CSV to fill forms

Grupo 33
Víctor Rosa
Martín Ordoño

Table of Contents

1. Introduction

JMeter is an open source load testing tool which has been highly used in testing the performance of any software program or website. This helps us to write load tests using different types of test elements like thread group, samplers, config elements and controllers etc.

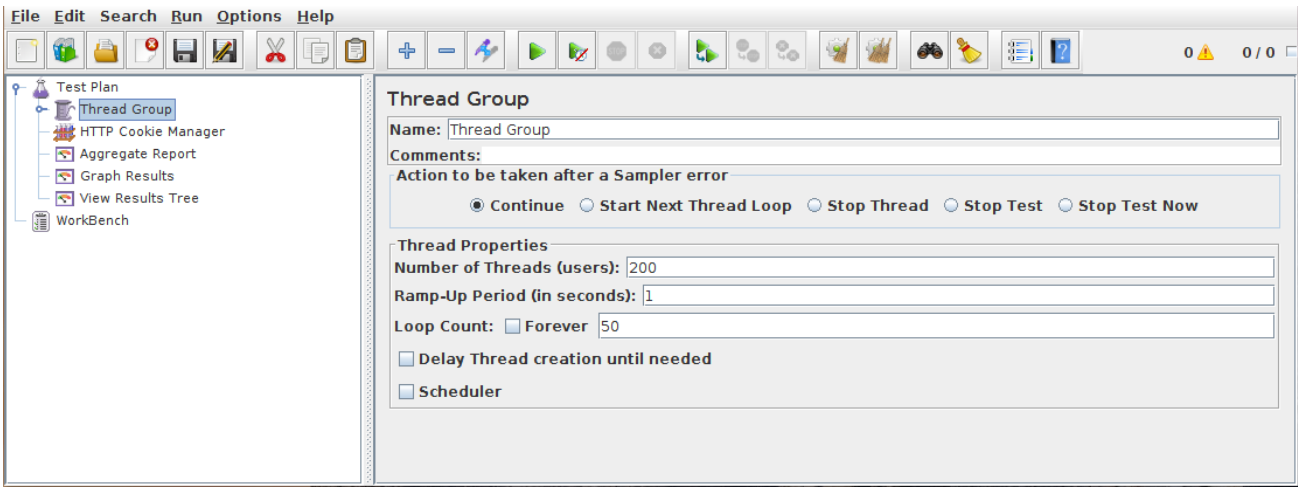
While writing automated load tests using JMeter it might be required to load data from a CSV file. JMeter has a CSV data set config element to read the data from file system. This element can be used to iterate over all lines of CSV one by one and each line can be fed to different types of samplers in a single thread group.

We are going to present an example use case for this, usesr loging in our system. We wanted to run a CSV based load test for JMeter. CSV file contained data in two columns named as **username** and **password** as follows

	Standard	Standard
1	username	password
2	admin	admin
3	hacker1	hacker1
4	hacker2	hacker2
5	company1	company1
6	company2	company2

2. Add the Thread Group

At this moment we are prepare to open jmeter applicationa and add the thread group.



