

With the addition of In-App variable Key and Value pairs, any data that would normally be hard coded or stored in external data sheets can now be easily stored within Awetest. By adding variable groups within Awetest any test within the project can use the Key:Value pair and can replace, for example, “username” and “password” for actual login data.

This also gives instructions on how to add groups and use the in-app variables within a script.

See the following document for step-by-step instructions:

- You can define Key:Value pairs directly in Awetest for use in scripts. To do this, from the home screen of www.awetest.com go to Web > Assets.

The screenshots show the Awetest web interface with two main views:

- Top Screenshot (WEB Tab Selected):**
 - Dashboard Metrics:** 214 JOBS, 73 of 278 TESTS PASSED, 2157 VALIDATIONS, 01.52 HS OF TESTING.
 - Activity Stream:** Shows recent activity: "1 day ago by Design Tester1" and "5 days ago by Admin/Interactive User". It also displays a message: "WEB JOB WITH 0 SCRIPTS IS QUEUED."
 - Active Test Cases:** A table showing test cases over time periods: 7 Days, 30 Days, 3 Months, All Time. The table includes rows for IE, tc2, tc1, and ForIE.
- Bottom Screenshot (ASSETS Tab Selected):**
 - Dashboard Metrics:** 128 JOBS, 47 of 192 TESTS PASSED, 1286 VALIDATIONS, 01.08 HS OF TESTING.
 - Activity Stream:** Shows recent activity: "1 day ago by Design Tester1" and "5 days ago by Admin/Interactive User". It also displays a message: "WEB JOB WITH 0 SCRIPTS IS QUEUED."
 - Active Test Cases:** A table showing test cases over time periods: 7 Days, 30 Days, 3 Months, All Time. The table includes rows for IE, tc2, tc1, and ForIE.

- Click Test Variables, then Add A Group.

The screenshot shows the 'Project Assets' section of the awet test web interface. On the left, there's a sidebar with 'ASSETS', 'ENVIRONMENTS', and 'TEST VARIABLES'. The 'TEST VARIABLES' option is highlighted with a red box. The main area is titled 'Assets' and contains sections for 'Library', 'Step Definitions', 'Sikuli Scripts', and 'Variable Documents', each with an 'Add' button.

- This will prompt your Variable Group name and description, then click Save.

The screenshot shows a modal dialog titled 'Add Variables Group'. It has two input fields: 'Name' with the value 'Login' and 'Description' with the value 'Assign username/password combinations for testing login functionality'. At the bottom of the dialog are 'CANCEL' and 'SAVE' buttons. In the background, the 'TEST VARIABLES' menu item is visible, and a tooltip for 'Using Test-Variables' provides instructions on how test variables are added as groups.

- Add in your variable Key and Value with optional note, then click Add. Note: Duplicate values for groups and values cannot be added.

The screenshot shows the aw\test web application interface. The top navigation bar includes links for HOME, WEB, inWEB, inAPP, SERVICES, MANAGEMENT, and FAQ, with WEB being the active tab. Below the navigation is a secondary menu with DASHBOARD, TEST CASES, JOBS, REPORTS, and ASSETS. On the left, a sidebar titled 'Project Assets' lists ASSETS, ENVIRONMENTS, and TEST VARIABLES, with 'TEST VARIABLES' expanded to show 'Login'. The main content area is titled 'Login' with a sub-instruction 'Click to edit'. It contains a sub-header 'Add Variable' with three input fields: 'Key', 'Value', and 'Note'. Below this is a table listing variables:

Key	Value	Note
Username	login123	<input checked="" type="checkbox"/>
Password	pass123	<input checked="" type="checkbox"/>

A blue 'Add' button is located at the bottom right of the 'Add Variable' section.

Using In-App Variables in a Script

- These test variables are downloaded along with the script in the shamisen > tmp > [TEST ID] folder.
- All the data is stored in the manifest.json and can be accessed using @params variable for cucumber script type.

```

C:\Awetest\shamisen\tmp\16258\cucumber\watirwd\manifest.json - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
manifest.json
1 {"params": {"browser": "FF", "version": "32.0", "script_file": "cucumber\watirwd.feature", "use_predefined_libraries": true, "include_screenshots": false, "log_properties": {"sequence": 0, "job_id": 6854, "project_version_id": null, "project_id": 55, "test_run_id": 16258, "script_id": 515, "company_id": 37, "test_case_id": 619, "test_category_id": 330}, "log_queue": "regression_test_log", "log_class": "RegressionTest::Log", "notify_queue": "notifications", "notify_class": "RegressionTest::TestRun", "notify_id": 16258, "is_debugging": true, "screencap_path": "/home/awetest/app/tomcat/webapps/ROOT/WEB-INF/tmp", "environment": {"name": "Example", "url": "http://www.google.com/", "node_name": "google example"}, "variables": {"Test variable Group": {"key1": "Value "}}}, "library": [], "variable_document_path": ["predefined_steps.rb"]})

```

Normal text file

length: 783 lines: 1 Ln: 1 Col: 784 Sel: 0 | 0 Dos/Windows ANSI as UTF-8 INS

PM 05:49 12-05-2015

- - For all the other script types, it can be accessed using

```
manifest_file = File.join(File.dirname(__FILE__), '..', 'manifest.json')
```

```
@params = JSON.parse(File.open(manifest_file).read)['params']
```

- @params['variables'][**'TEST_VARIABLE_NAME'**] should return a hash with the data for the given group name.

- Test variables can be downloaded as json by clicking on “download json” link

Project Assets

- ASSETS
- ENVIRONMENTS
- TEST VARIABLES

Test variable Group

Test Variables

Using Test-Variables:

- Test Variables are added as Groups : Click on the Add a Group to create a new group.
- These test variables are downloaded along with the script in the shamsien > tmp > [TEST ID] folder.
- All the data is stored in the manifest.json and can be accessed using @params variable for cucumber script type.
- For all the other script types, can be accessed using


```
manifest_file = File.join(File.dirname(__FILE__), '..', 'manifest.json')
@params = JSON.parse(File.open(manifest_file).read)['params']
```
- @params['variables'][TEST_VARIABLE_GROUP_NAME] should return a hash with the data for the given group name.

DOWNLOAD JSON **ADD A GROUP**

- For the above example, downloaded file contains
- `{"params": {"variables": {"Test variable Group": {"key1": "Value "}}}}`