Katalon Studio For Automated Testing

Sesi 5

Keyword Driven Automation

WHAT IS KEYWORD DRIVEN TESTING ?

According to ISO/IEC/IEEE 29119-5:
Keyword-Driven Testing is a way of describing test cases by using a predefined set of Keywords.

These Keywords are names which are associated with a set of actions that are required to perform a specific step in a test case.
By using keywords to describe test steps instead of natural language, test cases can be easier to understand, to maintain and to automate.
Keyword driven testing divides the test case creation process into two stages:
A. The first stage is Design and Development stage, and
B. the second stage is Implementation stage.

What is Design and Development Stage ?

Design and Development stage: In the first stage, the set of actions that would denote each keyword is designed.
This means that all the actions that need to be taken under a single keyword are sequentially identified and laid down.

What is Implementation Stage ?

Implementation stage: In this stage, the final execution can be either manual or automated or a combination of the two, depending on the situation. These commands are executed in a set and precise order which is determined in the first stage.

Example [WebUI]

Authenticate

```
'Open browser and navigate to demo AUT site.'
WebUI.openBrowser('')

'Input username and password on authentication dialog.'
WebUI.authenticate('http://the-internet.herokuapp.com/basic_auth', 'admin', 'admin', 12)

'Close browser'
WebUI.closeBrowser()
```

Example [Mobile]

Close Notification

```
'Start application on current selected android\'s device'
Mobile.startApplication(GlobalVariable.G_AndroidApp, false)

'Close any sudden notifications'
Mobile.closeNotifications()

'Close application on current selected android\'s device'
Mobile.closeApplication()
```

Example [WS]

Send Request

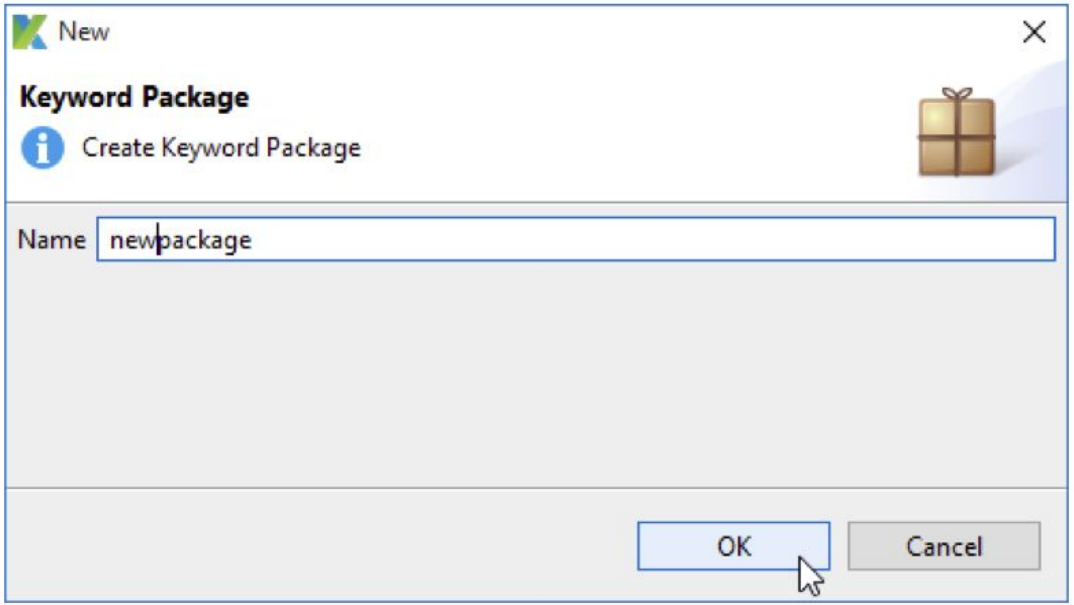
```
'Send a SOAP request and returns its response'
def response = WS.sendRequest(findTestObject('SOAP_ConvertWeight'))

'Verify converted weight after sending request is correct or not'
WS.verifyElementText(response, 'ConvertWeightResult', '3000')
```