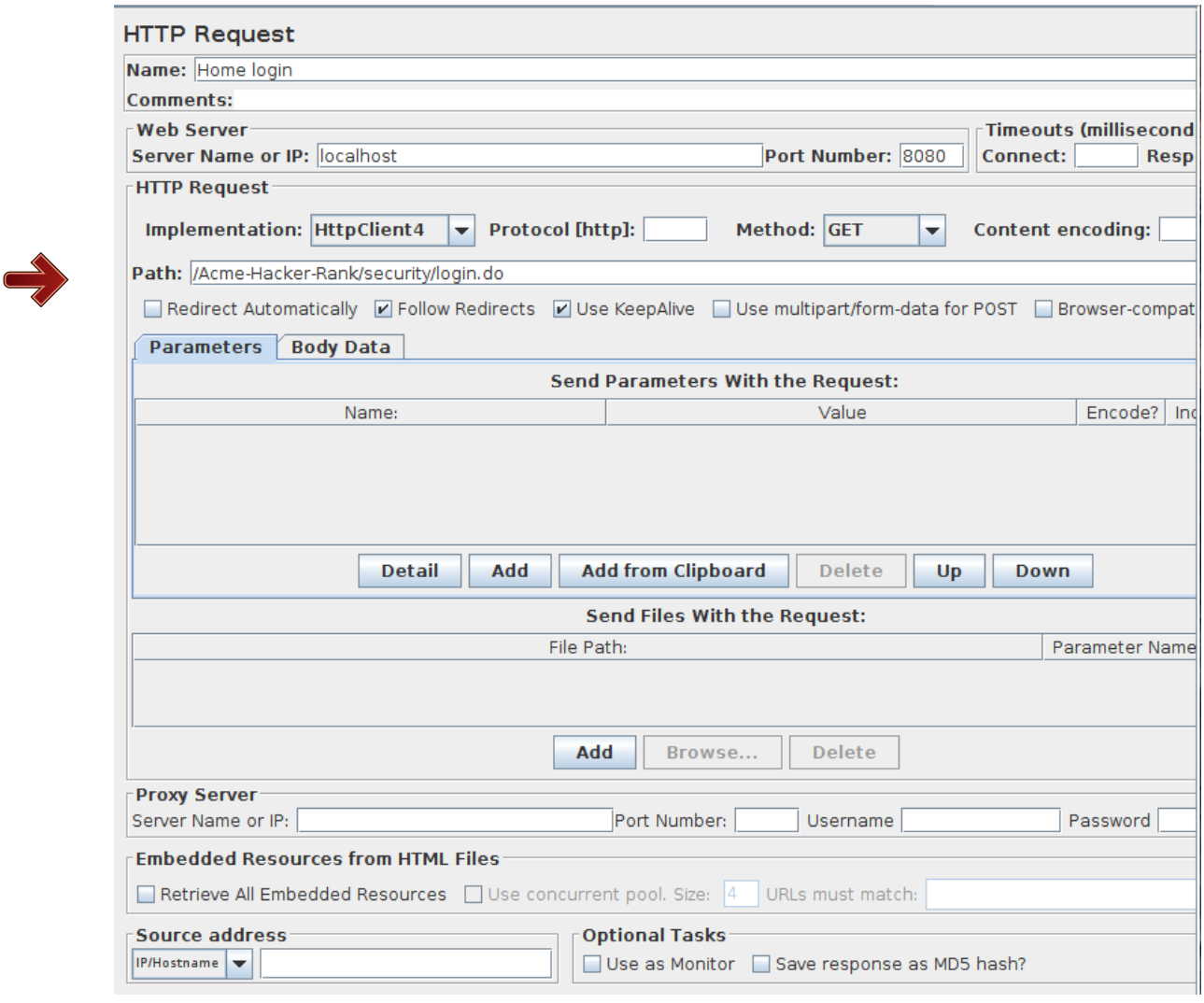
We want simulate an scenario where, 200 users loging and the test will be repeted 50 times.

3. Adding the Http resquest

Now that we have defined our users, it is time to define the task that we will be performing. In our Test Plan, we need to make two HTTP requests. The first one is for the Acme-Rank-Hacker home login page (http://Acme-Hacker-Rank/security/login.do), and the second one is for the login page (http:///Acme-Hacker-Rank/j_spring_security_check).

Start by adding the first HTTP Request to the JMeter Users element (Add → Sampler → HTTP Request). Then, select the HTTP Request element in the tree and edit the following properties:

