



ITEM 07 – JMETER CON CSV

María Victoria Calbet González
Marta Ramírez González
David Romero Esparraga
Jesús Ortiz Calleja
Guillermo Alcalá Gamero
Juan Carlos Utrilla Martín

Contenido

1.	Introducción	3
2.	Generar archivo CSV	4
3.	Configuración de Apache JMeter	4
3.1.	Adición de CSV Data Set Config	4
3.2.	Configuración de CSV Data Set Config:	5
3.3.	Definir valores de los parámetros en la petición	6
4.	Ejecutamos el test y comprobamos que funciona	7
5.	Fuentes	9

1. Introducción

En esta entrega de la asignatura, el nuevo concepto con el que estamos trabajando son las pruebas de rendimiento. Estas pruebas permiten ejecutar el sistema con una carga de trabajo similar a la real. Consisten en reproducir un caso de uso determinado por usuarios ficticios con el fin de estudiar los cuellos de botella que presenta el sistema. La reproducción de los casos de uso se hace mediante la ejecución de una serie de instrucciones previamente registradas por el usuario que está probando la aplicación.

El problema presente con estas pruebas es que los datos registrados son siempre los mismos, ya que se ejecuta el mismo caso de uso un gran número de veces con los mismos datos que se introdujeron durante el registro.

Para evitar este comportamiento, en el A+ se esta entrega, se nos plantea la opción de usar archivos CSV para definir los valores que queremos que se utilicen en las distintas iteraciones del caso de uso.



Estos archivos CSV, del inglés Comma-Separated Values, son un tipo de archivos en texto plano donde los valores están separados por comas. Cada línea del archivo corresponde a una entidad diferente y cada valor, un atributo de esa entidad. En nuestro sistema utilizaremos esta organización de los datos para definir cada una de las variables que estarían presentes en los formularios web. De esta manera, cada iteración hará uso de datos diferentes.

Ahora, en vez de reutilizar siempre los mismos valores, podemos definir qué datos son los que queremos que se usen en este tipo de pruebas.

Los mayores beneficios de este tipo de estrategias son la presencia de datos reales en el sistema y el control sobre qué datos se introducen. Al tener los datos en un archivo ajeno a las pruebas, podemos introducir valores que cubran numerosos casos de uso.

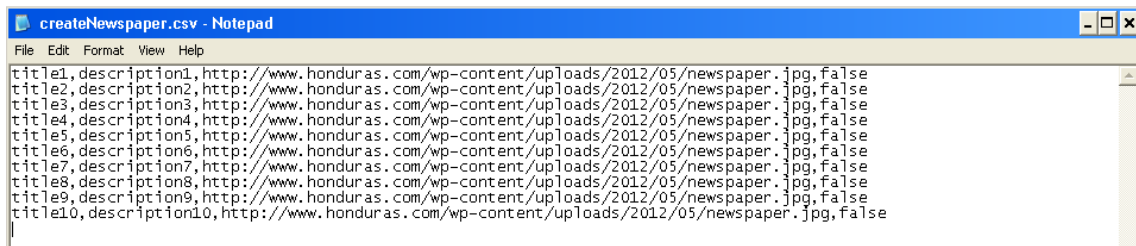
Un ejemplo sería el siguiente: Si hay una entidad que puede estar categorizada por un enumerado, lo normal es que no todas las entidades presenten el mismo valor. Un comportamiento de este estilo provocaría que algunos filtros no pudiesen ser probados, lo que afectaría a nuestro conocimiento sobre qué cantidad de elementos se consultarían en la base de datos en determinadas circunstancias.

2. Generar archivo CSV

Tal y como hemos estudiado en la lección 10, “Performance Testing”, necesitaremos tener una captura en JMeter de las peticiones necesarias para llevar a cabo la creación de un periódico. Necesitaremos hacer login, pulsar en “Create a newspaper”, rellenar el formulario que nos aparece, y enviarlo.

Un archivo CSV es un documento que contiene datos separados por comas.

Vamos a probar el caso de uso de crear un periódico. Necesitaremos añadir todos los datos que nos piden en el formulario, es decir, el título, descripción, imagen y la privacidad del periódico.

