

JMeter Performance Testing with CSV Files

Design and Testing 2



Ángel Delgado Luna

Belen Garrido López

Alejandro Rodríguez

Ezequiel Portillo Jurado

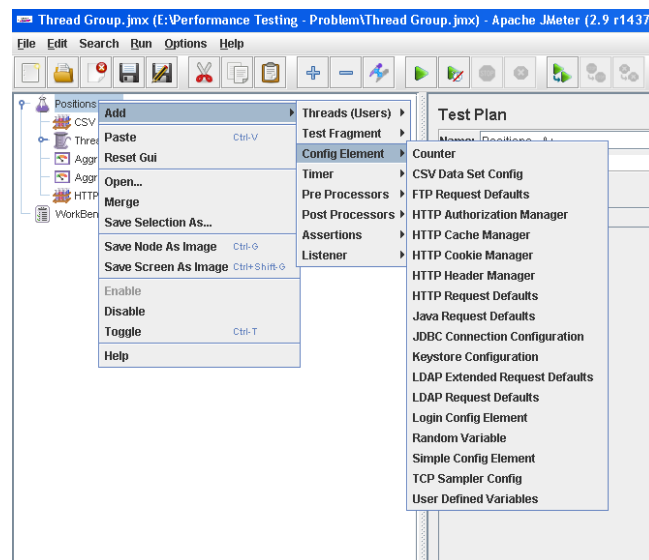
M^a De Gracia Piñero Pastor

Index

- 0. Running JMeter with CSV files**
- 1. Entity chosen and its parameter**
- 2. Process followed**
- 3. Results**
- 4. Bibliography**

0. Running JMeter with CSV files

First of all, we configure JMeter as explained in theory provided. Then, must create into our test plan, a CSV Data Set Config.



Then, we must define some parameters in order to let JMeter works with our CSV file.

CSV Data Set Config	
Name:	CSV Data Set Config
Comments:	
Configure the CSV Data Source	
Filename:	C:\Documents and Settings\Boss\Desktop\positions.csv
File encoding:	
Variable Names (comma-delimited):	id,ticker,title,description,deadline,profileRequired,skillsRequired,technologies,salary,_finalmode,save
Delimiter (use 't' for tab):	;
Allow quoted data?:	False
Recycle on EOF?:	True
Stop thread on EOF?:	False
Sharing mode:	All threads

Filename. In our case, we have set the file in our desktop.

FileEncoding. No necessary.

Variable Names. They must strictly be name in the same way as object java class. JMeter may have some problems when executing tests.

Delimiter. When a csv file, using excel tool, is created and this one is read using notepad tool from windows, columns appear separated by semicolons. We must tell JMeter, which delimiter is being used for the experiments.

1. Entity chosen and its parameter

The chosen entity has been 'Position'. It has the following parameters:

- Id (It is zero when a position is being created)
- Ticker (For our system, ticker is considered an entity. But, in the saving process is considered a String. We provide a string which fits to the pattern provided)
- Title
- Description
- Deadline
- Profile Required
- Skills Required
- Technologies
- Salary
- Final Mode
- Cancel
- Company (ManyToOne relation)

Using any tool named in the step before, we create a table of data. Be careful of name the columns in the same way as java attributes.

id	ticker	title	description	deadline	profileRequi	skillsRequire	technologies	salary	_finalmode	save
0	COMM-1326	company2	company3	2019%2F05%	company2	company2	company2	20.0	on	
0	COMM-1329	company4	company4	2019%2F05%	company2	company2	company2	20.1	on	
0	COMM-1328	company5	company5	2019%2F05%	company2	company2	company2	20.2	on	
0	COMM-1327	company6	company3	2019%2F05%	company2	company2	company2	20.3	on	
66	COMM-1235	pruebaModi	pruebaModi	2019%2F05%	pruebaModi	pruebaModi	pruebaModi	20.5	on	
67	COMM-1236	pruebaModi	pruebaModi	2019%2F05%	pruebaModi	pruebaModi	pruebaModi	20.6	on	
98305	Comm-8364	pruebaModi	pruebaModi	2019%2F05%	pruebaModi	pruebaModi	pruebaModi	20.7	on	

*Save parameter must be empty. It is the parameter used for save action.