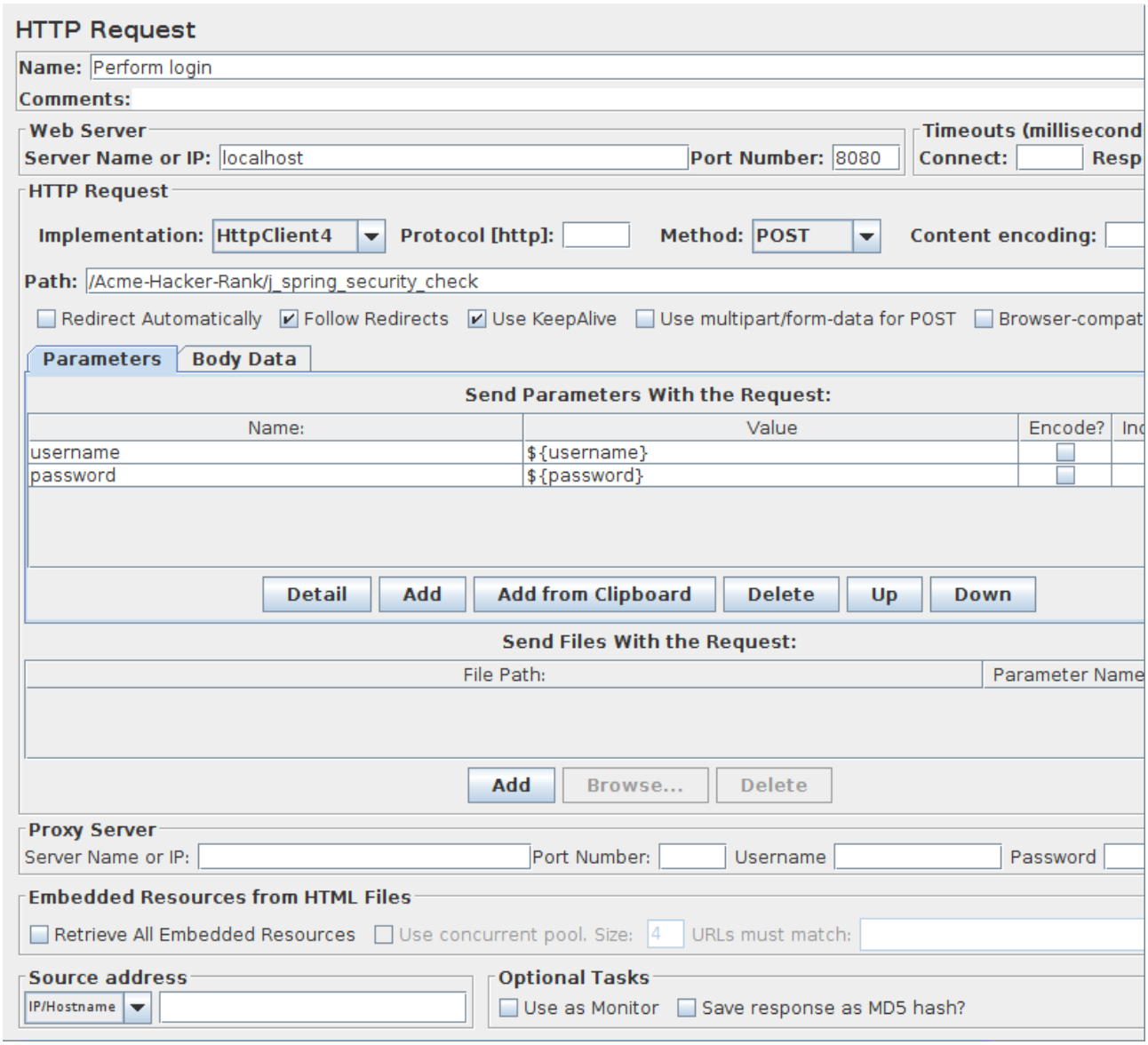
1. Change the Name field to "**Home login**".
2. Set the Path field to "**/Acme-Hacker-Rank/security/login.do**". Remember that you do not have to set the Server Name field because you already specified this value in the HTTP Request Defaults element.



Next, add the second HTTP Request and edit the following properties (see Figure 4.7:

1. Change the Name field to "Perfom login".
2. Set the Path field to "/Acme-Hacker-Rank/j_spring_security_check".
3. Set the parameter and theirs values.