CUSTOM KEYWORDS

Create a Package

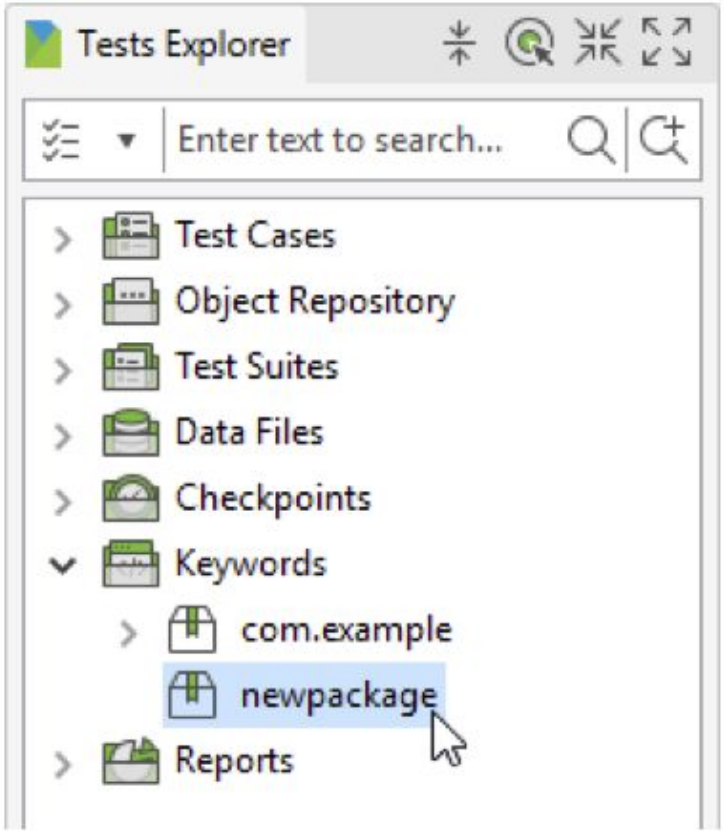
A package allows you to group custom keywords into a specific category, making keywords easier to search and use.

1. Select ***File > New > Package*** from the main menu to open the ***New Keyword Package*** dialog. Enter the name for your package and click ***OK***



CUSTOM KEYWORDS

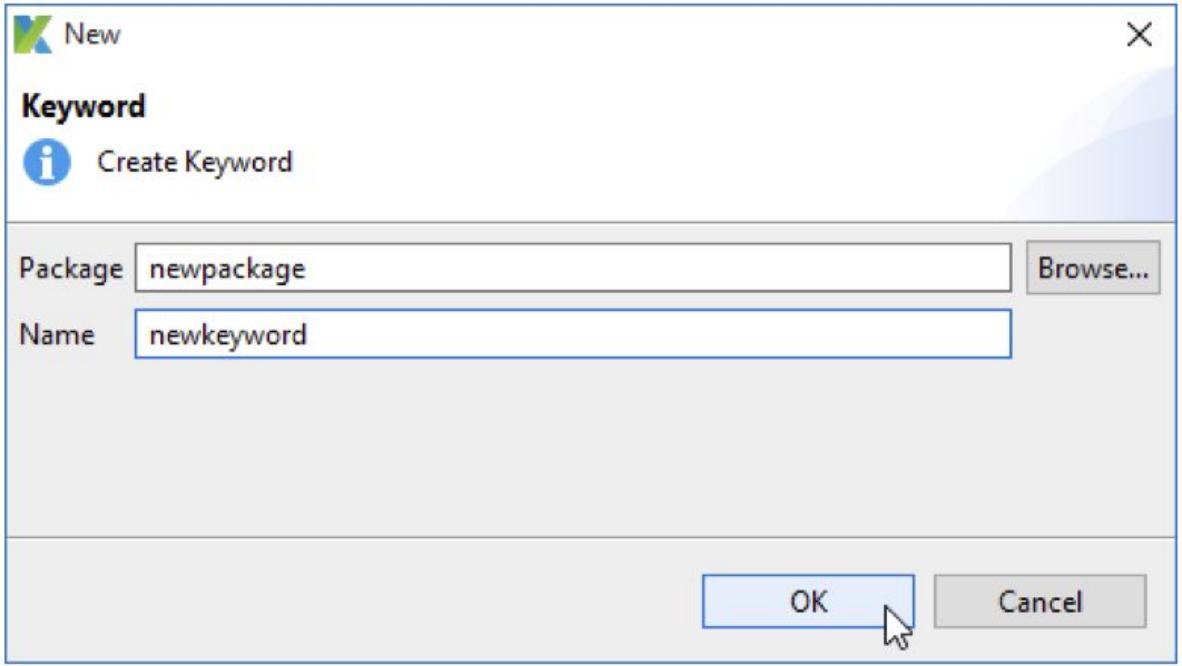
2. A new package is created under ***Keywords*** of Katalon Studio accordingly.



