

Variables for Running the Jmeter Script

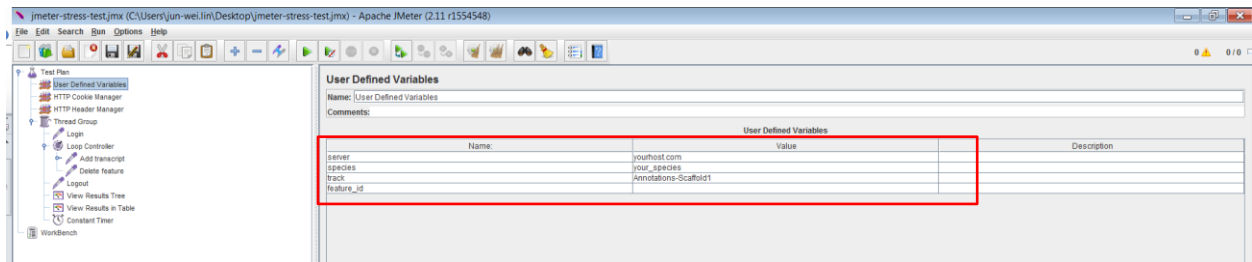
User Defined Variables:

If the operations are performed on http://yourhost.com:8080/your_species/jbrowse/?loc=Scaffold1..., then the User Defined Variables are:

server: yourhost.com

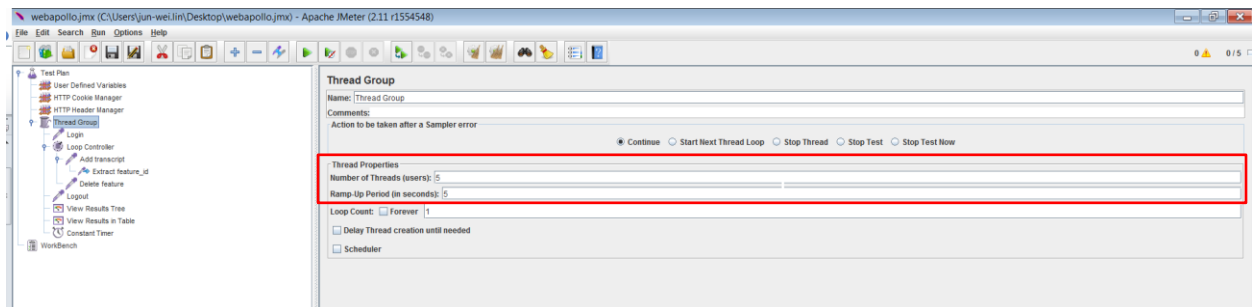
species: your_species

track: **Annotations-Scaffold1**



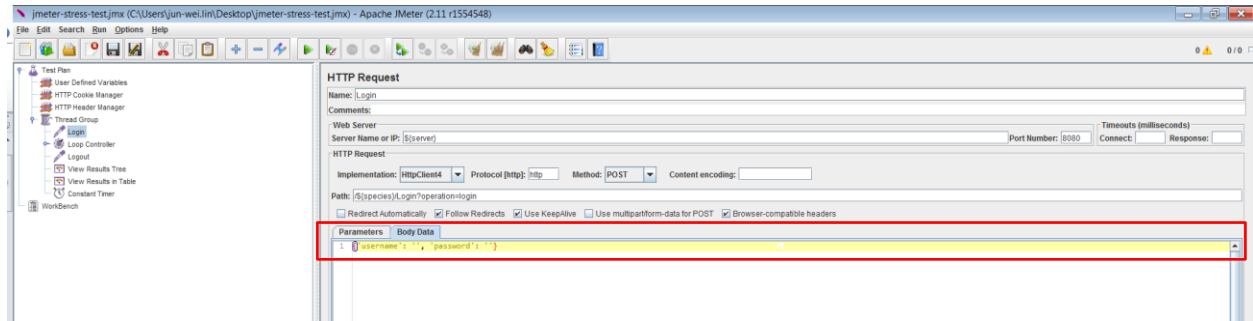
Number of Users:

Adjust number of users in "Thread Group". Ex. Number of Threads=5 and Ramp-up-Period=5 mean five users will be invoked in the speed of one per second.