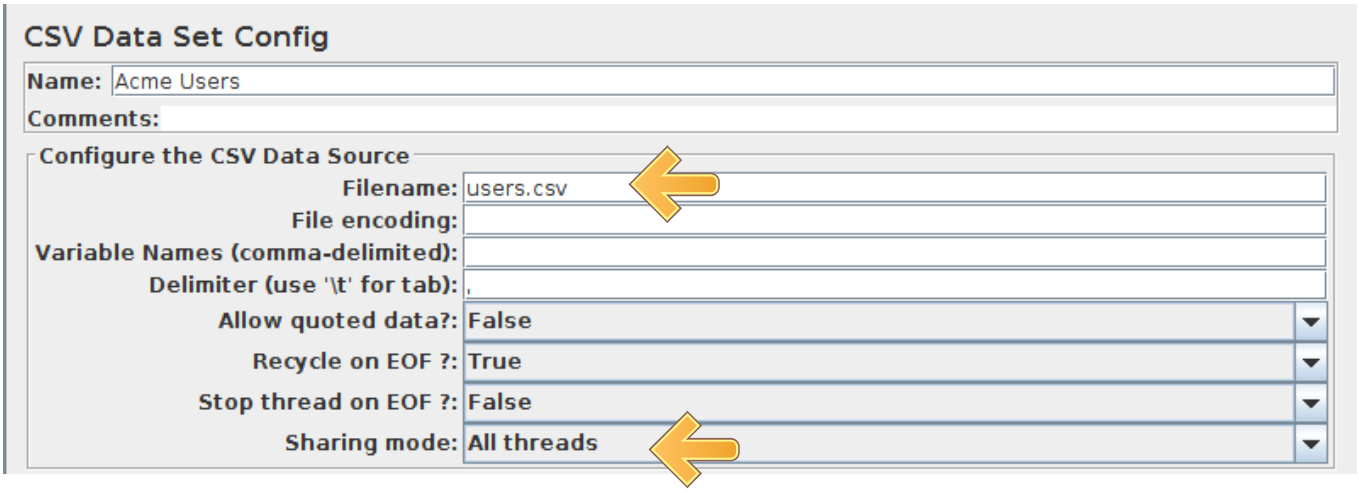
The most important thing is the parameter name must be equals to your first row in the csv file. You'll need to know the names of the fields used by the form, and the target page. These can be found out by inspecting the code of the login page.



4. Adding CSV Data Set Config

Let's add the 'CSV Data Set Config' element to our performance script (*Right click on 'Test Plan' -> Add -> Config Element -> CSV Data Set Config*). In this element, we need to specify the filename that contains user details and variable names, which will be used as containers for these values fetched from the file.

Keep in mind that there are two options: you need to specify the full path to the CSV file or you can just use a filename. But in this case, the file should be located in the same folder as the JMX performance script. We are going to use the default sharing mode: **All threads.**



5. Running the script

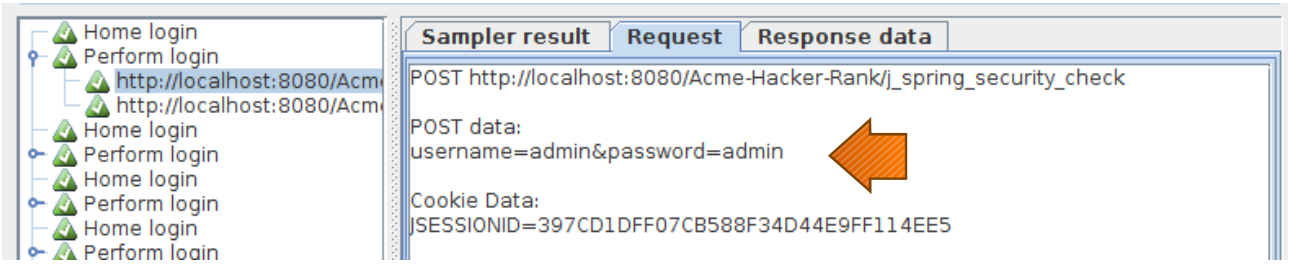
Now is time to run the script.