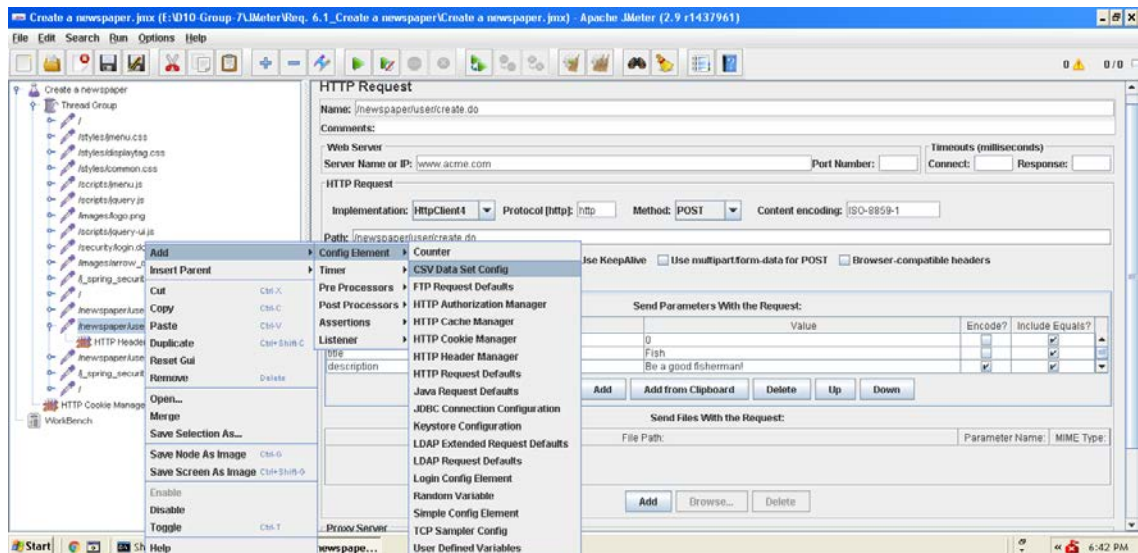


3. Configuración de Apache JMeter

Para poder implementar este archivo CSV en Apache JMeter, hay que realizar ciertas configuraciones en Apache JMeter.

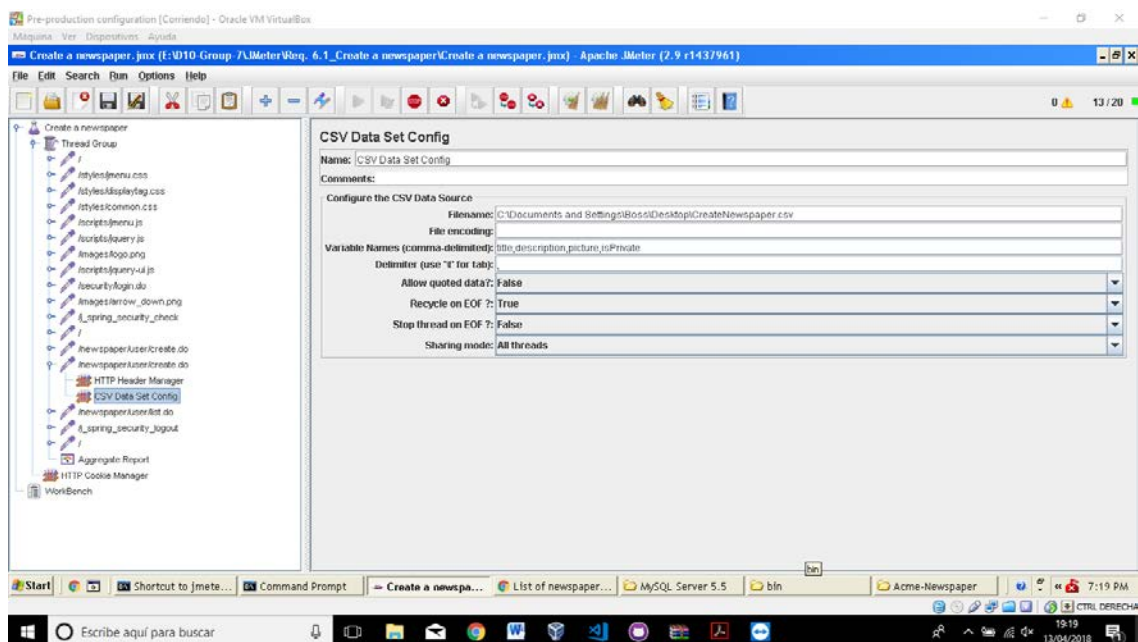
3.1. Adición de CSV Data Set Config

Lo primero que tenemos que hacer es añadir un CSV Data Set Config, para ello, hacemos click derecho en la petición que envía el formulario, en nuestro caso newspaper/create.do, pulsamos en add/Config Element/ CSV Data Set Config



3.2. Configuración de CSV Data Set Config:

Vamos a configurar el CSV Data Set Config para que la petición “/newspaper/user/create.do” utilice como datos el archivo csv.