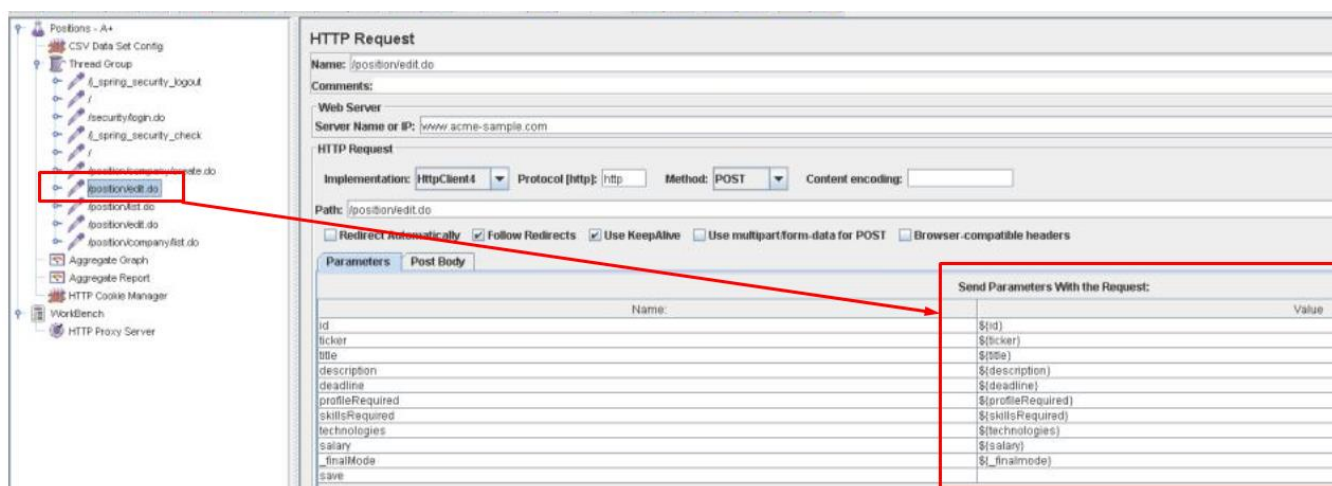
2. Process followed

According to the need of testing multiple cases in terms of performance, JMeter has allowed us with a csv file to register at every iteration multiple elements with other parameters.

The steps followed have been:

1. We start the HTTP proxy server. We record every user interaction related with position.
2. In position/edit.do, we modify all values from value's column. Setting each value in keys with the name of every column declared on the csv file.

*Note that the id which is different than zero, is the defined by the database when population process is executed.



3. Results

As it can be seen, JMeter records data into database. This system helps us to make multiple of performance test making use of different data respect to the saved on database.

```
mysql> select * from Position;
```

id	version	cancel	deadline	description	final_mode	profil
e_required	salary	skills_required	technologies	title	compa	
54	65	120	2019-12-15	position1	@	teache
54	62	1	2019-05-25	spring	position1	
Modifica 2	66	20.5	2019-05-26	pruebaModifica 2	pruebaModifica 2	prueba
54	63	1	2019-05-27	spring	pruebaModifica 3	prueba
Modifica 3	67	20.6	2019-05-28	pruebaModifica 3	pruebaModifica 3	prueba
54	64	1	2019-05-29	spring	pruebaModifica 4	prueba
Modifica 4	98305	20.7	2019-05-30	pruebaModifica 4	company4	prueba
54	98304	0	2019-05-20	company3		compan
y2	98309	20	2019-05-21	company2	company2	compan
54	98308	0	2019-05-22	company4		compan
y2	98311	20.1	2019-05-23	company2	company4	compan
54	98310	0	2019-05-24	company2		compan
y2	98313	20.2	2019-05-25	company2	company5	compan
54	98312	0	2019-05-26	company2		compan
y2	98315	20.3	2019-05-27	company3		compan
54	98314	0	2019-04-26	company2	company6	compan
on1	98317	20	Position1	Position1	Position1	Positi
54	98316					

Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
/spring_security_logout	100	25	22	33	17	121	0.00%	41.3/min	1.3
/	200	29	17	23	12	1306	0.00%	1.3/sec	3.8
/securitylogin.do	100	28	15	30	10	396	0.00%	41.3/min	1.3
/spring_security_check	100	41	32	58	22	448	0.00%	41.3/min	2.3
/position/company/create.do	100	47	42	67	24	239	0.00%	41.2/min	4.0
/position/edit.do	200	77	50	151	10	652	0.00%	1.3/sec	10.4
/position/list.do	100	81	61	149	35	447	0.00%	41.2/min	3.8
/position/company/list.do	100	73	54	91	37	369	0.00%	41.2/min	3.8
TOTAL	1000	51	31	68	10	1306	0.00%	6.3/sec	29.4

*Test done with 10 users and 100 iterations.

4. Bibliography

- [1] <https://guide.blazemeter.com/hc/en-us/articles/206733689-Using-CSV-DATA-SET-CONFIG>