"El arte de crear secuencias de comandos de solicitudes HTTP usando Curl" por Daniel Stenberg

Traducción del idioma inglés publicado por el señor Daniel Stenberg, título original "*The Art Of Scripting HTTP Requests Using Curl*" en:

`https://curl.se/docs/httpscripting.html`

Antesala

Este documento asume que está familiarizado con HTML y redes en general.

La creciente cantidad de aplicaciones que se trasladan a la web ha hecho que "*HTTP Scripting*" (guiones para HTML) sea ahora más solicitado y buscado. Poder extraer automáticamente información de la web, falsificar usuarios (*sic* n. del t.), publicar o cargar datos en servidores web son tareas importantes hoy en día.

Curl es una herramienta de línea de comandos para realizar todo tipo de manipulaciones y transferencias de URL, pero este documento en particular se centrará en cómo usarlo cuando se realizan solicitudes HTTP por diversión y ganancias. Asumiré que sabe cómo invocar curl --help o curl --manual para obtener información básica al respecto.

Curl no está escrito para hacer todo por usted. Hace las solicitudes, obtiene los datos, envía datos y recupera la información. Probablemente necesite unir todo utilizando algún tipo de lenguaje de *script* o o realizando invocaciones manuales repetidas.

El protocolo HTT (HTTP)

HTT es el protocolo utilizado para obtener datos de los servidores web. Es un protocolo simple que se basa en TCP/IP. El protocolo también permite enviar información al servidor desde el cliente utilizando algunos métodos diferentes, como se mostrará aquí.

HTTP son líneas de texto ASCII sin formato (*HTTP anteriores a HTTP 2, n. del t.*) que el cliente envía a un servidor para solicitar una acción en particular, y luego el servidor responde algunas líneas de texto antes de que el contenido real solicitado se envíe al cliente.

El cliente, **curl**, envía una solicitud HTTP. La solicitud contiene un método (como GET, POST, HEAD, etc.), varios encabezados de solicitud y, a veces, un cuerpo de solicitud. El servidor HTTP responde con una línea de estado (que indica si todo salió bien), encabezados de respuesta y, en la mayoría de los casos, también un cuerpo de respuesta. La parte del "cuerpo" son los datos sin formato que solicitó, como el HTML real o la imagen, etc.

Vea el protocolo en acción

El uso de la opción de curl --verbose (-v como opción abreviada) mostrará qué tipo de comandos envía **curl** al servidor, así como algunos otros textos informativos.

--verbose es la opción más útil cuando se trata de depurar o incluso comprender la interacción del servidor curl.

A veces incluso --verbose no es suficiente. Entonces, para eso, --trace y --trace-ascii ofrecen aún más detalles, ya que muestran todo lo que **curl** envía y recibe. Úselo así:

```
curl --trace-ascii debugdump.txt http://www.example.com/
```

Cronometre la consulta

Muchas veces se preguntará cuánto tiempo exactamente toma la consulta, o simplemente quiere saber la cantidad de milisegundos entre dos puntos en una transferencia. Para esas y otras situaciones similares, la opción --trace-time es lo que necesita. Antepondrá la hora a cada línea de salida de seguimiento:

```
curl --trace-ascii d.txt --trace-time http://example.com/
```

curl --trace-ascii d.txt --trace-time

Observe la consulta

Por defecto, **curl** envía la respuesta a stdout. Debe redirigirlo a algún lugar para evitar eso, la mayoría de las veces se hace con -o o -O.

URL

Especificaciones

El formato del localizador uniforme de recursos (formato URL) es la forma en que especifica la dirección de un recurso en particular en Internet. Usted los conoce, usted los ha visto URL como, por ejemplo, https://curl.se o https://subanco.com.ve un millón de veces. **RFC 3986** es la especificación canónica. Y sí, el nombre formal no es URL, es **URI**.

Anfitrión

El nombre del anfitrión (*host*) generalmente se resuelve usando DNS o su archivo /etc/hosts a una dirección IP y eso es con lo que **curl** se comunicará. Alternativamente, especifique la dirección IP directamente en la URL en lugar de un nombre.

Para el desarrollo y otras situaciones de prueba, puede señalar una dirección IP diferente para un nombre de *host* que la que se usaría de otro modo, utilizando la opción --resolve de curl:

```
curl --resolve www.example.org:80:127.0.0.1 http://www.example.org/
```

Número de puerto

Cada protocolo que admite **curl** opera en un número de puerto predeterminado, ya sea a través de TCP o, en algunos casos, UDP. Normalmente no tiene que tener eso en cuenta, pero a veces puede necesitar ejecutar en servidores de prueba con otros puertos o similares. Luego puede especificar el número de puerto en la URL con dos puntos y un número inmediatamente después del nombre de *host*. Como cuando se hace HTTP al puerto 1234:

```
curl http://www.example.org:1234/
```

El número de puerto que especifique en la URL es el número que utiliza el servidor para ofrecer sus servicios. A veces puede usar un *proxy*, y luego puede necesitar especificar el número de puerto de ese *proxy* por separado de lo que curl necesita para conectarse al servidor. Como cuando se usa un *proxy* HTTP en el puerto 4321:

```
curl --proxy http://proxy.example.com:4321 http://remoto.example.org/
```