Curl cheatsheet

This Curl cheat sheet contains commands and examples of some common Curl tricks.

Getting Started

Introduction

Curl is a tool for transferring data between servers, supporting protocols, including:

HTTP	HTTPS	FTP
IMAP	LDAP	POP3
SCP	SFTP	SMB
SMTP	etc...	

- Curl GitHub source repository (github.com)
- Curl Official Website (curl.se)

Options

```
-o <file>    # --output: write to file
-u user:pass # --user: authentication
```

```
-v   # --verbose: Make curl verbose during operation
-vv  # more verbose
-s   # --silent: don't show progress meter or errors
-S   # --show-error: When used with --silent (-sS), show errors but no progress meter
```

```
-i  # --include: include HTTP headers in the output
-I  # --head: header only
```

Request

```
-X POST # --request
-L # If the page redirects, follow the lir
-F # --form: HTTP POST data for multipart/
```

data

```
# --data: HTTP post data
# URL encoding (eg, status="Hello")
-d 'data'

# --data pass file
-d @file

# --get: send -d data via get
-G
```

Header information Headers

```
-A <str>      # --user-agent

-b name=val   # --cookie

-b FILE       # --cookie
-H "X-Foo: y" # --header

--compressed  # use deflate/gzip
```

SSL

```
    --cacert <file>
    --capath <dir>

-E, --cert <cert> # --cert: client certifi
    --cert-type # der/pem/eng
-k, --insecure # For self-signed certifica
```

Install

```
apk add --update curl # install in alpine
```

Example

CURL GET/HEAD

`curl -I https://quickref.me`	curl sends a request
`curl -v -I https://quickref.me`	curl request with details
`curl -X GET https://quickref.me`	use explicit http method for curl
`curl --noproxy 127.0.0.1 http://www.stackoverflow.com`	curl without http proxy
`curl --connect-timeout 10 -I -k https://quickref.me`	curl has no timeout by default
`curl --verbose --header "Host: www.mytest.com:8182" quickref.me`	curl get extra header
`curl -k -v https://www.google.com`	curl get response with headers

Multiple file upload

```
$ curl -v -include \
--form key1=value1 \
    --form upload=@localfilename URL
```

Prettify json output for curl response

```
$ curl -XGET http://${elasticsearch_ip}:92
```

CURL POST

`curl -d "name=username&password=123456" <URL>`	curl send request
`curl <URL> -H "content-type: application/json" -d "{ \"woof\": \"bark\"}"`	curl sends json

CURL script install rvm

```
curl -sSL https://get.rvm.io | bash
```

CURL Advanced

`curl -L -s http://ipecho.net/plain, curl -L -s http://whatismijnip.nl`	get my public IP
`curl -u $username:$password http://repo.dennyzhang.com/README.txt`	curl with credentials
`curl -v -F key1=value1 -F upload=@localfilename <URL>`	curl upload
`curl -k -v --http2 https://www.google.com/`	use http2 curl
`curl -T cryptopp552.zip -u test:test ftp://10.32.99.187/`	curl ftp upload
`curl -u test:test ftp://10.32.99.187/cryptopp552.zip -o cryptopp552.zip`	curl ftp download
`curl -v -u admin:admin123 --upload-file package1.zip http://mysever:8081/dir/package1.zip`	upload with credentials curl

Check website response time

```
curl -s -w \
'\nLookup time:\t%{time_namelookup}\nConnect time:\t%{time_connect}\nAppCon time:\t%{time_
     -o /dev/null https://www.google.com
```

Use Curl to check if a remote resource is available

```
curl -o /dev/null --silent -Iw "%{http_coc
```

Downloading file

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

Download all PNG files from the site (using GNU grep)

Download the file, save the file without changing its name

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

rename file

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

continue partial download

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

Download files from multiple domains

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

Download a series of files

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

Download a series of files (output `foo_file1.webp`, `foo_file2.webp...bar_file1_webp`, etc.)

Redirect output to file

```
$ curl http://url/file > file
```

Basic Authentication

```
$ curl --user username:password http://example.com/
$ curl -u username:password http://example.com/
```

Write to file instead of stdout

```
$ curl -o file http://url/file
$ curl --output file http://url/file
```

Download header information

```
$ curl -I url
# display header i
```

Write output to a file named remote_file

```
$ curl -o file http://url/file
$ curl --output file http://url/file
```

Execute remote script

```
$ curl -s http://url/myscript.sh
```

Configuration file

```
curl -K file
# read configuration from file
curl --config file
$HOME/.curlrc # default configuration file
```


Top Cheatsheet

- **Python** Cheatsheet — Quick Reference
- **Vim** Cheatsheet — Quick Reference
- **JavaScript** Cheatsheet — Quick Reference
- **Bash** Cheatsheet — Quick Reference

Recent Cheatsheet

- **Google Search** Cheatshee — Quick Reference
- **Kubernetes** Cheatsheet — Quick Reference
- **ES6** Cheatsheet — Quick Reference
- **ASCII Code** Cheatsheet — Quick Reference

QuickRef.ME

Share quick reference and cheat sheet for developers.

中文版 #Notes