

Protocolo de monitoreo

En este documento describimos el protocolo de monitoreo que debe ser utilizado para consultar y modificar el estado del servidor. El protocolo esta inspirado en POP3 ([RFC 1939](#)).

1- Operaciones básicas.

Inicialmente, el servidor da inicio al sistema de monitoreo escuchando conexiones al puerto TCP 8080. Cuando un cliente desee conectarse, debe establecer una conexión TCP a dicho puerto. El servidor le enviará un saludo, y luego intercambiarán comandos hasta que la conexión se cierre o aborte.

Los comandos de este protocolo consisten de una palabra clave (printable ASCII, case-insensitive y sin espacios) seguida de los argumentos que fueran requeridos para ese comando (como máximo 40 caracteres por argumento, no necesariamente case-insensitive, printable ASCII). Los comandos terminan con un CRLF (\r\n).

Las respuestas a los comandos consisten de un indicador de estado, posiblemente seguidos de más información y terminados por CRLF. Estas respuestas tendrán como máximo 512 bytes de longitud. Hay dos estados posibles: +OK y -ERR, que deberán ser enviados en mayúscula. En general, lo que se envía después del +OK o -ERR es una descripción de que sucedió, que puede depender de la implementación.

Ciertos comandos pueden enviar más de una línea de información, la línea final será aquella que contenga únicamente un octeto de terminación (0x46, “.”). Es decir, el final de la respuesta para los comandos multilinea es la secuencia “CRLF.CRLF”. Este octeto de terminación no se considerará parte de la respuesta. **Solamente las respuestas exitosas pueden ser multilinea.**

Una sesión abierta de este protocolo consiste de dos estados distintos. El primero es el estado de AUTENTICACIÓN. En este estado el cliente debe autenticarse antes de poder hacer ninguna otra operación (con excepción de la acción de cierre). Una vez autenticado, se pasa a un estado de TRANSACCIÓN, en el cual se le permite al cliente ejecutar el resto de los comandos.

2- Estado de AUTENTICACIÓN

Una vez que se inicia la conexión y el servidor envía al cliente el saludo (por ejemplo “+OK Monitor server ready”) es responsabilidad del cliente autenticarse. Se debe soportar autenticación mediante usuario y contraseña.

A continuación se listan los comandos válidos para el estado de AUTENTICACIÓN:

TOKEN <token>

Descripción: el servidor determina si el token ingresada es igual a la que tiene registrada en su base de datos, y si lo es retorna en forma exitosa. El retorno exitoso de este comando pasa a la sesión al estado de TRANSACCIÓN.

Restricciones: puede ser utilizado únicamente en el estado de AUTENTICACIÓN, luego de un llamado exitoso a USER.

Ejemplos:

C: PASS password

S: +OK

C: PASS not_password

S: -ERR invalid password for user juan

C: PASS password

S: -ERR no registered user

3- Estado de TRANSACCIÓN

Una vez autenticado, el cliente puede acceder a la información del servidor.

A continuación se listan los comandos válidos para el estado de TRANSACCIÓN:

USERS

Descripción: lista los usuarios del sistema. No tiene argumentos y la respuesta es multilínea. De forma de que sea acorde al protocolo, los usernames no pueden contener caracteres no printable, y se define un máximo de 20 usuarios con usernames de como mucho 20 caracteres de largo de forma que la respuesta entre en los 512 caracteres máximos de la respuesta.

Ejemplos:

C: USERS

S: +OK listing users:

juan

mateo

Joaquin

.

STATS

Descripción: muestra las estadísticas actuales del servidor. Es multilínea y envía la cantidad de conexiones históricas, concurrentes, el total de bytes transferidos y cualquier otra información que el servidor considere pertinente.

Ejemplos:

C: STATS

S: +OK

B 324

H 23

C 2453

.

DISSECTOR-STATUS

Descripción: muestra el estado (prendido-apagado) del disector de contraseñas.

Ejemplos:

C: DISSECTOR-STATUS

S: +OK

ON

.

C: DISSECTOR-STATUS

S: +OK

OFF

.

BUFFSIZE

Descripción: muestra el tamaño del buffer en bytes.

Ejemplos:

C: BUFFSIZE

S: +OK

2048

.

SET-DISSECTOR-STATUS <status>

Descripción: prende o apaga el disector de contraseñas. Si <status> tiene algún valor distinto de “on” u “off”, arroja un error.

Ejemplos:

C: SET-DISSECTOR-STATUS on

S: +OK dissector is now turned on

C: SET-DISSECTOR-STATUS what

S: -ERR invalid status for dissector

SET-BUFFSIZE <bytes>

Descripción: setea el tamaño del buffer a <bytes>. Si bytes no es un entero positivo, arroja error.

Ejemplos:

C: SET-BUFFSIZE 1024

S: +OK buffer size changed to 1024 bytes

C: SET-BUFFSIZE what

S: -ERR invalid buffer size

ADD-USER <userpass>

Descripción: agrega un usuario al sistema, donde userpass es un string de la forma user:pass. El nombre de usuario y contraseña deben cumplir los mismos requisitos que los definidos anteriormente en el comando users.

Ejemplos:

C: ADD-USER juan:pass

S: +OK added user juan

C: ADD-USER juan:

S: -ERR error adding user juan

4- Comentarios finales

A continuación listamos los comandos disponibles en cualquiera de los estados:

CAPA

Descripción: muestra los comandos disponibles en el servidor.

Ejemplos:

C: CAPA

S: +OK

USERS

STATS

BUFFSIZE

SET-BUFFSIZE

DISSECTOR-STATUS
SET-DISSECTOR-STATUS

.

QUIT

Descripción: al recibir el comando, el servidor se encarga de cerrar la conexión con el cliente.

Ejemplos:

C: CAPA
S: +OK
USERS
STATS
BUFFSIZE
SET-BUFFSIZE
DISSECTOR-STATUS
SET-DISSECTOR-STATUS

.

Si desde cualquier estado se ejecuta un comando no conocido, se devolverá un estado de error y un comentario acorde. Por último, si se pasan argumentos a un comando que no los requiere o el formato de los argumentos no es el requerido siempre se retorna un estado de error.