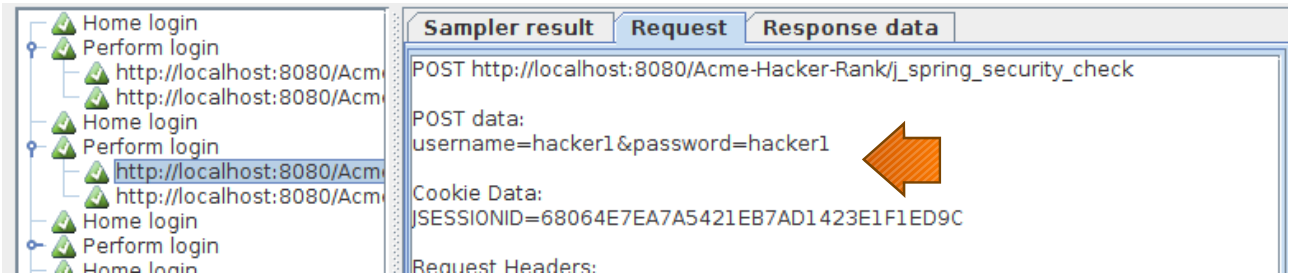
After running our script, you can see that all users took username and passwords respectively based on the order and line number:

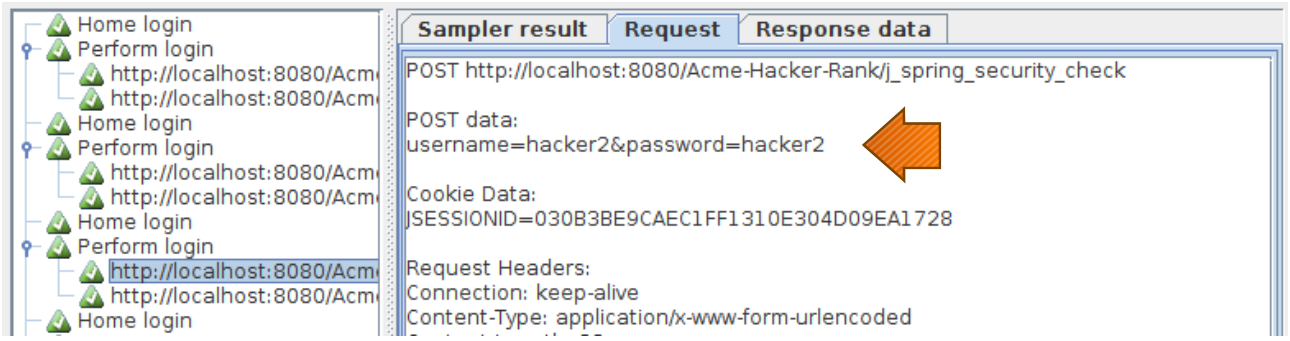
• The first user used the email and password from the first line.

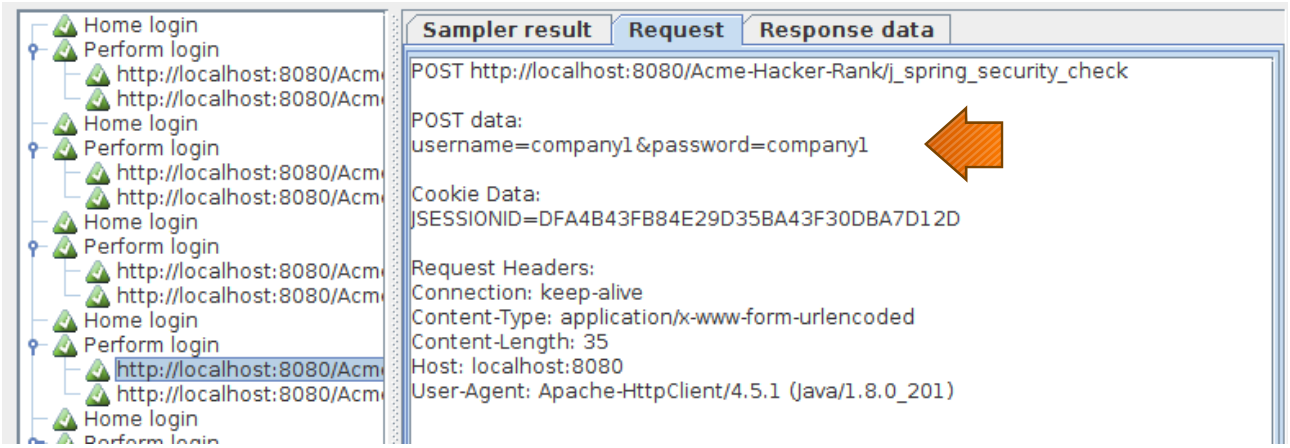
• Second used the email and password from the second line.