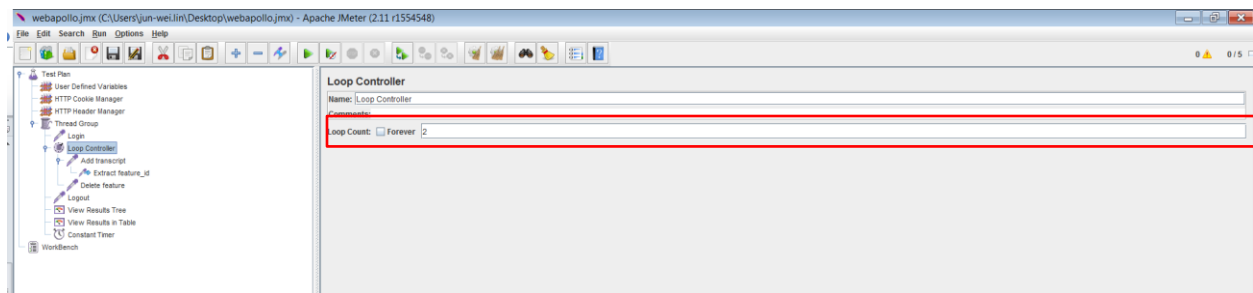
Login

Put username/password for logging in the web apollo instance in “Login”. If the instance is not on port 8080, put the correct port number.



Loop Controller:

“Loop Count” is the repeating times for the group of operations. “Forever” means the script never stops until manually stopped.

