

Comando Curl

1. Lectura y guardad de URL's

- a. Lectura simple de una página web

```
# curl http://www.google.es
```

- b. Guardado de un página web en un fichero. Utilizaremos la opción -o seguida del nombre del fichero con el que guardaremos el recurso:

```
# curl -o prueba.html http://www.google.es
```

- c. Utilizar autenticación HTTP:

```
# curl -y user:pass https://www.secure-site.com
```

- d. Podemos ver la respuesta completa a una petición, que incluye una descripción tanto de la entrada como de la salida de datos del servicio. Esto lo haremos mediante la opción:

```
# curl -trace-ascii nacho.txt http://www.google.com
```

2. Peticiones GET

- a. Se puede hacer petición a un rango de páginas, del mismo dominio, gracias a las expresiones regulares. En el siguiente ejemplo, se van a guardar las páginas de la 1 a 12 en sus correspondientes ficheros pages[1-12].html:

```
# curl -o pages#1.html http://ejemplo.com/pages.php?id=[1-12]
```

- b. Si el servicio RESTful tiene reglas de reescritura para sus URL's, podemos hacer peticiones GET sin usar parámetros en la URL:

```
# curl http://localhost/login_restful/Api.php?peticion=usuarios //con parametros en la url
```

```
# curl http://localhost/login_restful/usuarios //gracias a las reglas de reescritura
```

3. Peticiones POST

- a. Podemos simular el envío de datos (simulando formulario) con dos opciones. Con la opción -d indicaremos el envío de variables y valores y con -F indicaremos la ruta de un fichero a subir.

- Parámetros simples:

```
# curl -d "campo1=valor_campo1&campo2=valor_campo2 ..." url_a_donde_hacemos_el_post.com
```

- Subida de ficheros

```
# curl -F "nombre_del_input=@path_al_file_a_subir" url_a_donde_hacemos_el_post.com
```

- Envío de datos y ficheros

Si queremos hacer una petición en la que se envíen tanto datos como un fichero. Podríamos pensar en una combinación de las opciones -d y -F. Pero eso ocasionaría un error ya que tienen diferentes content-types. Mientras que -F usa multipart/form-data para el envío Post; -d usa application/x-www-form-urlencoded. La solución es usar la opción -F varias veces.

```
# curl -i -F "name=test" -F "filedata=@localfile.jpg" http://www.ejemplo.org/upload
```

4. Peticiones PUT

- a. PUT y DELETE son considerados 'custom-method', al contrario que GET y POST. Por lo tanto tenemos que indicarlo a la herramienta cURL con la opción -X:

```
# curl -X PUT -d "telefono=1-800-999-9999" http://www.servidor.com/personas/persona/1
```

5. Peticiones DELETE

```
# curl -X DELETE http://www.servidor.com/personas/persona/1  
# curl -X DELETE http://www.servidor.com/personas.php?id=1  
# curl -X DELETE -d "id=1" http://www.servidor.com/borrarUsuario
```

6. Lectura de información

- a. Podemos mostrar únicamente las cabeceras de la respuesta y no la respuesta e sí. En esta ocasión no utilizaremos la opción -i, vista anteriormente, ya que mostraba ambas cosas.

```
# curl --head http://www.google.com/
```

7. FTP

- a. Listar un directorio de un ftp:

```
# curl ftp://usuario:pass@ejemplo.com/directory/
```

Fuente: [Curl MASTER](#)