Pick Case From Queue Sample



Prepared by

**Zachery Olson**

Solution Architect

Contributors

**List of Contributors**

Revision and Signoff Sheet

Change Record

|  |  |  |  |
| --- | --- | --- | --- |
| Date | Author | Version | Change Reference |
|  | zacolson | 1.0 | Initial draft for review/discussion |
|  |  |  |  |
|  |  |  |  |

Reviewers

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Version Approved | Position | Date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

Table of Contents

[1 Introduction 5](#_Toc130986499)

[1.1 Pre-requisites 5](#_Toc130986500)

[1.2 Local Setup 5](#_Toc130986501)

[1.2.1 Populate CSV 6](#_Toc130986502)

[1.2.2 Update test parameters 6](#_Toc130986503)

[1.2.3 Execute in GUI mode 7](#_Toc130986504)

[1.2.4 Execute in CLI mode 7](#_Toc130986505)

[1.3 Azure load test setup 9](#_Toc130986506)

[1.3.1 CSV Configuration 9](#_Toc130986507)

[1.3.2 User Defined Variables 9](#_Toc130986508)

[1.3.3 Create the Test 10](#_Toc130986509)

[1.4 Resources 11](#_Toc130986510)

1. Introduction

This document will provide the guidance required to set up and execute a single load test script against Microsoft Dataverse using Apache JMeter. The provided test sample will pick an existing queue item from a queue.

* 1. Pre-requisites

**Identity:** The sample is configured to work with cloud-based identities that use Azure Active Directory as an Identity Provider. The user account that will execute the test must have multi-factor authentication disabled.

**Application:** The Pick Case from Queue sample is based on Microsoft Dynamics 365 Customer Service, and it uses the Customer Service Hub model-driven application.

**Sample Data:** In Microsoft Dataverse, make sure that the following data is available for the test to use:

1. Case (minimum 20 active row)
2. Queue (minimum 1 active row)
3. QueueItem (minimum 20 active row)

**Note:** The sample will look for active queue items to pick. More active cases and queue items will be required if the picking rate is higher than the current count of active queue items and cases.

* 1. Local Setup

Load the sample into a local installation of Apache JMeter to set up and verify the test will be executed successfully.

* + 1. Populate CSV

The test will read information from a csv configuration file on every test iteration. The csv consists of the user elements required to execute the test. Add each test user’s information to the PickCaseFromQueue.csv configuration file. The following parameters are required:

* **Username** – The UPN of the user identity
* **Password** – The password for the test user

Populate the configuration as shown in the example below. Add a row for every user that will execute the test.

|  |  |
| --- | --- |
| userName | password |
| [UPN0@tenant.onmicrosoft.com](mailto:UPN0@tenant.onmicrosoft.com) | Password0 |
| [UPN1@tenant.onmicrosoft.com](mailto:UPN1@tenant.onmicrosoft.com) | Password1 |

* + 1. Update test parameters

Load the sample into a local installation of Apache JMeter to set up and verify test execution.

1. Follow the steps below to load an existing Apache JMeter (.jmx file) script file:
   1. In Apache JMeter, click File – Open
   2. Use the explorer window to navigate to file or repository containing the PickCaseFromQueue.jmx file.
   3. Select the PickCaseFromQueue.jmx file and click Open.
2. In User Defined Variables, update the value column for the host and tenantId variables:
   1. host – your Microsoft Dataverse environment url (e.g., yourorg.crm.dynamics.com)
   2. tenantId - your Microsoft 365 tenantid.
   3. cshAppId - the GUID of your Customer Service Hub model-driven application
3. Select PickCaseFromQueue. Update the Thread Group details:
   1. Number of Threads (users) – total concurrent users to simulate.
   2. Ramp up period (seconds) – duration to increase the threads to the total number of threads.
   3. Loop Count – number of times to execute a single thread.
4. Expand PickCaseFromQueue. In the CSV Data Set Config, update the Filename by browsing to the PickCaseFromQueue.csv populated in step 1.2.1
5. The script is pre-configured with a 5 second delay between test iterations (loops). To adjust the delay, update the Pause duration in the Flow Control Action at the bottom of the PickCaseFromQueue thread group.
   * 1. Execute in GUI mode

PickCaseFromQueue can be started in GUI mode using one of the following methods:

1. In the Run menu, click Start.

or

1. Click the Start  button in the toolbar.

or

1. Right click on PickCaseFromQueue, and click Start or Validate

Use the View Results Tree to monitor the progress of the test. View Results Tree is a listener that will capture the requests and samples generated by the samplers in the test. Use View Results Tree to view request and response information, test, and validate post processor extractors.

