



Introduction to Bash video lecture

09 - Downloading files



You need to download a file

- Let's say your teacher tells you to download a script and run it
- You know that you want the script in a directory where you are working with scripts
- So, why not `cd` to that directory and download the file from the command line?
- This is much quicker than opening a browser and download to `~/Downloads/` and then move the file to the correct place
- There are a few commands for downloading files from a web server
- We'll look at them in this video

wget

```
$ wget "http://some.url/some/file"
```

```
$ wget "http://some.url/some/file" -O desired.name
```

```
# Use quotes around the URL to avoid shell expansion and
```

```
# problems with spaces and ampersands etc
```

curl

```
curl -JO "url-to-the-file"
```

```
  ^      ^      ^
  |      |      |
  |      |      +--> The URL goes here
  |      +-----> flags telling curl how it should work
  |                  (download only, use suggested filename)
+-----> the command "curl"
```

Anatomy of a URL

https://raw.githubusercontent.com/progund/datorkunskap-kompendium/master/school/courses/command_line/meals.txt

↑ ↑
https - Use the protocol HTTPS (secure HTTP)

↑
:// - comes after the protocol

↑
raw.githubusercontent.com - the domain name - identifies the computer

↑
/progund/datorkunskap-kompendium/master/school/courses/command_line/ - path

↑
meals.txt - the resource (filename)

Server - confusion

- Can mean a powerful computer connected to a network
- Can mean an application running on a computer
- “The web server is running on our server in the server room”

HTTP crash course

- Protocol used on the web
- A *client* requests some resource from a *server*
- The client uses an HTTP *method* like GET
GET /pictures/henrik.png HTTP/1.1
- The client sends along some *headers* with information about the client
- The server responds with headers and then the requested resource

Example response headers

```
200 OK
Cache-Control: max-age=300
Connection: close
Date: Wed, 24 Jul 2019 13:10:34 GMT
Via: 1.1 varnish
Accept-Ranges: bytes
Vary: Authorization,Accept-Encoding
Content-Length: 274
Content-Type: text/plain; charset=utf-8
Expires: Wed, 24 Jul 2019 13:15:34 GMT
Client-Date: Wed, 24 Jul 2019 13:10:34 GMT
Client-Peer: 151.101.84.133:443
```

here comes the actual file....

Typical clients for HTTP

- Browsers
- wget
- curl
- Android (your apps use some library to fetch data and files over HTTP)
- iOS (same)