>	Seguir redirecionamentos
<b>-o file</b>	Gravar saída em arquivo
<b>-O</b>	Salvar com nome do arquivo remoto
<b>-C -</b>	Retomar download interrompido
<b>--max-time 30</b>	Timeout após 30 segundos

## Métodos HTTP

### GET e HEAD

```
curl https://api.example.com/users
curl -I https://example.com # HEAD (somente headers)
curl -i https://example.com # incluir headers de resposta
```

### 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
```

## Headers

### Definindo 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
```

### Headers de Resposta

<b>-i</b>	Incluir headers de resposta na saída
<b>-I</b>	Buscar somente headers (HEAD)
<b>-D file</b>	Despejar headers de resposta em arquivo
<b>-w '%{http_code}'</b>	Imprimir código de status HTTP

## Autenticação

### Auth Básica e por 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
```

### Métodos de Auth

<b>-u user:pass</b>	Autenticação básica
<b>--digest</b>	Auth HTTP Digest
<b>--negotiate</b>	Auth Kerberos/SPNEGO
<b>--ntlm</b>	Autenticação NTLM
<b>-n</b>	Usar credenciais de ~/.netrc

## Dados e Formulários

### Enviando Dados

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

### Upload de Arquivos

```
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>	Atalho: define Content-Type + Accept como JSON
<b>-T file</b>	Upload de arquivo via PUT

## SSL/TLS

### Opções de Certificado

```
curl --cacert ca.pem URL # bundle de CA personalizado
curl --cert client.pem URL # certificado do cliente
curl --cert client.pem --key key.pem URL
curl -k URL # ignorar verificação TLS (só dev)
```

### Flags TLS

<b>-k / --insecure</b>	Ignorar verificação de certificado TLS
<b>--cacert file</b>	Usar certificado CA personalizado
<b>--cert file</b>	Certificado do cliente
<b>--key file</b>	Chave privada do cliente
<b>--tlsv1.2</b>	Forçar TLS 1.2 mínimo
<b>--tlsv1.3</b>	Forçar TLS 1.3 mínimo

## Saída e Depuração

### Verbose e Trace

```
curl -v URL # saída verbose
curl --trace dump.txt URL # trace completo para arquivo
curl --trace-ascii - URL # trace para stdout
curl -w "\n{%http_code}\n" URL # formato de saída personalizado
```

### Variáveis Write-Out

<b>%{http_code}</b>	Código de status HTTP da resposta
<b>%{time_total}</b>	Tempo total em segundos
<b>%{time_connect}</b>	Tempo para estabelecer conexão
<b>%{size_download}</b>	Bytes baixados
<b>%{speed_download}</b>	Velocidade média de download
<b>%{redirect_url}</b>	URL de redirecionamento (se houver)
<b>%{ssl_verify_result}</b>	Resultado de verificação SSL (0 = OK)

### Exemplo Write-Out

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

## Padrões Comuns

### Fluxo de API

```
# 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
```

## Padrões de Download

```
# Download with progress bar
curl -# -0 https://releases.example.com/v2.tar.gz
# Resume interrupted download
curl -C - -0 https://releases.example.com/v2.tar.gz
# Download multiple files
curl -0 https://url/file1 -0 https://url/file2
```

## Auxiliares de Script

```
# 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
```

## Proxy e Rede

### Configurações de 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 Resolução

<b>--resolve host:port:addr</b>	Forçar resolução DNS para addr
<b>--dns-servers 8.8.8.8</b>	Usar servidor DNS personalizado
<b>--interface eth0</b>	Usar interface de rede específica
<b>-4 / -6</b>	Forçar IPv4 / IPv6

## Configuração e Avançado

### Arquivos de Configuração

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

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

### Flags Úteis

<b>--retry 3</b>	Tentar novamente em erros transitórios
<b>--retry-delay 2</b>	Aguardar entre tentativas (segundos)
<b>--compressed</b>	Solicitar e descomprimir gzip/br
<b>--limit-rate 100k</b>	Limitar velocidade de transferência
<b>-Z</b>	Transferências paralelas (curl 7.66+)
<b>--create-dirs</b>	Criar diretórios do caminho para -o