curl & wget

A fake ReST api JSON test site :

https://jsonplaceholder.typicode.com

---

weget 'url' just created a new file itself
To force wget to send output to desired file by telling wget to output its payload to stdout (with flag -O-) and supress its own output (with flag -q):

```
wget -qO- 'http://api.openweathermap.org/data/2.5/weather?q=Ellesmere
Port,uk&appid=4595334ecc0bf10a9aa1461cc222ef14' >> wgettest-qO-.json
```

---

try

```
wget -qO- 'http://api.openweathermap.org/data/2.5/weather?q=Ellesmere
Port,uk&appid=4595334ecc0bf10a9aa1461cc222ef14' >> wgettest2-qO-.json | python -m
json.tool >> wgettest2-qO-Prettyjson.json
```

Curl and libcurl

query url,
GET POSt PUT DELETE
auth users ,
save responces to file.
Transfer cookies, test speeds etc

Good for testing ReST api's

$ curl http:// xxx   or say $ curl http://localhost:3000
-> gives script from browser
Testing JSON responce
$ curl http://xxxx   will give json pacge back -> BUT !

cURL info
https://curl.haxx.se/docs/httpscripting.html

To get responce header 'i'nformation with the JSON package use:
$ curl -i http://xxxx
and for just the 'head'er 'I'nformation, ie on its own
$ curl -I http://xxxxx   or $ curl --head http://xxxxx

To GET , POST, PUT (update), DELETE ie http methods

GET

```
$ curl http://    is itself a GET
```

POST sending '**d**'ata (e.g. a first and last name to an api called xxxxx)

```
$ curl -d "first=Tom&last=Ormiston" http:// xxxxx
```

example 2

```
$ curl -d "title=myHello&body=hello world" https://jsonplaceholder.typicode.com/posts
```

PUT (update)

we 'request a command ' (-X) to PUT 'd'ata (PUT -d)

```
$ curl -X PUT -d "first=Thomas&last=Ormiston" http:// xxxxx
```

example 2

```
curl -X PUT -d "title=myHello&body=hello world"
https://jsonplaceholder.typicode.com/posts/99
```

DELETE

we 'request a command ' (-X) to DELETE

```
$ curl -X DELETE http://xxxxxx
```

example 2

```
 curl -X DELETE https://jsonplaceholder.typicode.com/posts/99
```

For a'u'thentication -> use 'u'

for   username: bob   password:123

```
$ curl -u bob:123  http://xxxxx
```

D'o'wnload a file/page - the 'o'utput (also format is prettyfied json)

```
$ curl -o pic1.jpg http://xxxxx
```

this image data is downloaded and the output saved in a file automaticly created called pic1.jpg

Save the output of a JSON packge to a file

```
$ curl -o jsfile.json http://xxxxx
```

or example 2 using the fake json rest api tester site:

```
$ curl -o dowloadJson.json https://jsonplaceholder.typicode.com/posts/6
```

Downloading not a GET request as such use capital 'O' ie -O,
this doesn't need a filename as it come from the remote url

```
$ curl -O http://xxxx
```

eg, this just downloads a file called 'posts'

```
$ curl -O https://jsonplaceholder.typicode.com/posts
```

This is good for downloading images too

```
$ curl -O http://bit.ly/2v4Dx8t
```

```
'  -O  ' remote name  ---> Write output to a file named as the remote file
'  -o  '  output FILE  ---> Write to FILE instead of stdout
```

RE-DIREDTION   -->   'L'

if we do

```
$ curl http://google.com
```

```
--->
<TITLE>302 Moved</TITLE></HEAD><BODY>
<H1>302 Moved</H1>
The document has moved
<A HREF="http://www.google.co.uk/?
gfe_rd=cr&ei=YYSVWbqSFa338Aew3Z_oBw">here</A>.
</BODY></HTML>
```

The reason is that the site has moved to www.google.com
Hence we use a re-direct option ' -L '

```
$ curl -L http://google.com
```

now works ok because to redirects to www.google.com automaticly

File ' T 'ransfersing using FTP   ---> ' T ' flag

This needs authentication so use the -u
so for this example
username: tom@appijumbo.com   password:1234 file is 'myfile.txt'

To upload myfile.txt

```
$ curl -u tom@appijumbo.com:1234  -T myfile.txt ftp://ftp.appijumbo.com
```

To download myfile.txt

```
$ curl -u tom@appijumbo.com:1234  -O ftp://ftp.appijumbo.com/myfile.txt
```

To install Yarn

```
curl -sS https://dl.yarnpkg.com/debian/pubkey.gpg | sudo apt-key add -
echo "deb https://dl.yarnpkg.com/debian/ stable main" | sudo tee
/etc/apt/sources.list.d/yarn.list
```