CUSTOM KEYWORDS

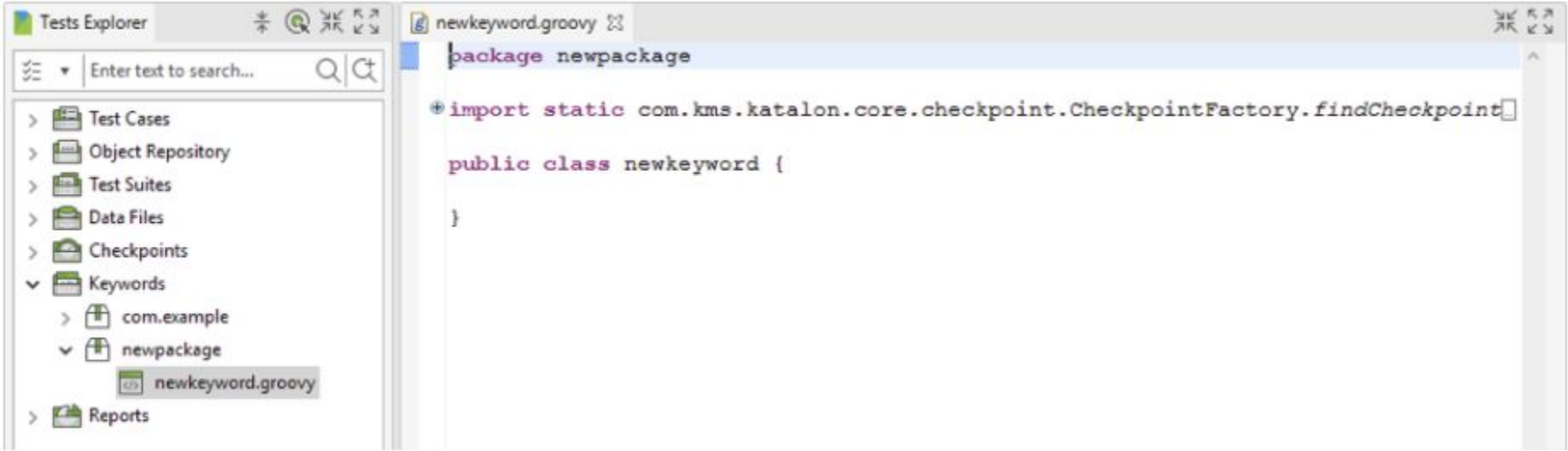
Create a Custom Keyword

1. Select ***File > New > Keyword*** from the main menu to open the ***New Keyword*** dialog. Enter the name for your keyword and select a ***package*** for the keyword. Click ***OK***.



