ENHANCING LOAD TESTING EFFECTIVENESS: UNLEASHING THE POWER OF THE JMETER SIMPLE TABLE SERVER



JMeter Simple Table Server: An Introduction

JMeter is a popular open-source tool used for performance testing, load testing, and functional testing of web applications. One of the lesser-known features of JMeter is the Simple Table Server (STS), which allows you to store and retrieve test data in a simple tabular format. In this blog post, we'll explore how to implement the STS.

What is the STS?

The STS is a server application that allows you to store and retrieve test data in a tabular format. It provides a simple and easy-to-use interface for creating and managing test data, which can be used in your JMeter test plans. You can store data in tables and use them to simulate real-world scenarios, such as user login credentials, product information, and customer details.

Challenges of using/managing input files locally

- Manually splitting input file record and distributing it on each node.
- If you plan to add a new use case, you will have to repeat the same process again.
- What if you plan to add a new JMeter secondary machine? Now you will have to manage files for each node. For example, 40 files for six nodes are 240 total files.
- What if you want to update part of your input files?

• What if your use case has Maker-Checker flow? Managing FIFO(First In First Out) input list in distributed nodes is quite tedious.

A real-world use case scenario

Consider a scenario where you have 40 thread groups that are using unique user logins. Moreover, you have a distributed JMeter setup with five secondary machines. In this case, you will have to manually split users and copy those files to each secondary machine. so, you end up having 200 input files (40 files for each node equals 200).





Benefits of using the STS

• **Improved test performance**: The STS can improve test performance by allowing virtual users to share data. This reduces the need for virtual users to read and write data from external data





Figure 2.

- **Easy data sharing**: The STS simplifies data sharing between virtual users. Virtual users can easily access and modify data stored in the STS table, making it easier to create complex test scenarios and create, read, update, and delete the table on the fly.
- **Flexibility**: The STS can be used with CSV files, which makes it easy to integrate with different types of applications.
- **Realistic test scenarios**: The STS enables virtual users to share data, which can create more realistic test scenarios. For example, multiple virtual users can access and update the same data simultaneously, simulating a real-world scenario such as Maker-Checker flow.
- **Centralized data management**: The STS provides centralized data management in a single location, which simplifies the management and maintenance of test data.
- **Improved test accuracy**: By allowing virtual users to share, access, and modify the same data, you can reduce data inconsistencies and errors and gain complete control on how you read the file, i.e., unique, sequential, or random.

Configuring the STS

Implementing STS can be done in just a few steps:

Step 1: Download and Install JMeter

To use the STS, you must have JMeter installed on your machine. You can download the latest version from the official website and follow the installation instructions.

Step 2: Download the STS Plugin

Once you have installed JMeter, download the <u>STS plugin</u> and install it by copying the JAR file to the "lib/ext" folder in your JMeter installation directory.

Step 3: Configure the STS

After installing the plugin, you need to configure the STS by adding the following properties in the jmeter.properties file.

In this example, I chose the dataset directory location at C:/jmeter_for_sts/dataset. You can select any location.

By default, the port is 9191, which you can change per your requirements.



Figure 3.

Now we will create a CSV file with test data and upload it to the server (at the above "datasetDirectory" location).

The CSV file should contain the test data in subsequent rows as illustrated below:

A	Α	В	С	D	E	F
1	login1	password1				
2	login2	password2				
3	login3	password3				
4	login4	password4				
5	login5	password5				
6	login6	password6				
7						
8						
9						
	1					

Figure 4.

Now you can execute following command in your browser and it will display the number of records in the file: <u>http://hostname:port/sts/INITFILE?FILENAME=env1_login.csv</u>





Step 4: There are various ways to start the STS. In this example, I am starting by using the simple-table-server.bat file. Go to JMeter's bin directory and click on the "simple-table-server.bat" file.

