



Basic Usage & Options

Basic Syntax

```
curl [options] [URL]
```

Basic curl command structure. Options modify the behavior, and URL specifies the target.

```
curl https://example.com
```

Simplest usage: retrieves the content of example.com and prints it to standard output.

Output and Verbosity

-o Saves the output to the specified file instead of printing to stdout.

<file> or **--output <file>**

Example:

```
curl -o downloaded.html https://example.com
```

-v or **--verbose** Enables verbose mode, showing detailed information about the request process (headers, connection details, etc.).

Example:

```
curl -v https://example.com
```

-s or **--silent** Silent mode. Doesn't show progress meter or error messages. Useful for scripts.

Example:

```
curl -s https://example.com > /dev/null
```

-S or **--show-error** When used with **-s**, shows error messages but suppresses the progress meter.

Example:

```
curl -sS https://example.com
```

Headers

-i or **--include** Includes the HTTP headers in the output.

Example:

```
curl -i https://example.com
```

-I or **--head** Performs a HEAD request, retrieving only the HTTP headers. Good for checking resource status.

Example:

```
curl -I https://example.com
```

-H "Header: Value" or **--header "Header: Value"** Adds a custom header to the request.

Example:

```
curl -H "Content-Type: application/json" https://example.com
```

--compressed Requests a compressed response using deflate or gzip.

Example:

```
curl --compressed https://example.com
```

Requests and Data Handling

Request Types

-X <command> or **--request <command>** Specifies the HTTP request method (e.g., GET, POST, PUT, DELETE).

Example:

```
curl -X POST https://example.com
```

-L or **--location** Follows HTTP redirects. Useful when a resource has moved.

Example:

```
curl -L https://example.com
```

-G or **--get** Sends data with the HTTP GET method.

Example:

```
curl -G -d "param1=value1&param2=value2" https://example.com
```

Data Transfer

-d 'data' or **--data 'data'** Sends HTTP POST data, URL-encoded.

Example:

```
curl -d "name=John&age=30" https://example.com
```

-d @file or **--data @file** Sends data from a file as HTTP POST data.

Example:

```
curl -d @data.json https://example.com
```

-F Used for multipart/form-data, typically for file uploads.

<name=content> or **--form <name=content>**

Example:

```
curl -F "file=@my_image.jpg" https://example.com/upload
curl -F "data=some text" https://example.com
```

Authentication

-u Provides username and password for HTTP authentication.

<user:password> or **--user <user:password>**

Example:

```
curl -u user:password https://example.com
```

--basic Use HTTP Basic Authentication (default when using **-u**).

Example:

```
curl --basic -u user:password https://example.com
```

--digest Use HTTP Digest Authentication

Example:

```
curl --digest -u user:password https://example.com
```

SSL/TLS Options

Certificate Verification

<code>--cacert <file></code>	Specifies the CA certificate file for verifying the server's certificate.
Example:	<code>curl --cacert ca.pem https://example.com</code>
<code>--capath <dir></code>	Specifies a directory containing CA certificates.
Example:	<code>curl --capath /etc/ssl/certs https://example.com</code>
<code>-k</code> or <code>--insecure</code>	Skips SSL certificate verification. Use with caution!
Example:	<code>curl -k https://self-signed.example.com</code>

Client Certificates

<code>-E <cert></code> or <code>--cert <cert></code>	Specifies the client certificate file.
Example:	<code>curl --cert client.pem https://example.com</code>
<code>--cert-type <type></code>	Specifies the client certificate type (DER, PEM, ENG).
Example:	<code>curl --cert client.pem --cert-type PEM https://example.com</code>
<code>-key</code>	Private key file name (SSL)
Example:	<code>curl --key private.key https://example.com</code>

Other SSL Options

<code>--ciphers <list></code>	SSL ciphers to use
Example:	<code>curl --ciphers ECDHE-RSA-AES128-GCM-SHA256 https://example.com</code>
<code>--tlsv1.0</code> , <code>--tlsv1.1</code> , <code>--tlsv1.2</code> , <code>--tlsv1.3</code>	Force curl to use specified TLS version.
Example:	<code>curl --tlsv1.2 https://example.com</code>
<code>--ssl-allow-beast</code>	Allow security flaws present in older SSL/TLS versions
Example:	<code>curl --ssl-allow-beast https://example.com</code>

Advanced Usage

Cookies

<code>-b <name=value></code> or <code>--cookie <name=value></code>	Sends a cookie with the request.
Example:	<code>curl -b "sessionId=12345" https://example.com</code>
<code>-b <file></code> or <code>--cookie <file></code>	Reads cookies from a file.
Example:	<code>curl -b cookies.txt https://example.com</code>
<code>-c <file></code> or <code>--cookie-jar <file></code>	Stores received cookies in a file.
Example:	<code>curl -c cookies.txt https://example.com</code>

User Agent

<code>-A <string></code> or <code>--user-agent <string></code>	Sets the User-Agent header.
Example:	<code>curl -A "My Custom Agent" https://example.com</code>

Error Handling

<code>--fail</code>	Fail silently (no output at all) on HTTP errors.
Example:	<code>curl --fail https://example.com/nonexistent</code>
<code>--retry <num></code>	Retry request specified number of times if transient problems occur
Example:	<code>curl --retry 3 https://example.com</code>
<code>--max-redirs <num></code>	Maximum number of redirects allowed
Example:	<code>curl --max-redirs 5 https://example.com</code>