CUSTOM KEYWORDS

2. A new keyword is created under the specified ***package*** accordingly.



CUSTOM KEYWORDS

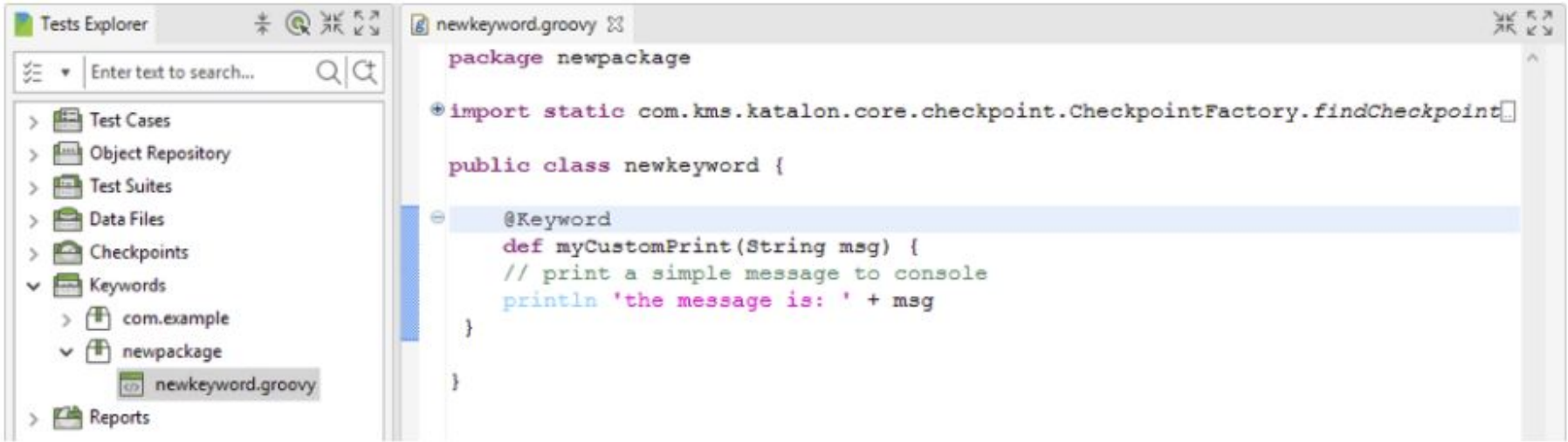
3. Enter the following block of code in your class to define a custom keyword in Java/Groovy:

where:

Item	Description
@Keyword	The annotation to indicate that the block of code below is the definition of a keyword.
keywordName	The name that you would like to use for your custom keyword
parameters	The list of parameters that would be used in the custom keyword

CUSTOM KEYWORDS

For example :



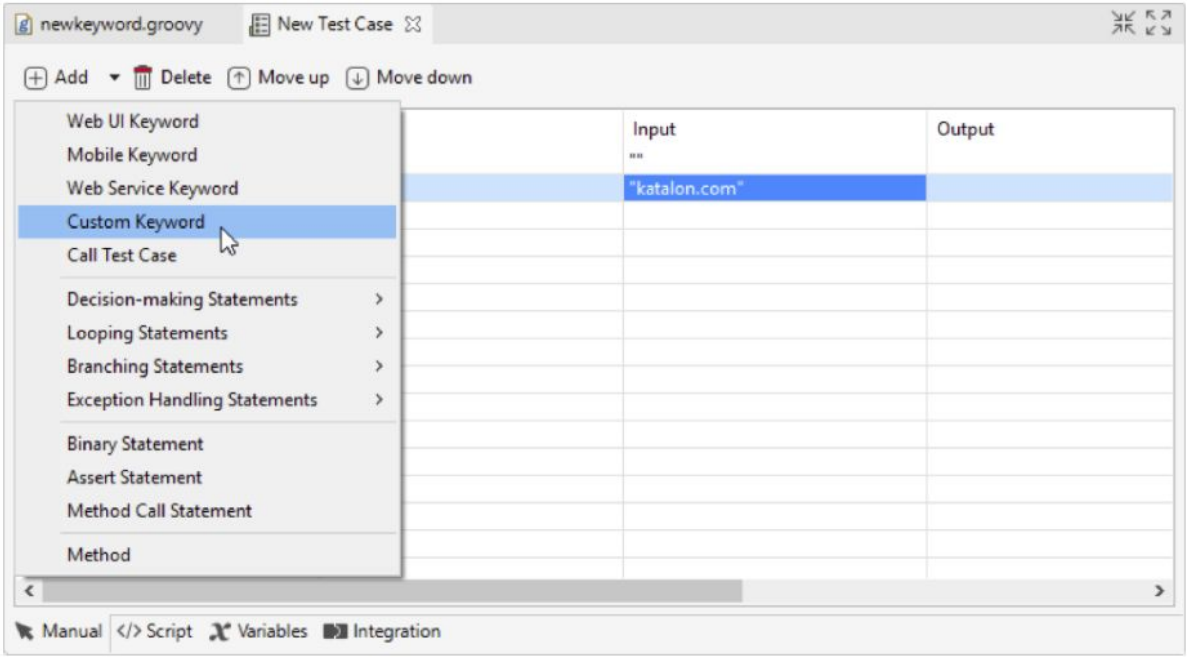
4. Save the ***Keyword*** file when you're done.