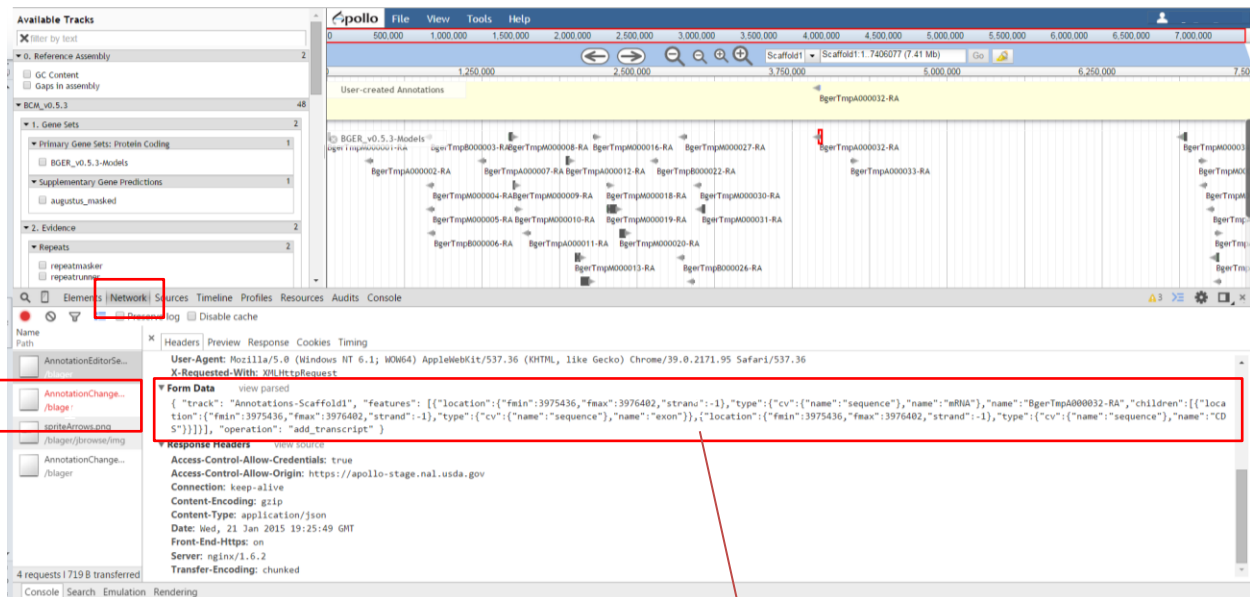


Operations

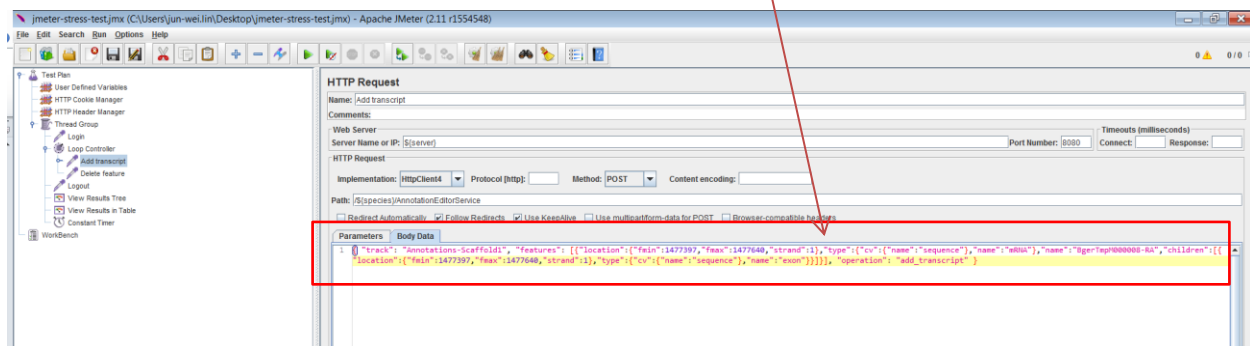
We communicate with the web apollo instance through its web service APIs. Typical operations are:

1. Add a feature by posting json data to the server.
2. Parse the response json to get the feature ID we just added.
3. Delete the feature with the feature ID.

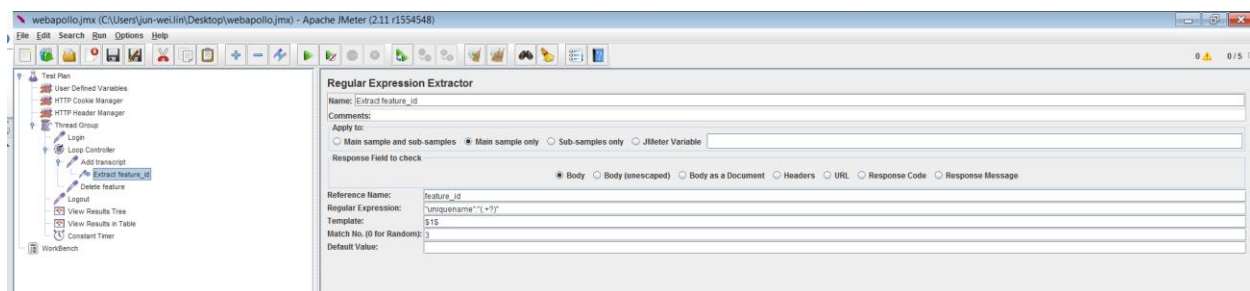
In order to know what data should be posted to server for adding a feature, we first do it manually, and copy the posted data into our script to replay the operation. We can find the data in the network panel of chrome browser.



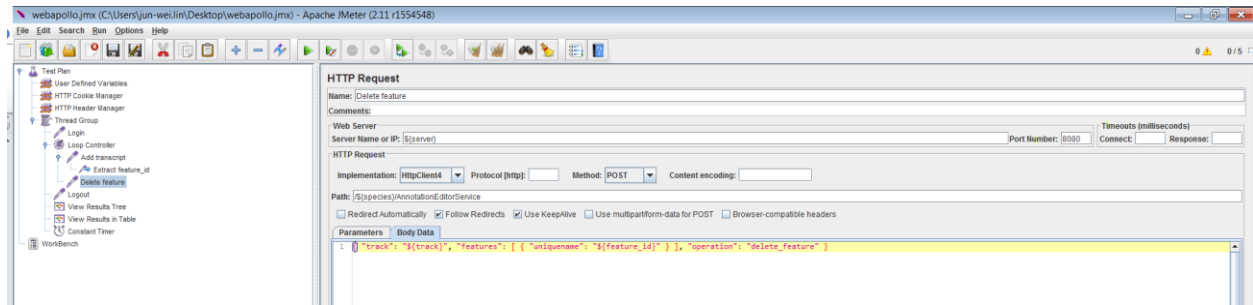
The Form Data is what we need to put into our jmeter script “Add transcript”



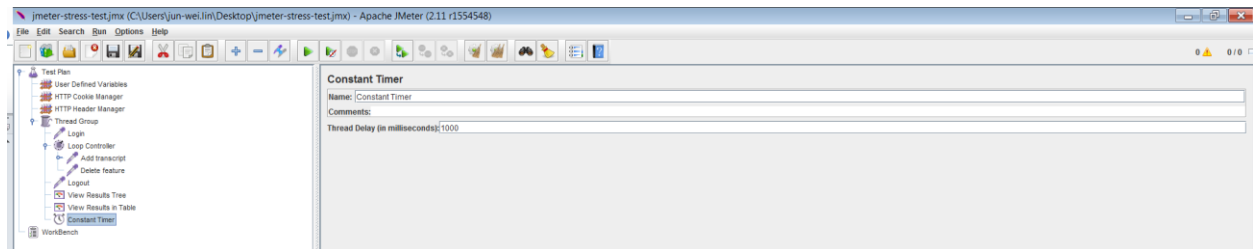
The “Extract feature_id” step extracts the feature ID (the third “uniqueName:” of the response) with a simple regular expression, and by default no modification is needed.



