**Virtual Table Server (VTS) For Any PT Tool**

**Description:**

Virtual table Service (VTS) is an application to handle data in multiple scripts during load test without the efforts of data creation beforehand. It is an HP’s product and they have tested it with one lakh virtual users. It acts a centralized repository to store data, which can be shared between VUsers during the test execution. It is already implemented to use via loadrunner. In this document, we explain you how to configure VTS with jmeter and silk performer performance testing tool.

**Requirements:**

* Admin rights on at least one injector machine to install VTS

**Beta - Tested on Tools:**

* Jmeter
* Silk Performer

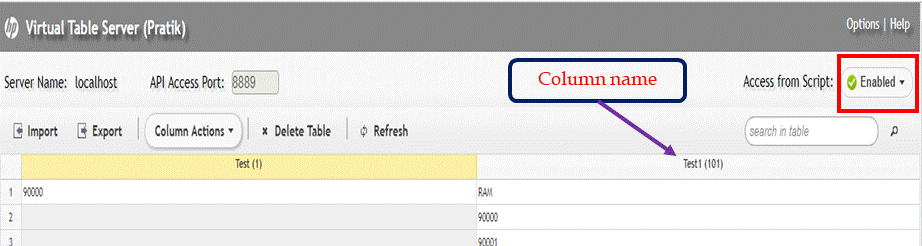
**Installation of VTS Server:**

1. Download the setup from given link
2. Double Click on setup and click on Next > Next
3. Put Admin port as 4000
4. Click on finish
5. After installation, check whether “VTSService” is running in services tab of task manager.
6. Stop this service
7. Go to the installation directory
8. Open /WEB/configure.json
9. Change “"enableDiag": false,” to “"enableDiag": true,”
10. Save the file (if not allowed then save somewhere else and then replace the file)
11. Start “VTSService” from services
12. Open browser and check whether below URL is accessible or not

http://<VTS \_Server\_IP>:4000 /data/diag/apis

1. There are two ports which will be used by default –
   1. Port 4000 – It is an admin port which the hosts the default table. In case you need more tables you can configure from this main instance (not advisable to host more than 1 table; instead create more columns in one table instance)
   2. Port 8888 – It is a default API accessible port, to use in your script. To perform data management activity this port will be used. (In case this port not available you can change to your preferable port)
2. To add columns in the table select column and click on column actions.

**VTS Table:**



**Request flow of VTS to implement in any script:**

1. Request handshake (connection to VTS table is set in this)
2. Below custom data operation can be performed on the tables –
   1. Add – data is added in the mentioned column
   2. Get – data is fetched from the mentioned column
   3. Retrieve - data is fetched and deleted from the mentioned column

**Advantages –**

* Runtime data movement from one script to another
* No need to handle CSV file before every Run
* Before using VTS for your first load test, it is preferable to create data

**Using VTS via Jmeter:**

* **First handshake request sent to establish a connection**

This request can be put in **once only controller**

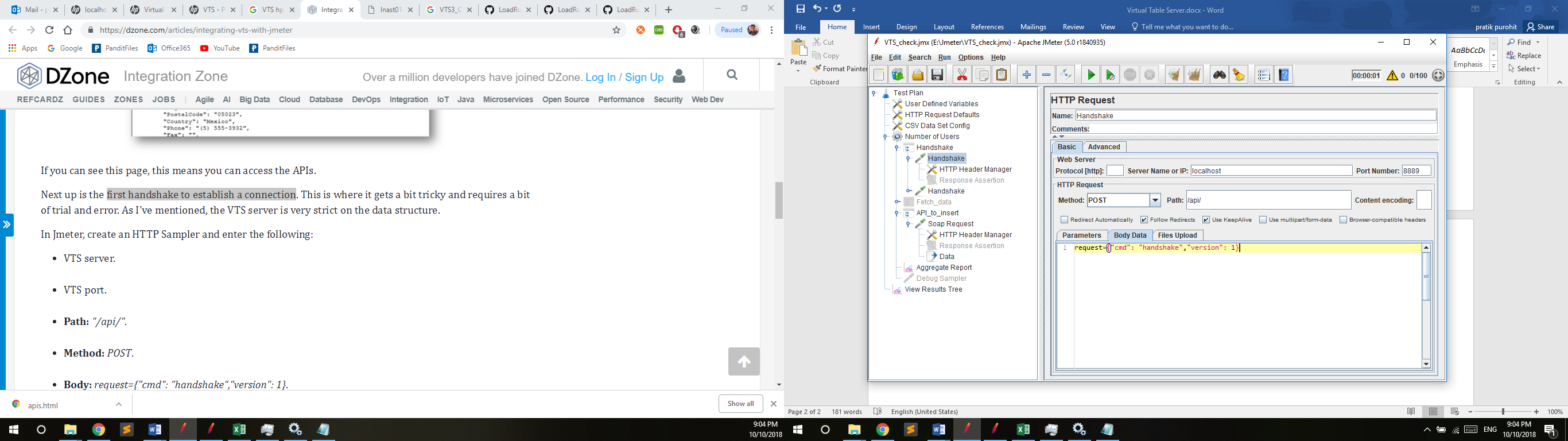
Request:

Header: Content-Type: application/x-www-form-urlencoded

POST http://localhost:<api-port>/api/

POST data:

request={"cmd": "handshake","version": 1}



* **Inserting Data in VTS table**

Request:

Header: Content-Type: application/x-www-form-urlencoded

POST http://localhost: <api-port>/api/

POST data:

request={

"cmd": "add",

"version": "1.0",

"data": {

"new": {

"<column\_name>": "<sample\_data>",

"<column\_name1>": "<sample\_data1>"

