Parameters and Variables

Fundamentals:

When performing load testing, it will be common to pass test specific information into other steps of the test. To do this, you will be using Parameters and Variables. The basic difference between the two can be summarized as follows:

Parameters:

Are treated mostly like constants and will exist on a global scope that can be referenced universally by all testing threads.

Examples:

```
NODE_CABINET_TYPE = 4
NODE_DRAWER_TYPE = 5
NODE_FOLDER_TYPE = 6
NODE_FILE_TYPE = 7
```

Variables:

Variables are local to the thread running the tests and will be used for passing ids and the like between test steps. Variables declared on the Test Plan will be copied to a local variable configuration by each testing group. Variables can be assigned to using post processors within JMeter.

Assigning and Referencing Variables and Parameters:

JSON Extractor PostProcessor:

This will be used in general to pull key information out of responses from the server. General information regarding both basic and more advanced use of this post processor can be found here:

https://octoperf.com/blog/2017/03/09/how-to-extract-data-from-json-response-using-jmeter/

Our naming conventions for variables will be in all caps, underscore separated, using the object and property as the name.

Example:

```
TARGET_NODE_ID
TARGET_NODE_TYPE
TARGET_ROLE_ID
```

While TARGET will not be used in all cases, in general the TARGET ID is the singular ID we intend to perform the next test step on. In most steps, the PostProcessors will adjust information to the TARGET IDs based on the server response. Following that, PreProcessor logic that exists outside of the individual tests will use that information to configure certain other variables that may be used in the following step.

To reference a variable that has been stored, you can use the following syntax almost universally throughout JMeter:

```
${TARGET_NODE_ID}
${TARGET_NODE_TYPE}
```

If additional scripting is required, it is also possible to use the BeanShell Processors to access the information and perform logic on it. This is Java, but not too far off from c# and most logic should be easily implemented.

Revision #1
Created 18 July 2022 22:37:08 by Bryce Holloway
Updated 30 June 2024 13:52:41 by Bryce Holloway