

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

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

```
-i Incluir headers de resposta na saída
-I Buscar somente headers (HEAD)
-D file Despejar headers de resposta em arquivo
-w '%{http_code}' 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

```
--u user:pass Autenticação básica
--digest Auth HTTP Digest
--negotiate Auth Kerberos/SPNEGO
--ntlm Autenticação NTLM
-n 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

```
-d application/x-www-form-urlencoded
-F multipart/form-data
--json Atalho: define Content-Type + Accept como JSON
-T file 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

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

```
%{http_code} Código de status HTTP da resposta
%{time_total} Tempo total em segundos
%{time_connect} Tempo para estabelecer conexão
%{size_download} Bytes baixados
%{speed_download} Velocidade média de download
```

```
%{redirect_url} URL de redirecionamento (se houver)
%{ssl_verify_result} 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 -# -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
```

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 --no-proxy "*" local,localhost" URL
```

DNS e Resolução

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

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