simple-table-server - Shortcut	- 🗆 X
loglevel=INFO 13:03:37.274 [main] INFO	org.jmeterplugins.protocol.http.control.HttpSimpleTableServer - Creating HttpSimpleTable from
13:03:37.277 [main] INFO	org.jmeterplugins.protocol.http.control.HttpSimpleTableServer
13:03:37.277 [main] INFO 13:03:37.277 [main] INFO s/dataset	org.jmeterplugins.protocol.http.control.HttpSimpleTableServer - SERVER_PORT : 5678 org.jmeterplugins.protocol.http.control.HttpSimpleTableServer - DATASET_DIR : C:/jmeter_for_st
13:03:37.277 [main] INFO 13:03:37.277 [main] INFO 13:03:37.278 [main] INFO	org.jmeterplugins.protocol.http.control.HttpSimpleTableServer - ADD TIMESTAMP : true org.jmeterplugins.protocol.http.control.HttpSimpleTableServer - DEAMON PROCESS : false org.jmeterplugins.protocol.http.control.HttpSimpleTableServer - charsetEncodingHttpResponse :
UTF-8 13:03:37.278 [main] INFO 8	org.jmeterplugins.protocol.http.control.HttpSimpleTableServer - charsetEncodingReadFile : UTF-
13:03:37.278 [main] INFO -8	org.jmeterplugins.protocol.http.control.HttpSimpleTableServer - charsetEncodingWriteFile : UTF
13:03:37.278 [main] INFO	org.jmeterplugins.protocol.http.control.HttpSimpleTableServer
13:03:37.278 [main] INFO 13:03:37.283 [main] INFO	org.jmeterplugins.protocol.http.control.HttpSimpleTableServer - STS_VERSION : 3.1 org.jmeterplugins.protocol.http.control.ServerRunner - Server started
13:03:37.284 [main] INFO 13:03:37.284 [main] INFO	org.jmeterplugins.protocol.http.control.HttpSimpleTableServer - INITFILE at STS startup org.jmeterplugins.protocol.http.control.HttpSimpleTableServer - jmeterPlugin.sts.initFileAtSta
, d	csv
13:03:37.284 [main] INFO rtupRegex=false	org.jmeterplugins.protocol.http.control.HttpSimpleTableServer - jmeterPlugin.sts.initFileAtSta
13:03:37.294 [main] INFO Communications.csv 13:03:37.336 [main] INFO FILE?FILENAME=Communications	<pre>org.jmeterplugins.protocol.http.control.HttpSimpleTableServer - INITFILE : 1 = 0, fileName = o org.jmeterplugins.protocol.http.control.HttpSimpleTableServer - http://localhost:5678/sts/INIT is.csv. response=<html><title>OK</title>cbody>1450x/body></html></pre>

Figure 6.

Upon successful startup, you will see a screen like the one shown above with the server port number and the location of the file.

It will also load the list of files you specified in the initFileAtStartup parameter in jmeter.properties. As a best practice, always add your input files in the "initFileAtStartup parameter." Files that are not mentioned will disappear when the STS restarts.

Now, try to access the URL IP address: <port>/sts.

It should give you the output below with all the commands related to the STS.



Once you see this page, you have successfully configured the STS.

Load File	Load file in memory: http://hostname:port/sts/INITFILE?FILENAME=file.csv This is command will load the file in STS. You will have to actually copy the file to STS directory
Read	Get one line from list: http://hostname:port/sts/READ?FILENAME=file.txt&[READ_MODE=[<i>FIRST,LAST,RANDOM</i>]] &[KEEP=[<i>TRUE,FALSE</i>]] There are various ways through which you can read the file. Either First, Last or Random. Keep TRUE is used if you want to keep the record after getting read. False will delete the record once it is read from the table.
S Save	Save the specified linked list in a file to the default location: http://hostname:port/sts/SAVE?FILENAME=file.txt&[ADD_TIMESTAMP=[<i>FALSE,TRUE</i>]] This command will actually save the file. This is mostly used in maker-checker flow.
S Status	Display the list of loaded files and the number of remaining lines for each linked list: http://hostname:port/sts/STATUS This command will display the status of all the files you have loaded in sts. It will also display the number of records in the file
Remove	Remove all of the elements from the specified list: http://hostname:port/sts/RESET?FILENAME=file.txt It will remove the files which you have loaded in STS.

Figure 8.