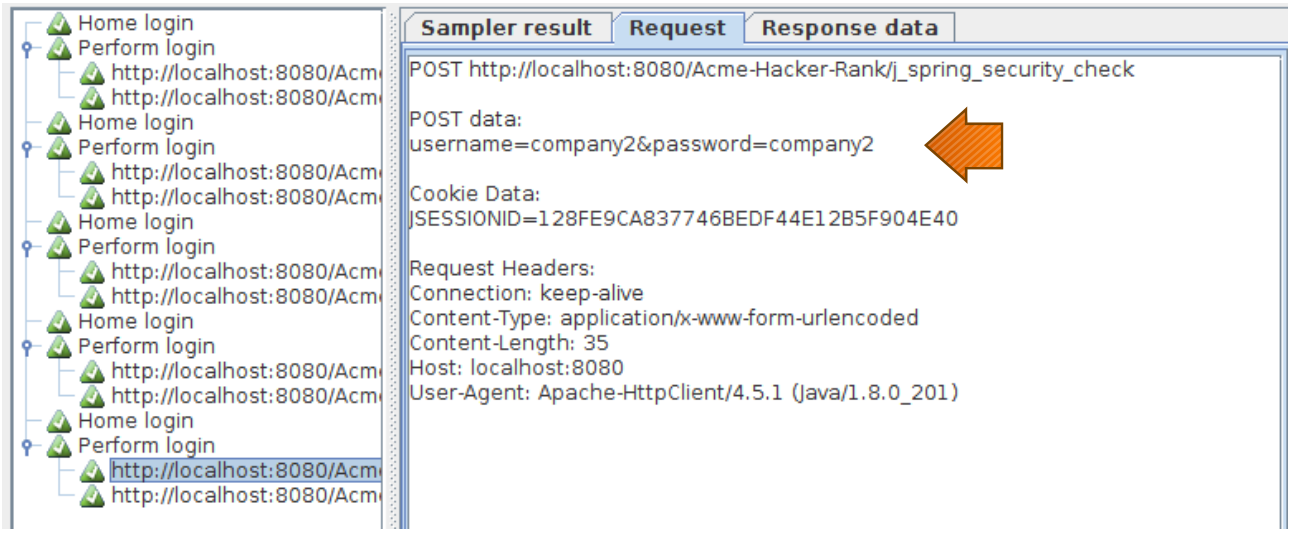
•…



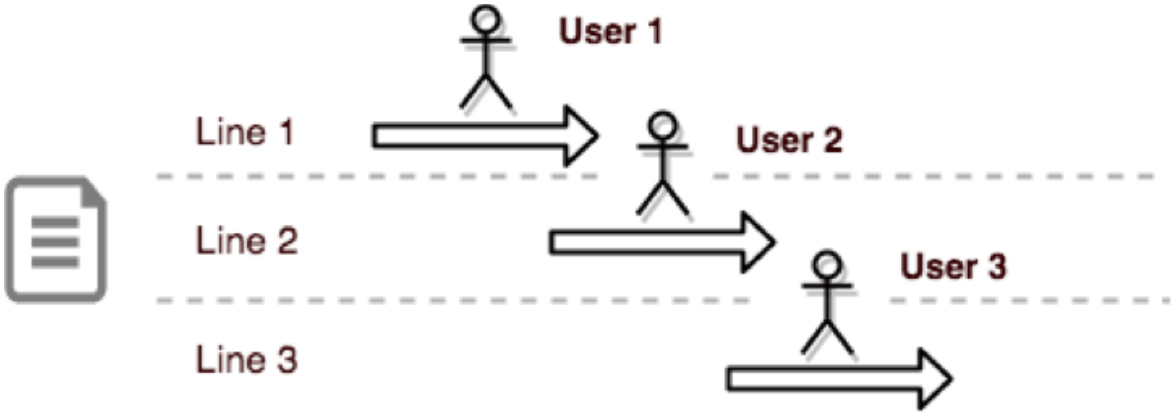

Home login
Perform login
http://localhost:8080/Acm
http://localhost:8080/Acm
Home login
Perform login
http://localhost:8080/Acm
http://localhost:8080/Acm
Home login
Perform login
Sampler result
Request
Response data
POST http://localhost:8080/Acme-Hacker-Rank/j_spring_security_check
POST data:
username=hacker1&password=hacker1
Cookie Data:
JSESSIONID=68064E7EA7A5421EB7AD1423E1F1ED9C
Request Headers:


Home login
Perform login
http://localhost:8080/Acm
http://localhost:8080/Acm
Home login
Perform login
http://localhost:8080/Acm
http://localhost:8080/Acm
Home login
Perform login
http://localhost:8080/Acm
http://localhost:8080/Acm
Home login
Sampler result
Request
Response data
POST http://localhost:8080/Acme-Hacker-Rank/j_spring_security_check
POST data:
username=hacker2&password=hacker2
Cookie Data:
JSESSIONID=030B3BE9CAEC1FF1310E304D09EA1728
Request Headers:
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded