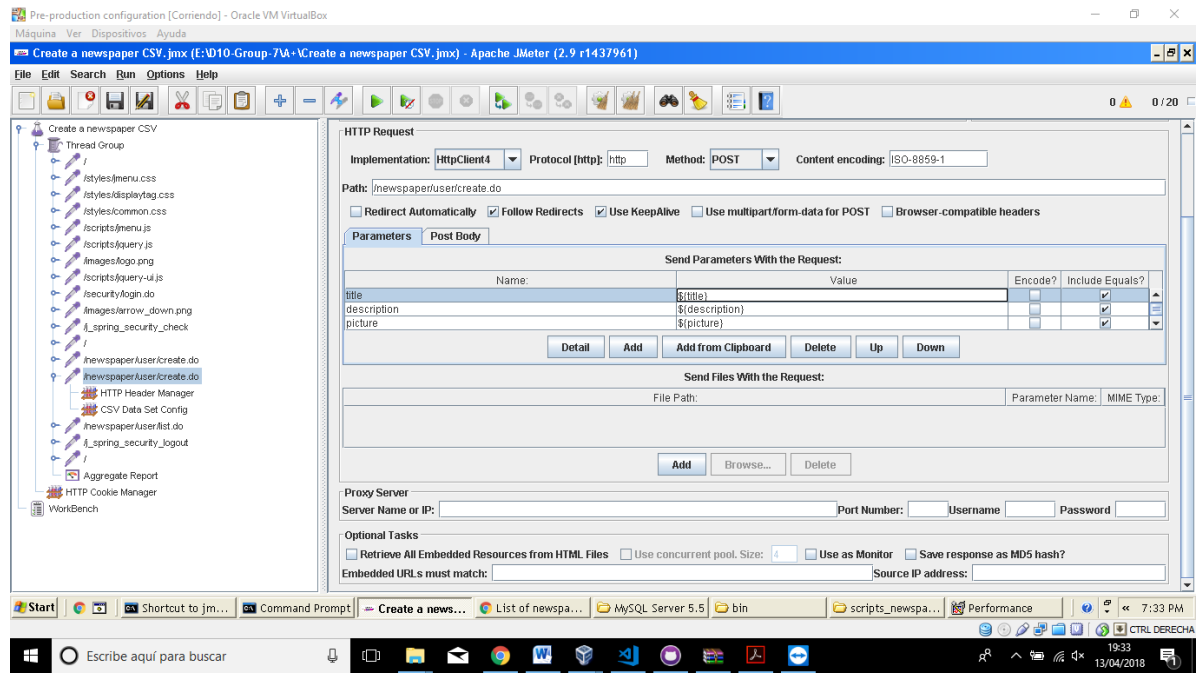


- **Paso 1:** en el apartado *Filename* introducimos la URL, donde se encuentra el archivo .csv que hemos creado previamente.
- **Paso 2:** en el apartado *Variable Names*, tenemos que añadir los nombres de las variables del documento. Éstos deben estar separados por comas.
- **Paso 3:** en el apartado “*Allow quoted data?*”: no permitiremos que se introduzcan datos entre comillas.
- **Paso 4:** el apartado “*Recycle on EOF?*”: lo establecemos a *True*
- **Paso 5:** el apartado “*Stop thread on EOF?*”: lo establecemos a *False*
- **Paso 6:** el apartado “*Sharing mode*”: marcamos *All threads*

3.3. Definir valores de los parámetros en la petición

Ahora tenemos que decirle a Apache JMeter que los valores que va a recibir la petición /newspaper/user/create.do son variables, ya que hemos creado un csv con valores distintos para cada caso.