How to integrate the STS into your test plan:

Assuming you followed the above steps and your STS is now fully up and running. Let's try to implement this in your actual test plan. We will practice how one can read the STS data and use it in subsequent requests.

1. Load the file in memory with the INITFILE command.



2. Read the file using the READ command and depending upon your choice, you can also read the file through FIRST/LAST/RANDOM commands.

File Edit Search Run Options Tools Help
O:00:08 A 0 0/1 (O:00:08 O:00:08
test dan HTTP Header manager HTTP Code Manager Mane:

3. Using Regular Expression, extract the username and password from the response data.

🖃 🗍 test plan	
HTTP Header manager	Regular Expression Extractor
	Name: jusername
HTTP Cooke Manager	
	Comments:
K HI IP Request Defaults	Apply to:
Setup Virtual Table	C Main sample and sub-samples
Connect_to_STS	
Image: Star Star Star Star Star Star Star Star	Field to check
🛨 🍵 Test Fragment	
🖻 🚳 Thread Group	Gedy C Body (unescaped) C Body as a Document C Response Headers C Request Headers C URL C Response Co Construction of the second s
/sts/READ?FILENAME=env1 login.csv&RE	
Jusername	Name of created variable: username
a password	Regular Expression:
TC DaskEvents 01 Dastel Users Dass	
Trial Time	Template (sis where its capturing group number, starts at 1): [\$1\$
	Match No. (0 for Random):
TC_DashEvents_02_Portal_Login	
III Think Time	
UC_DashEvents_xx_Click_Dashboard	
표 🎢 Think Time	
+ Loop Controller	
TC DashEvents 07 Portal Logout	
Pacing	
Pound ReapChall Campler	
beansnell Sampler	
global_error	Since we are reading csy file, we will create a regular
🗤 🏑 View Results Tree	expression to extract 1st column value.
🥁 jp@gc - Synthesis Report (filtered)	Natas in this assessments the set O as howen in most film.
Summary Report	Note: In this example I have 2 column in my input file.
	<u></u>
Erre test plan	Regular Expression Extractor
test plan HTTP Header manager User Defined Variables	Regular Expression Extractor
test plan User Defined Variables HTTP Cookie Manager	Regular Expression Extractor Name: password
test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager	Regular Expression Extractor Name: password Comments:
test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Request Defaults	Regular Expression Extractor Name: password Comments:
test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cocke Manager HTTP Request Defaults ⊟ ↔ Setup Virtual Table	Regular Expression Extractor Name: password Comments:
test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Former Connect to STS	Regular Expression Extractor Name: password Comments:
	Regular Expression Extractor Name: password Comments:
test plan test plan User Defined Variables HTTP Cookie Manager HTTP Cookie Manager HTTP Cache Manager HTTP Request Defaults Gonect_to_STS ⊕ fist/INITFILE?FILENAME=env1_login. Test Frament	Regular Expression Extractor Name: password Comments:
	Regular Expression Extractor Name: password Comments:
test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Connect_to_STS Setup Virtual Table Test Fragment Test Fragment Connect_to_STS Connect_STS	Regular Expression Extractor Name: password Comments:
test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Connect_to_STS Connect_to_STS Connect_to_STS Test Fragment Thread Group (sts/READ/FILENAME=env1_login.csv&RE	Regular Expression Extractor Name: password Comments:
test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Setup Virtual Table Setup Virtual Table Setup Virtual Table Test Fragment Thread Group Intered Group Username	Regular Expression Extractor Name: password Comments:
test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Request Defaults Connect_to_STS Connec	Regular Expression Extractor Name: password Comments:
 test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Connect_to_STS Connect_to_STS Test Fragment Thread Group (sts/READ?FILENAME=env1_login.csv&RE username password TC_DashEvents_01_Portal_Home_Page 	Regular Expression Extractor Name: password Comments:
	Regular Expression Extractor Name: password Comments:
	Regular Expression Extractor Name: password Comments:
test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager Ists/INITFILE?FILENAME=env1_login.csv&RE Username Dassword Ists/READ?FILENAME=env1_login.csv&RE Username Dassword Ists/LEAD HTT HTT HTT HTT HTT HTT HTT HTT	Regular Expression Extractor Name: password Comments:
test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cockie Manager HTTP Cache Manager HTTC CashEvents_01_Portal_Home_Page Think Time HTTP CashEvents_02_Portal_Login HTTP Cache Manager HTTP C	Regular Expression Extractor Name: password Comments:
 test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Connect_to_STS StylinitrilE?FILENAME=env1_login. Test Fragment Thread Group (sts/READ?FILENAME=env1_login.csv&RE Username Setup Virtual Table Tc_DashEvents_01_Portal_Home_Page Think Time UC_DashEvents_xx_Click_Dashboard Think Time 	Regular Expression Extractor Name: password Comments:
 test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Connect_to_STS Sts/INITFILE?FILENAME=env1_login.csv&RE Tsts Fragment Thread Group /sts/READ?FILENAME=env1_login.csv&RE username password T C_DashEvents_01_Portal_Home_Page Think Time UC_DashEvents_xx_Click_Dashboard Think Time Loop Controller 	Regular Expression Extractor Name: password Comments:
 test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cocke Manager HTTP Cocke Manager HTTP Cocke Manager HTTP Request Defaults Setup Virtual Table Connect_to_STS Setup Virtual Table Connect_to_STS Test Fragment Thread Group //sts/READ?FILENAME=env1_login.csv&RE Jusername Jusername Jusername TC_DashEvents_01_Portal_Home_Page Think Time UC_DashEvents_xx_Click_Dashboard Think Time Loop Controller TC_DashEvents_07_Portal_Logout 	Regular Expression Extractor Name: password Comments:
test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cockie Manager HTTP Cache Manager Istrict Tragment Username Wassword Username Wassword Username HTTC_DashEvents_02_Portal_Login Think Time Uop Controller HTC_DashEvents_07_Portal_Logout Waing	Regular Expression Extractor Name: password Comments:
test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cockie Manager HTTP Cache Manager HTTR Cache Manager HTt	Regular Expression Extractor Name: password Comments:
 test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Connect L_STS /sts/INITFILE?FILENAME=env1_login.csv&RE (sts/READ?FILENAME=env1_login.csv&RE Username (sts/READ?FILENAME=env1_login.csv&RE Tc_DashEvents_01_Portal_Home_Page Think Time TC_DashEvents_02_Portal_Login Think Time UC_DashEvents_07_Portal_Logout Pacing BeanShell Sampler global_error 	Regular Expression Extractor Name: password Comments:
 test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Cache Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Connect_to_STS Connect_to_STS Sts/INITFILE?FILENAME=env1_login.csv&RE Test Fragment Thread Group (sts/READ?FILENAME=env1_login.csv&RE Jassword TC_DashEvents_01_Portal_Home_Page Think Time CC_DashEvents_02_Portal_Login Think Time Uc_DashEvents_07_Portal_Logout TC_DashEvents_07_Portal_Logout Pacing BeanShell Sampler global_error Response Assertion 	Regular Expression Extractor Name: password Comments:
 test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Cache Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Connect_to_STS Sts/INITFILE?FILENAME=env1_login.csv&RE Test Fragment Thread Group /sts/READ?FILENAME=env1_login.csv&RE username password TC_DashEvents_01_Portal_Home_Page Think Time UC_DashEvents_02_Portal_Login Think Time Loop Controller TC_DashEvents_07_Portal_Logout Pacing BeanShell Sampler global_error Response Assertion View Results Tree 	Regular Expression Extractor Name: password Comments:
test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cockie Manager Ists/READ?FILENAME=env1_login.csv&RE Username Dassevord Username Dassevord Ists/READ?FILENAME=env1_login.csv&RE Dassevord Trc_DashEvents_01_Portal_Home_Page Think Time Uc_DashEvents_02_Portal_Logout Think Time Uop Controller T_C_DashEvents_07_Portal_Logout Pacing BeanShell Sampler global_error	Regular Expression Extractor Name: password Comments:
