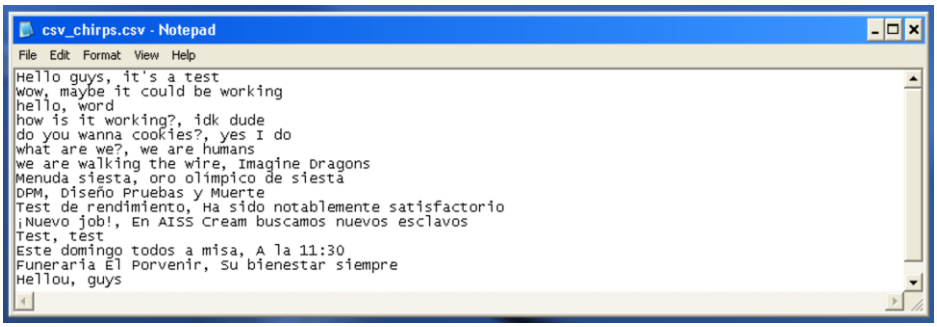
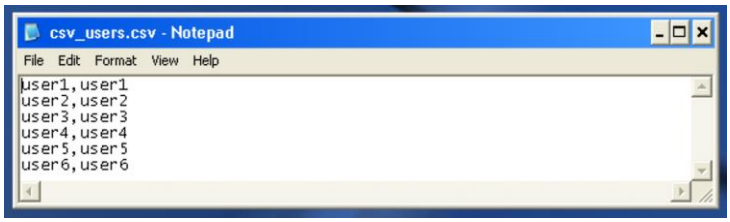
D10 – A+

Parameterization en JMeter

En este informe se va a proceder a explicar como usar una de las funcionalidades de JMeter llamada Parameterization que nos permitirá introducir múltiples conjuntos de datos en los formularios para, entre otras cosas, realizar el test de la forma más real posible. Estos conjuntos de datos estarán almacenados como CSV en un archivo de extensión csv, de forma que los distintos parámetros estén separados o delimitados por una coma:



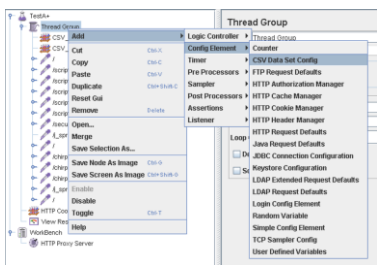


El caso de uso que vamos a tratar para explicar la *parameterization* consiste en autenticar distintos usuarios (csv_users.csv) y que cada uno cree distintos chirps (csv_chirps.csv).

La estructura del *csv_users* será la primera columna el nombre de usuario y la segunda será la contraseña y en el *csv_chirps* la primera columna se referirá al título del chirp y la segunda a la descripción de este.

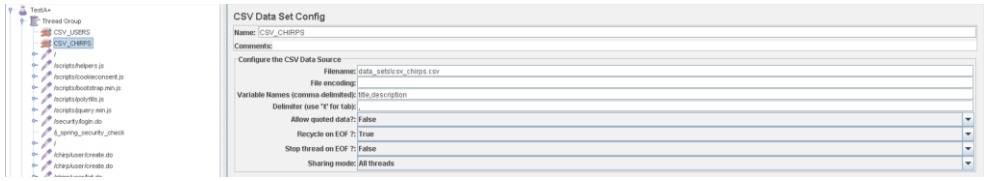
Partimos la captura que hemos hecho previamente en el test de *CreateChirpTest*, y en la cual los datos siempre son los mismos. Eso es precisamente es lo que vamos a cambiar, que los datos vayan cambiando utilizando el CSV.

Para añadir datos, hacemos clic derecho en *Thread Group*, luego vamos a *Add* > Config *Element* > CSV Data Set Config, tal y como aparece en la siguiente imagen:

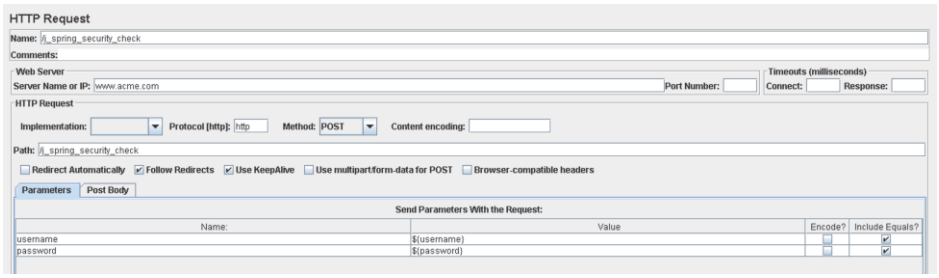


Una vez se haya creado el nuevo CSV Data Set Config, debemos configurar los parámetros *Filename* y *Variable names*. En la primera indicamos la ruta en la que se encuentra el archivo csv tomando com raíz la carpeta **bin**. Para realizar este ejemplo nosotros hemos colocado los *data sets* en una carpeta llamada data_sets que se encuentra dentro de bin, por ende, el valor de Filename para cargar csv_users.csv será: data_sets\csv_users.csv. El segundo parámetro a configurar será en el que indiquemos cuál será el nombre de cada una de las columnas de nuestro data set, para así poder referirnos a dichos valores en las HTTP Request, en el caso de csv_users.csv, hemos rellenado este parámetro con el siguiente valor: username,password. El resto de los parámetros los dejamos por defecto, aunque pueden ser cambiados si utilizamos una codificación de archiva o un delimitador distinto, en nuestro caso mantenemos dichos valores.

