The configurable JMeter script needs accompanying csv files to run. Let's say the URL of the WebApollo instance is http://gmod-dev.nal.usda.gov:8080/cercap. Then in *testdata* directory, we need a subdirectory called *cercap*, and under *cercap* there are two csv files: *usernames.csv* stores usernames and passwords to login the instance, and *operations.csv* stores scaffold names and operations posted to server. The format of *username.csv* is:

```
USERNAME1,PASSWORD1

USERNAME2,PASSWORD2

…
```

All username/password would be used sequentially to login the instance. The format of *operations.csv* is:

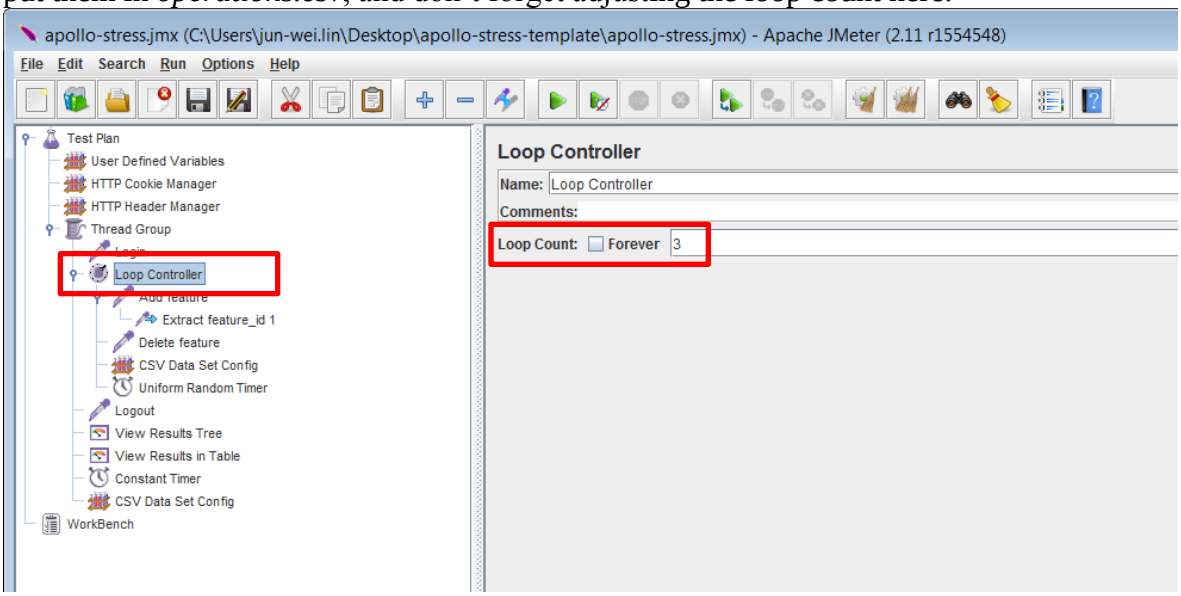
```
SCAFFOLD_NAME1   DATA_TO_BE_POSTED1   NUMBER_TO_MATCH1

SCAFFOLD_NAME2   DATA_TO_BE_POSTED2   NUMBER_TO_MATCH2

…
```

The columns are separated by tab. NUMBER_TO_MATCH is the number used for retrieving feature ID from the responded data by regular expression. For instance, the following is responded json data after adding a feature, and to delete the feature next, we need the 4th *uniquename* (*21F1D14EC304A02FE1F74A9889E47427*) here. So the NUMBER_TO_MATCH is 4.

By default we have three operations for the jmx script to run, if you want more operations, just put them in *operations.csv*, and don't forget adjusting the loop count here:



To run the jmx script with command in Windows:

```
[JMETER_DIR]/bin/jmeter -t apollo-stress.jmx -Jusers=USER -Jloop=LOOP -
Jorganism=ORGANISM -Jserver=SERVER
```

For example, the following command triggers 3 users on *cercap* instance performing a test 5 times. Each user in a test performs a cycle of 8 actions (login, add and then delete feature on three scaffolds, logout); each action had a random delay of 0 to 3 seconds.

```
[JMETER_DIR]/bin/jmeter -t apollo-stress.jmx -Jusers=3 -Jloop=5 -
Jorganism=cercap -Jserver=gmod-dev.nal.udsa.gov
```

We can conduct stress test by spawning processes running on different instances on a server.

p.s. plus -n argument can trigger JMeter in non-GUI mode and generate a log file.