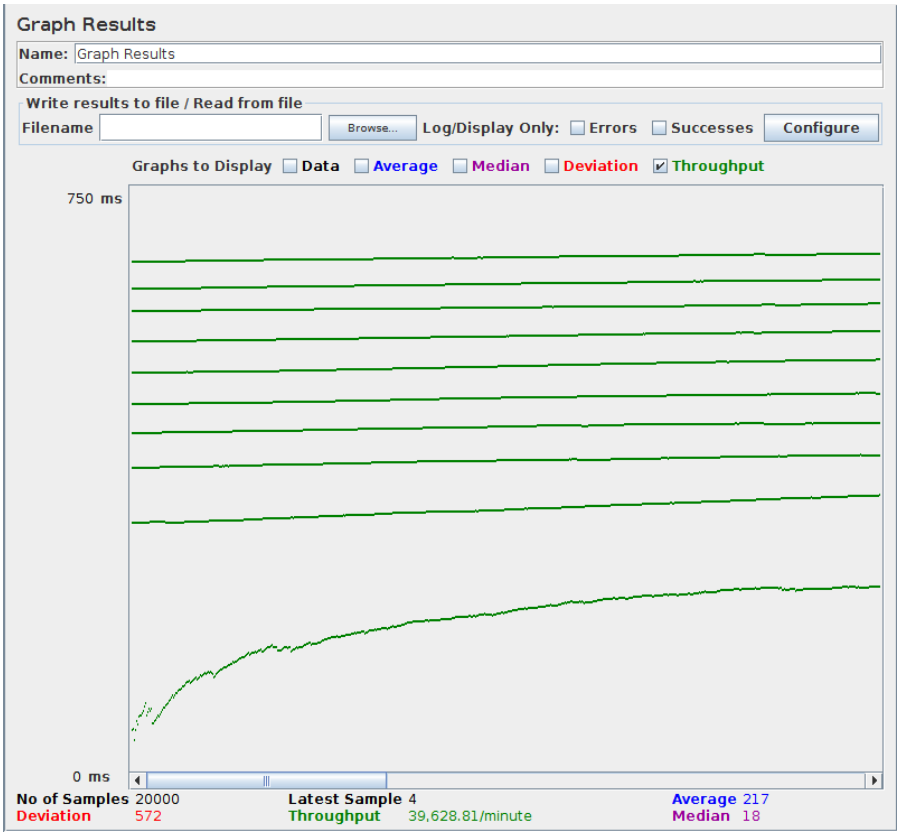

Home login
Perform login
http://localhost:8080/Acm
http://localhost:8080/Acm
Home login
Perform login
http://localhost:8080/Acm
http://localhost:8080/Acm
Home login
Perform login
http://localhost:8080/Acm
http://localhost:8080/Acm
Home login
Perform login
http://localhost:8080/Acm
http://localhost:8080/Acm
Home login
Sampler result
Request
Response data
POST http://localhost:8080/Acme-Hacker-Rank/j_spring_security_check
POST data:
username=company1&password=company1
Cookie Data:
JSESSIONID=DFA4B43FB84E29D35BA43F30DBA7D12D
Request Headers:
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 35
Host: localhost:8080
User-Agent: Apache-HttpClient/4.5.1 (Java/1.8.0_201)


Home login
Perform login
http://localhost:8080/Acm
http://localhost:8080/Acm
Home login
Perform login
http://localhost:8080/Acm
http://localhost:8080/Acm
Home login
Perform login
http://localhost:8080/Acm
http://localhost:8080/Acm
Home login
Perform login
http://localhost:8080/Acm