Tras cargar los datos de csv_users.csv, procedemos a hacer lo mismo con el otro data set: csv_chirps.csv, cuyos parámetros de configuración quedarán de la siguiente forma:

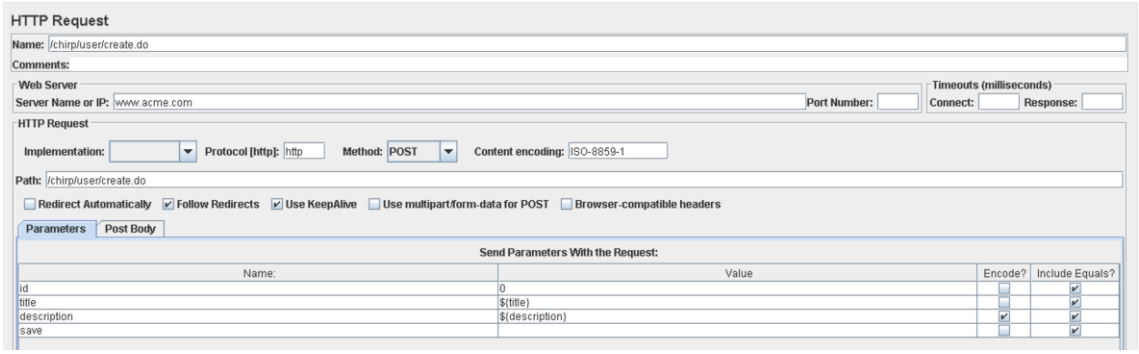


Una vez cargados los sets de datos, pasamos a incorporar las variables que hemos creado (username, password, title y descripción) en las peticiones HTTP de tipo POST que son las que se envían desde los formularios.

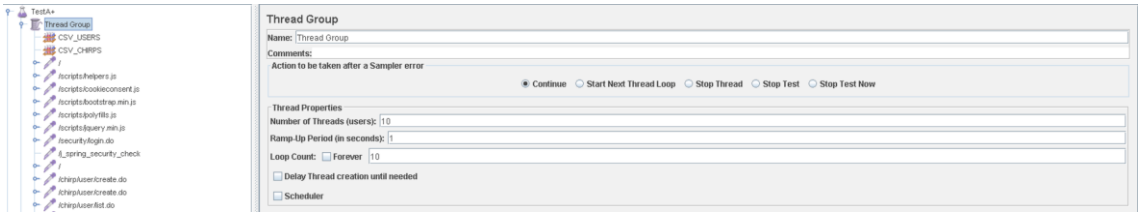


Seleccionamos la petición "/j_spring_security_check". Dentro de los parámetros que se envían en esta, debemos cambiar el campo del valor por las variables que definíamos en los CSV Data Set Config. Para hacer referencia a dichas variables debemos usar el siguiente formato ${nombreVariable}, en nuestro caso en el valor del campo username escribiremos ${username} y en password ${password}.

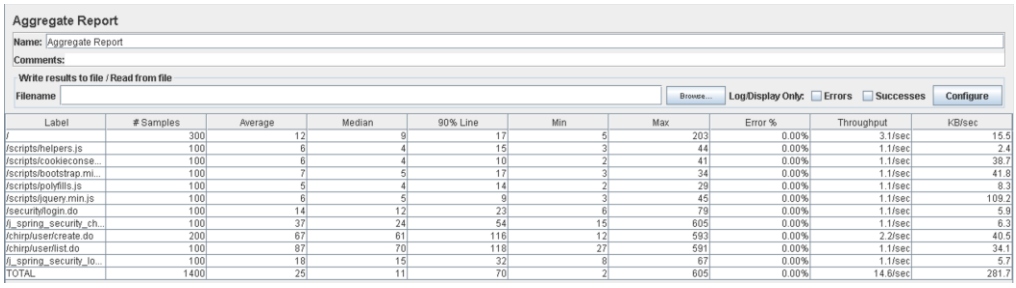
Seguiremos por la petición POST "/chirp/user/create.do", en la cual cambiaremos sus parámetros por los nombres de variable correspondientes que definimos por cada columna al registrar el data_chirps.csv en el CSV Data Set Config.



Una vez configurado todo, terminamos de indicar en el *Thread Group* los campos *Number of Threads* y *Loop Count*, para este caso vamos a hacer una pequeña prueba:



E iniciamos el test de rendimiento, que da los siguientes resultados:



Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/	300	12	9	17	5	203	0.00%	3.1/sec	15.5
/scripts/helpers.js	100	6	4	15	3	44	0.00%	1.1/sec	2.4
/scripts/cookieconse...	100	6	4	10	2	41	0.00%	1.1/sec	38.7
/scripts/bootstrap.mi...	100	7	5	17	3	34	0.00%	1.1/sec	41.8
/scripts/polyfills.js	100	5	4	14	2	29	0.00%	1.1/sec	8.3
/scripts/jquery.min.js	100	6	5	9	3	45	0.00%	1.1/sec	109.2
/security/login.do	100	14	12	23	6	79	0.00%	1.1/sec	5.9
/j_spring_security_ch...	100	37	24	54	15	605	0.00%	1.1/sec	6.3
/chirp/user/create.do	200	67	61	116	12	593	0.00%	2.2/sec	40.5
/chirp/user/list.do	100	87	70	118	27	591	0.00%	1.1/sec	34.1
/j_spring_security_lo...	100	18	15	32	8	67	0.00%	1.1/sec	5.7
TOTAL	1400	25	11	70	2	605	0.00%	14.6/sec	281.7