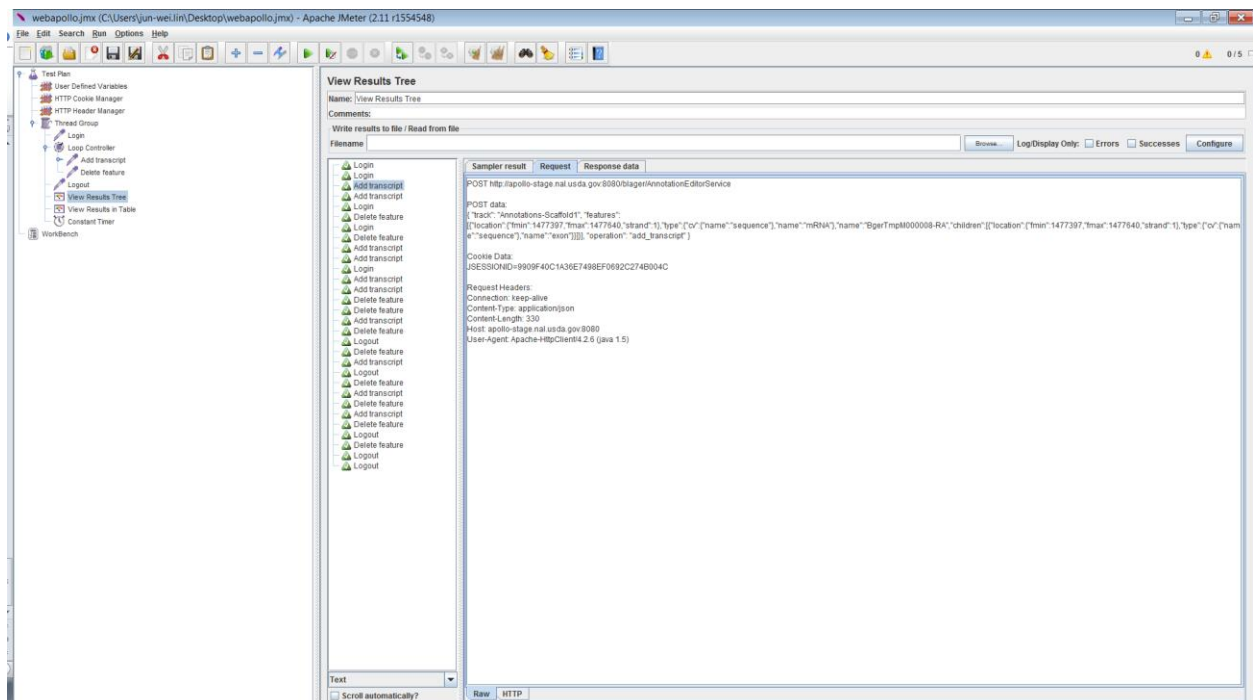
Finally the “Delete feature” step. By default no modification is needed.



Adjust the frequency of actions through the Constant Timer -> Thread Delay.



The result of running the script can be seen in “View Result Tree” or “View Results in Table”



P.S.

If the “Extract feature_id” step fails, modify “Match No” by examining the response to get the correct feature ID. The response can be seen in the network panel, too, and pasting it on an online regex tester would help.

The screenshot shows the Apollo genome browser interface. The top panel displays the Reference Assembly track, followed by Gene Sets (Primary Gene Sets: Protein Coding, BGER v0.5.3-Models, Supplementary Gene Predictions, Augustus_masked), Evidence (Repeats, repeatmasker, repeatrunner), Mapped DNA (Mash, ext2genome), and Mapped Proteins. The main view shows a genomic track with various annotations, including BGER v0.5.3-Models and BGER v0.5.3-Models. The network panel at the bottom shows a request to /blazer/ibrowse/src/dojo/_base/lang.js:78.

The screenshot shows the regexpal.com website, a JavaScript regular expression tester. The input is "uniqueidname:". The output shows a list of matches, including "uniqueidname:02900244E3315E0AD8D4992AE4869478-CDS". The website also includes a search bar, a list of matches, and a "RegexBuddy" button.