http://localhost:8080/Acm
Home login
Perform login
http://localhost:8080/Acm
http://localhost:8080/Acm
Sampler result
Request
Response data
POST http://localhost:8080/Acme-Hacker-Rank/j_spring_security_check
POST data:
username=company2&password=company2
Cookie Data:
JSESSIONID=128FE9CA837746BEDF44E12B5F904E40
Request Headers:
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 35
Host: localhost:8080
User-Agent: Apache-HttpClient/4.5.1 (Java/1.8.0_201)

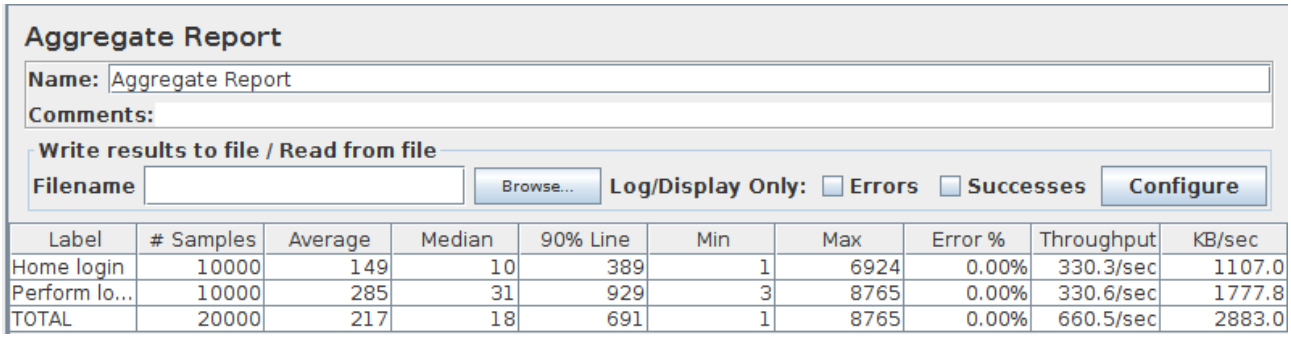
The **‘All threads’** sharing mode means that the file will be shared between all threads and each request will read one line in the CSV file, in sequential order. This way is default for ‘CSV Data Set Config’. It will look like this:



6. Graph Result



7. Aggregate Report



Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
Home login	10000	149	10	389	1	6924	0.00%	330.3/sec	1107.0
Perform lo...	10000	285	31	929	3	8765	0.00%	330.6/sec	1777.8
TOTAL	20000	217	18	691	1	8765	0.00%	660.5/sec	2883.0