CUSTOM KEYWORDS

Custom keywords in Manual view

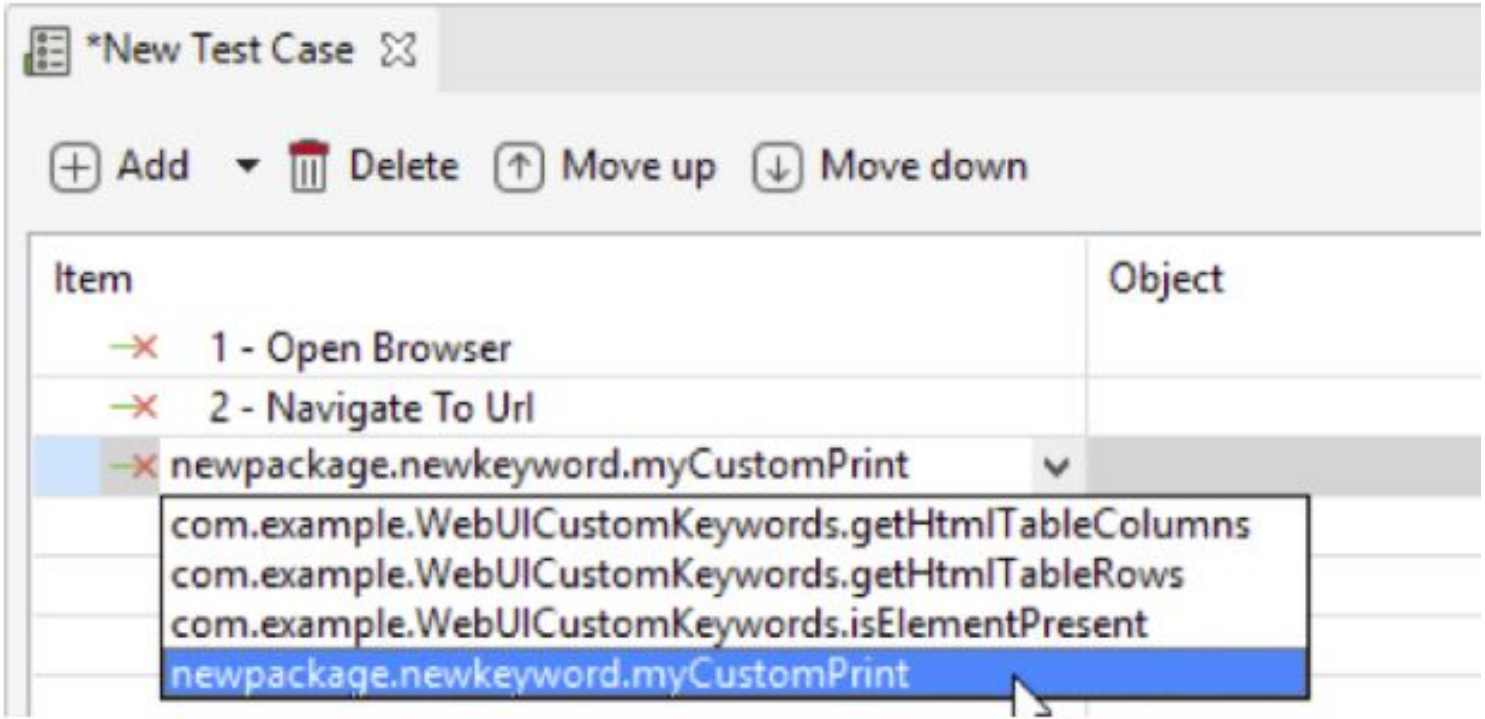
Follow the steps below to use your defined custom keywords in the ***Manual view*** of a test case:

1. Open a test case in ***Manual view***, then add ***Custom Keyword*** in script from the command toolbar.



CUSTOM KEYWORDS

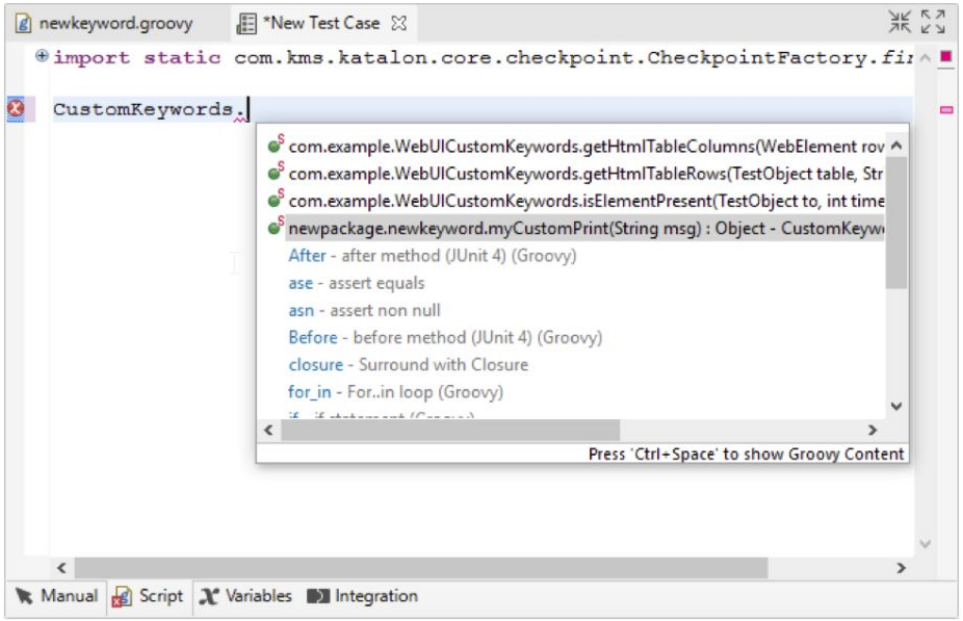
2. A new test step is added with a list of all defined custom keywords. Select your recently created keyword here.



CUSTOM KEYWORDS IN SCRIPTING VIEW

Follow the steps below to use your defined custom keywords in the ***Scripting view*** of a test case:

- *The **Class** CustomKeywords allows you to access all custom keywords. Enter the following syntax into the script editor: CustomKeywords.*
- *The **Content Assist** function will be invoked immediately after you type the **dot** character. **Content Assist** provides context-sensitive suggestions for code completion. Therefore, all the custom keywords defined in your test project will be displayed as below:*



CUSTOM KEYWORDS IN SCRIPTING VIEW

- *Select the recently created keyword and provide all parameters as required.*

The following API docs provide details on the functions used to work with custom keywords:

Driver Factory: getWebDriver() - Get the currently active web driver.

Test Object:

- addProperty(String name, ConditionType condition, String value) - Add a new property to a test object
- setProperties(List<TestObjectProperty> properties) - Set properties of a test object
- getObjectId() - Get object ID
- findPropertyValue(String name, boolean caseSensitive) - Find the value of a property using its name

CUSTOM KEYWORDS IN SCRIPTING VIEW

Keyword Util:

- logInfo(String message) - Log message
- markError(String message) - Mark a keyword as an error
- markErrorAndStop(String message) - Mark a keyword as an error and stop execution
- markFailed(String message)- Mark a keyword as failed and continue execution
- markFailedAndStop(String message) - Mark a keyword as failed and stop execution
- markPassed(String message) - Mark a keyword as passed
- markWarning(String message) - Mark a keyword as warning