 test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager Setup Virtual Table Connect to_STS HTTP Cache Manager Iss Tragment Trest Fragment Trest Fragment TC_DashEvents_01_Portal_Home_Page Think Time TC_DashEvents_02_Portal_Login Think Time UC_DashEvents_07_Portal_Logout Pacing BeanShell Sampler global_error Response Assertion View Results Tree jp@gc - Synthesis Report (filtered) 	Regular Expression Extractor Name: password Comments:
 test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Connect to_STS Setup Virtual Table Connect to_STS Sts/INITFILE?FILENAME=env1_login.csv&RE Test Fragment TC_DashEvents_01_Portal_Home_Page Think Time TC_DashEvents_02_Portal_Login Think Time UC_DashEvents_07_Portal_Logout Pacing BeanShell Sampler global_error Response Assertion View Results Tree jp@gc - Synthesis Report (filtered) 	Regular Expression Extractor Name: password Comments:
 test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Connect_to_STS Setup Virtual Table Connect_to_STS /sts/INITFILE?FILENAME=env1_login.csv&RE Jthread Group (sts/READ?FILENAME=env1_login.csv&RE Jthread Group TC_DashEvents_01_Portal_Home_Page Think Time TC_DashEvents_02_Portal_Login Think Time UC_DashEvents_07_Portal_Logout Think Time Loop Controller TC_DashEvents_07_Portal_Logout Pacing BeanShell Sampler global_error Response Assertion View Results Tree jp@gc - Synthesis Report (filtered) Summary Report 	Regular Expression Extractor Name: password Comments:
 test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Connect_to_STS Connect_to_STS Test Fragment Thread Group (sts/READ?FILENAME=env1_login.csv&RE (sts/READ?FILENAME=env1_login.csv&RE (sts/READ?FILENAME=env1_login.csv&RE (b) TC_DashEvents_01_Portal_Home_Page Think Time TC_DashEvents_var_Click_Dashboard Think Time Uc_DashEvents_var_Click_Dashboard Think Time Loop Controller BeanShell Sampler global_error Response Assertion View Results Tree jp@gc - Synthesis Report (filtered) Summary Report 	Regular Expression Extractor Name: password Comments:
Itest plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cockie Manager HTTP Cache Manager Mathematic Mat	Regular Expression Extractor Name: password Comments:
Itest plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Connect_to_STS Image: Setup Virtual Table Image: Test Fragment Image: Test Fragment Image: Test Fragment Image: Test Setup Virtual Table	Regular Expression Extractor Name: password Comments:
Itest plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager Iter State Manager Iter State Manager HTTP Cache Manager Iter State Manager Ite	Regular Expression Extractor Name: password Comments:
 test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Connect to_STS //sts/INITFILE?FILENAME=env1_login.csv&RE Test Fragment TC_DashEvents_01_Portal_Home_Page Think Time TC_DashEvents_02_Portal_Login TC_DashEvents_07_Portal_Logout TC_DashEvents_07_Portal_Logout Pacing BeanShell Sampler global_error Response Assertion View Results Tree jp@gc - Synthesis Report (filtered) Summary Report 	Regular Expression Extractor Name: password Comments:
 test plan HTTP Header manager User Defined Variables HTTP Cookie Manager HTTP Cache Manager HTTP Cache Manager HTTP Request Defaults Setup Virtual Table Connect_to_STS Setup Virtual Table Connect_to_STS Setup Virtual Table Test Fragment Thread Group Sts/READ?FILENAME=env1_login.csv&RE Username Setup Virtual Table TC_DashEvents_01_Portal_Home_Page Think Time TC_DashEvents_02_Portal_Login Think Time UC_DashEvents_07_Portal_Logout Pacing BeanShell Sampler global_error Response Assertion View Results Tree jp@gc - Synthesis Report (filtered) Summary Report 	Regular Expression Extractor Name: password Comments:

Likewise, you can read other parameters (if you have more column in the files).

4. Let's run the script and check the output.



Now you can use these extracted parameters in subsequent requests, as needed.

Conclusion

The JMeter Simple Table Server is a useful feature that allows you to store and retrieve test data in a tabular format. It provides an easy-to-use interface for managing test data and can be used to simulate real-world scenarios in your JMeter test plans. By following the steps outlined in this blog post, you can easily implement the STS and start using it in your performance testing projects.