

# REFERENCIA RÁPIDA DE CURL

Peticiones HTTP, cabeceras, autenticación, formularios, depuración

## Uso Básico

### Peticiones Simples

```
curl https://example.com # petición GET
curl -o file.html https://url # guardar en archivo
curl -O https://url/file.tar.gz # guardar con nombre remoto
curl -L https://url # seguir redirecciones
```

### Flags Comunes

```
-s Modo silencioso (sin progreso)
-S Mostrar errores en modo silencioso
-f Fallar silenciosamente ante errores HTTP
-L Seguir redirecciones
-o file Escribir salida en archivo
-O Guardar con nombre de archivo remoto
-C - Reiniciar descarga interrumpida
--max-time 30 Timeout después de 30 segundos
```

## Métodos HTTP

### GET y HEAD

```
curl https://api.example.com/users
curl -I https://example.com # HEAD (solo cabeceras)
curl -i https://example.com # incluir cabeceras de respuesta
```

### POST

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

### PUT, PATCH y 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
```

## Cabeceras

### Establecer Cabeceras

```
curl -H "Content-Type: application/json" URL
curl -H "Accept: text/html" URL
curl -X "Custom: value" URL
curl -H "Header1: v1" -H "Header2: v2" URL
```

### Cabeceras de Respuesta

```
-i Incluir cabeceras de respuesta en la salida
-I Obtener solo cabeceras (HEAD)
-D file Volcar cabeceras de respuesta en archivo
-w '%{http_code}s' Imprimir código de estado HTTP
```

## Autenticación

### Auth Básica y 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 Autenticación básica
--digest HTTP Digest auth
--negotiate Auth Kerberos/SPNEGO
--ntlm Autenticación NTLM
-n Usar credenciales de ~/.netrc
```

## Datos y Formularios

### Enviar Datos

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

### Subir Archivos

```
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 Abreviación: establece Content-Type + Accept a JSON
-T file Subir archivo via PUT
```

## SSL/TLS

### Opciones de Certificado

```
curl --cacert ca.pem URL # paquete CA personalizado
curl --cert client.pem URL # certificado cliente
curl --cert client.pem --key key.pem URL
curl -k URL # omitir verificación TLS (solo dev)
```

### Flags TLS

```
-k / --insecure Omitir verificación del certificado TLS
--cacert file Usar certificado CA personalizado
--cert file Certificado del cliente
--key file Clave privada del cliente
--tls1.2 Forzar mínimo TLS 1.2
--tls1.3 Forzar mínimo TLS 1.3
```

## Salida y Depuración

### Verbose y Traza

```
curl -v URL # salida detallada
curl --trace dump.txt URL # traza completa a archivo
curl --trace-ascii - URL # traza a stdout
curl -w "%{http_code}\n" URL # formato de salida personalizado
```

### Variables Write-Out

```
%{http_code} Código de estado HTTP de respuesta
%{time_total} Tiempo total en segundos
%{time_connect} Tiempo para establecer conexión
%{size_download} Bytes descargados
```

```
%{speed_download} Velocidad promedio de descarga
%{redirect_url} URL de redirección (si existe)
%{ssl_verify_result} Resultado de verificación SSL (0 = OK)
```

### Ejemplo de Write-Out

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

## Patrones Comunes

### Flujo de Trabajo API

```
# GET JSON y 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
```

### Patrones de Descarga

```
# Descargar con barra de progreso
curl -# -O https://releases.example.com/v2.tar.gz
# Reiniciar descarga interrumpida
curl -C - -O https://releases.example.com/v2.tar.gz
# Descargar múltiples archivos
curl -O https://url/file1 -O https://url/file2
```

### Ayudas para Scripting

```
# Verificar si URL es accesible
curl -sf -o /dev/null https://example.com && echo OK
# Guardar cookies y reutilizar
curl -c cookies.txt -b cookies.txt URL
# Limitar velocidad de petición
curl --limit-rate 100k URL
```

## Proxy y Red

### Configuración 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 y Resolución

```
--resolve host:port:addr Forzar resolución DNS a addr
--dns-servers 8.8.8.8 Usar servidor DNS personalizado
--interface eth0 Usar interfaz de red específica
-4 / -6 Forzar IPv4 / IPv6
```

## Config y Avanzado

### Archivos de Config

```
# -/, curlrc - opciones por defecto
--silent
--location
--max-time 30
# Usar archivo de config explícitamente
curl -k myconfig.txt URL
```

### Flags Útiles

```
--retry 3 Reintentar ante errores transitorios
--retry-delay 2 Espera entre reintentos (segundos)
--compressed Solicitar y descomprimir gzip/br
--limit-rate 100k Limitar velocidad de transferencia
-Z Transferencias paralelas (curl 7.66+)
--create-dirs Crear directorios de ruta para -o
```