},

"option": 0

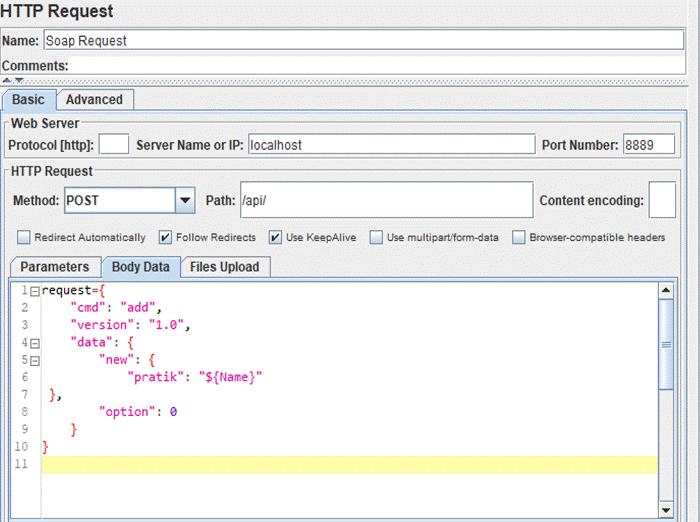
}

}

Request format is in JSON so before adding in script it is advisable to check JSON syntax.

Explanation of above syntax –

* If you want to add data in multiple columns then add column name and corresponding data in new section of the above request.
* "option" has the following values:
  + 0 : add as same row
  + 1 : add as stack
  + 2 : add as unique stack
* Refer below image on where to add the request -



* **Fetching Data and Delete it from VTS table :**

Request:

Header: Content-Type: application/x-www-form-urlencoded

POST http://localhost: <api-port>/api/

POST data:

request={

"cmd": "retrieve",

"version": "1.0",

"data": {

"columns": [

"<column\_name>",

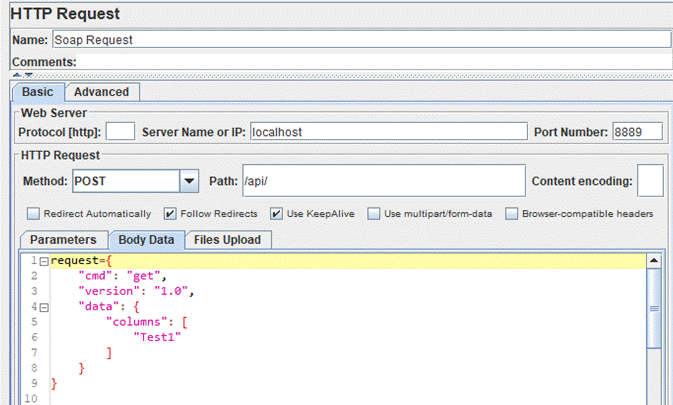
“<column\_name1>”

]

}

}

* Refer below image on where to add the request -



* **Fetching data from VTS table without deleting data :**

Request:

Header: Content-Type: application/x-www-form-urlencoded

POST http://localhost: <api-port>/api/

POST data:

request={

"cmd": "get",

"version": "1.0",

"data": {

"columns": [

"<column\_name>"

],

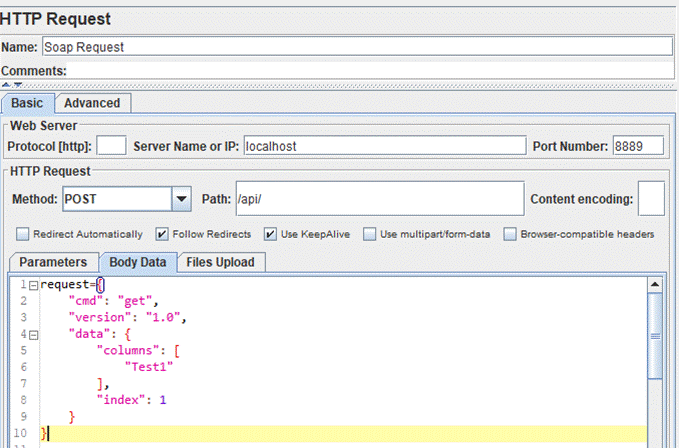
"index": 1

}

}

Here index represents the row number from where data will be fetched.

*\*Disadvantage – All users will pick the same data for every iteration, thereby not maintaining data uniqueness during test execution.*



***\*Note:- While adding data in VTS, it sometimes add a blank row (very rare case). This would cause an issue while fetching data. To overcome this add a while loop to skip the null values and move to the actual value. Refer jmeter script for the same.***

**VTS via Silk Performer:**

* **Function Declaration:** As given in the silk Script. Any change in the number of columns would require change in function declaration.
* **First handshake request sent to establish a connection**

Call Function – (to be put in Tint)

Connect\_VTS("<IP>","<PORT>")

* **Inserting Data in VTS table (Add)**

Call Function

Add\_data("<IP>","<PORT>","<Column\_name>","< sample\_data > ");

* **Fetching Data and Delete it from VTS table (Retrieve)**

Call Function

Retrieve\_data("<IP>","<Port>","< Column name>");

**Please use it and let me know your feedback!**

**Download Links:**

**https://qualitykiosksp-my.sharepoint.com/:f:/g/personal/pratik\_purohit\_qualitykiosk\_com/ErcZWPu-UvFPj7AgOhYlWFIBF09EZcN0BhqbWnwsvS7HIw?e=SrdpCU**