* + 1. Execute in CLI mode

GUI mode has limitations that might impact the performance of a load test. Use CLI mode when executing a load test at full scale. PickCaseFromQueue can be started in CLI mode following the steps below:

1. Open a command prompt on the machine where Apache JMeter is installed.
2. Change the command directory to the jmeter bin folder.
3. Enter the command: jmeter -n -t path\PickCaseFromQueue.jmx -l path\testresults.csv -e -o path\Results

The command options are defined as:

|  |  |
| --- | --- |
| Option | Definition |
| -n | This specifies JMeter is to run in cli mode. |
| -t | Name of JMX file that contains the Test Plan |
| -l | Name of JTL or CSV file to log sample results to |
| -e | Generate report dashboard after load test |
| -o | Output folder where to generate the report dashboard after load test. The folder must not exist or be empty. |

Additional command options and their definitions can be found [here](https://jmeter.apache.org/usermanual/get-started.html#non_gui).

* 1. Azure load test setup

Before you can create a load test in Azure Load Testing using PickCaseFromQueue.jmx, the CSV Data Set Config must be prepared before it is uploaded in order to pass validation, and the User Defined Variables must be updated to accept environment variables as parameters passed in from Azure Load Testing.

* + 1. CSV Configuration

Azure Load Testing uploads the JMX file and all related files in a single folder. When you reference an external file in your JMeter script, verify that you only use the file name and remove any file path references.

* Remove any file path reference from the CSV Data Set Config and reference the file name only.

Azure Load Testing doesn't preserve the header row when splitting your CSV file.

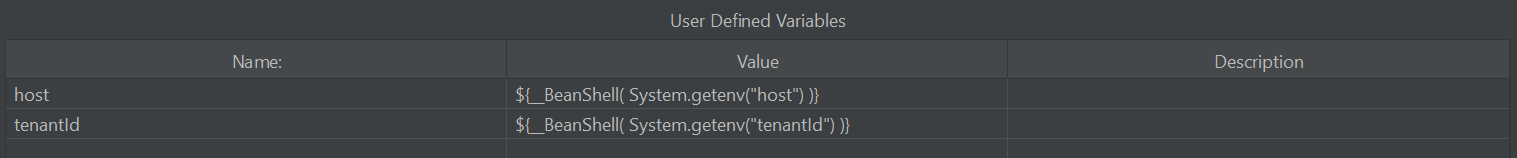
* Enter the CSV field names in Variable Names.
* Set “Ignore first line” to false in the CSV Data Set Config.
* Remove the header row from PickCaseFromQueue.csv.

A screenshot of a computer

Description automatically generated

* + 1. User Defined Variables

Update User Defined Variable values to use the System.getenv() function. The System.getenv("<my-variable-name>") function takes the environment variable name as an argument.



* + 1. Create the Test

Create a load test in Azure Load Testing based on PickCaseFromQueue by following the steps below:

1. In Microsoft Azure, navigate to and open your Azure Load Testing resource.
2. Under Tests, click Tests.
3. Click + Create and choose the option “Upload a JMeter script”
4. On the Basics tab:
   1. Enter a Test name and Test description.
5. On the Test plan tab:
   1. Upload the PickCaseFromQueue.jmx file.
   2. Upload the PickCaseFromQueue.csv file.
   3. Uncheck the “Run test after create” box if you do not wish to run the test automatically after the test is created.
6. On the Parameters tab:
   1. Add a **host** environment variable. Provide your Microsoft Dataverse environment url as the value.
   2. Add a **tenantId** environment variable. Provide your Microsoft 365 tenantid as the value.
   3. Add a **cshAppId** environment variable. Provide the GUID of your Customer Service Hub model-driven application.
7. On the Load tab:
   1. Configure the test engine instances to a meet the target load for the test. The number of threads specified in the jmx file represents the number of thread (virtual users) executed by one test engine instance.
   2. Network traffic to Power Platform applications will be routed through Public networks.
8. On the Test criteria tab:
   1. Define the criteria to specify the performance expectations of the system under test. Use the defined criteria to determine the failure conditions for the test when the criteria evaluate to true.
9. Once the validation has passed on the Review + create tab, click Create.

If the “Run test after create” is checked, the test will execute with no further action required.

* 1. Resources

Apache JMeter CLI - <https://jmeter.apache.org/usermanual/get-started.html#non_gui>

Apache JMeter report dashboard - <https://jmeter.apache.org/usermanual/generating-dashboard.html>

Create a load test with an existing JMeter script - <https://learn.microsoft.com/en-us/azure/load-testing/how-to-create-and-run-load-test-with-jmeter-script>

Read CSV data - <https://learn.microsoft.com/en-us/azure/load-testing/how-to-read-csv-data?pivots=experience-azp>

Parameterize load tests - <https://learn.microsoft.com/en-us/azure/load-testing/how-to-parameterize-load-tests>