opensource.com

Curl Cheat Sheet

BY SETH KENLON

Curl is is a tool to transfer data to and from a server, supporting protocols including HTTP, FTP, IMAP, LDAP, POP3, SCP, SFTP, SMB, SMTP, and more.

Web browsing

Fetch HTTP headers

```
curl --head "https://example.com"
```

List contents of a directory

```
curl --list-only "https://example.com/foo/"
```

Redirect query as specified by a 3xx response

```
curl --location "https://iana.org"
```

Fail quickly

```
curl --fail-early "http://opensource.ga"
```

Check whether a site is down

```
curl --head --show-error "http://example.com"
```

Expand a shortened URL

```
curl --head --location "https://bit.ly/2yDyS4T"
```

Downloading files

Download a file, saving the file without changing its name

```
curl --remote-name "https://example.com/linux-distro.iso"
```

Rename the file

```
curl --remote-name "http://example.com/index.html" --output foo.html
```

Continue a partial download

```
curl --remote-name --continue-at - "https://example.com/linux-distro.iso"
```

Download a file from multiple domains

```
curl "https://www.{example,w3,iana}.org/index.html" --output "file_#1.html"
```

Download a sequence of files (outputs foo_file1.webp, foo_file2.webp...bar_file1_webp, etc)

```
curl "https://{foo,bar}.com/file_[1-4].webp" --output "#1_#2.webp"
```

Download a sequence of files (outputs foo_file1.webp, foo_file2.webp...bar_file1_webp, etc)

```
curl "https://{foo,bar}.com/file_[1-4].webp" --output "#1_#2.webp"
```

Downloading files (continued)

Download all PNG files from a site (uses GNU grep)

```
curl https://example.com | grep --only-matching 'src="[^"]*.[png]"' | \
cut -d\" -f2 | \
while read i; do curl https://example.com/"${i}" -o "${i##*/}"; done
```

API interaction

Query an API endpoint

```
curl "https://gitlab.com/api/v4/projects"
```

Send raw data to an API endpoint

```
curl --data "Some data" "https://example.com/api/v4/endpoint"
```

Send form data (emulates a form and Submit button)

```
curl --form "username=seth" --form "password=12345678" \
"https://example.com/api/v4/endpoint"
```

Send a file as form data (uses the @ syntax)

```
curl --form "profile=@me.jpg" "https://example.com/foo/bar"
```

Send contents of a file as form data (uses the < syntax)

```
curl --form "description=<file.md" "https://example.com/foo/bar"
```

Send HTTP header data (for example, an authorization token)

```
curl --header "Authorization: Bearer F66eD5ffXHp2Y" \
"https://example.com/api/v4/endpoint"
```

Specify HTTP Method

```
curl --request POST --data "Foo: bar" "https://example.com/api/endpoint"
```

Make an API call to a service

```
curl --request PUT --header "PRIVATE-TOKEN: your_access_token_here" \
--data "?per_page=10" "https://gitlab.com/api/v4/namespaces"
```