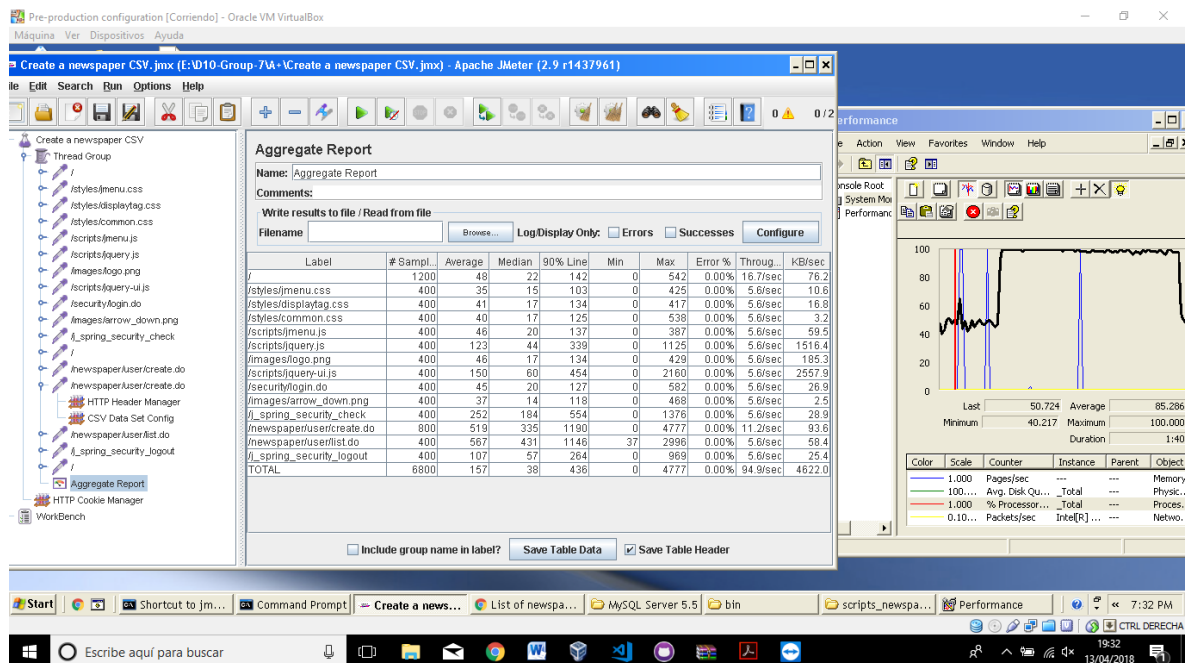
Para ello, tenemos que añadir \${Nombre de la variable} en la columna de Value.



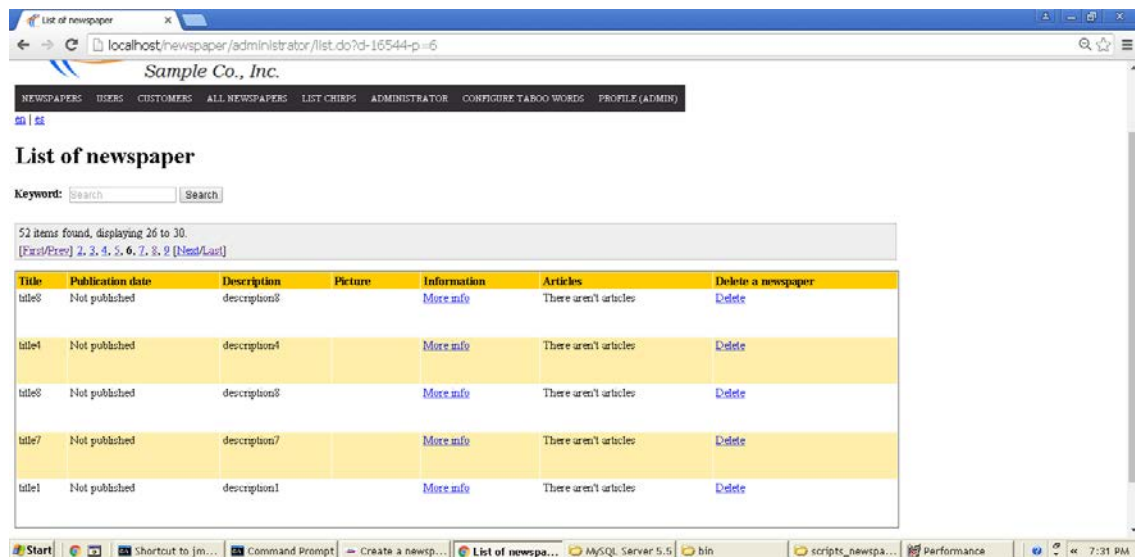
4. Ejecutamos el test y comprobamos que funciona

Configuramos el Thread Group tal y como hemos hecho en los tests de rendimiento de nuestro proyecto. En este caso, hemos ejecutado los tests para 20 usuarios y 20 ciclos. El computador en el que se van a realizar las pruebas tiene las siguientes características:

- Modelo del procesador: Intel(R) Core(TM) i5-2450M CPU @ 2.50GHz 2.50GHz
- Memoria RAM Virtual: 1536MB
- Disco duro: KINGSTON_SUV400S 250GB SSD



Ejecutamos nuestra aplicación, y comprobamos que efectivamente se están creando periódicos con los atributos que añadimos en el archivo .csv



Nota: El campo picture no aparece, ya que la captura se hizo con el servidor proxy parado, y no disponía de conexión a internet.

5. Fuentes

- **Wikipedia - CSV: Valores Separados por comas:**
https://es.wikipedia.org/wiki/Valores_separados_por_comas
- **Guía para test de carga y estrés:**
<https://gist.github.com/Virako/c1a8132fdddc4c868378>
- **Using CSV DATA SET CONFIG:** <https://guide.blazemeter.com/hc/en-us/articles/206733689-Using-CSV-DATA-SET-CONFIG>