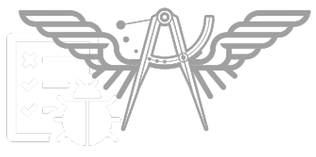
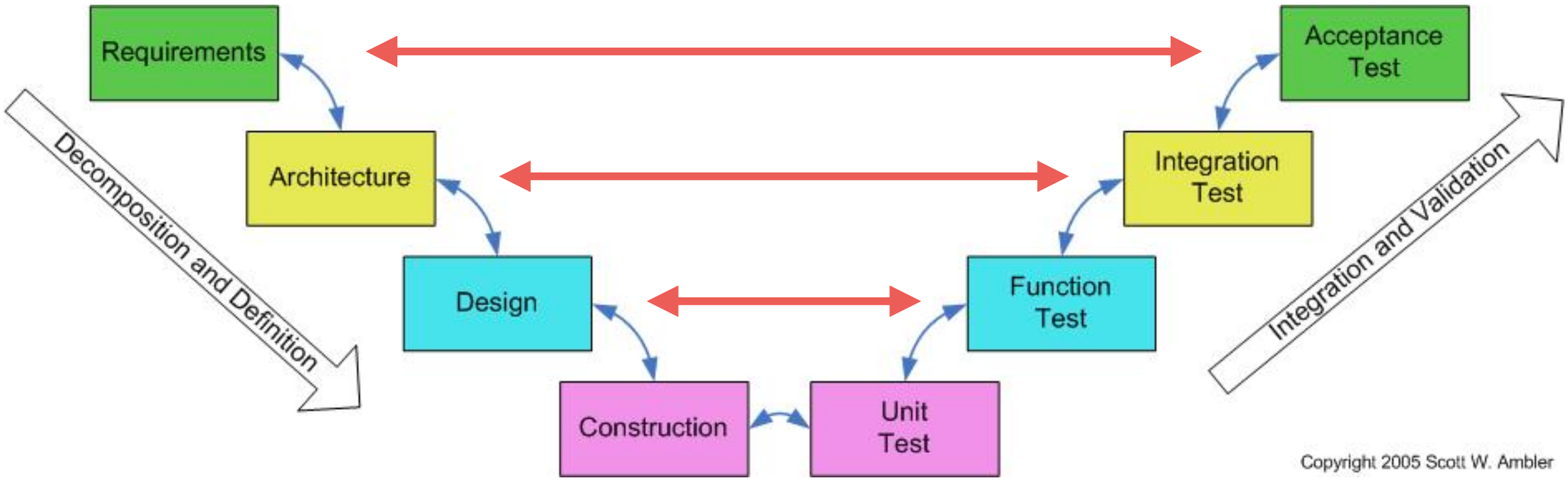
สยามชำนาญกิจ
SIAM CHAMNAN KIT

Automated testing with
Katalon and Robot framework

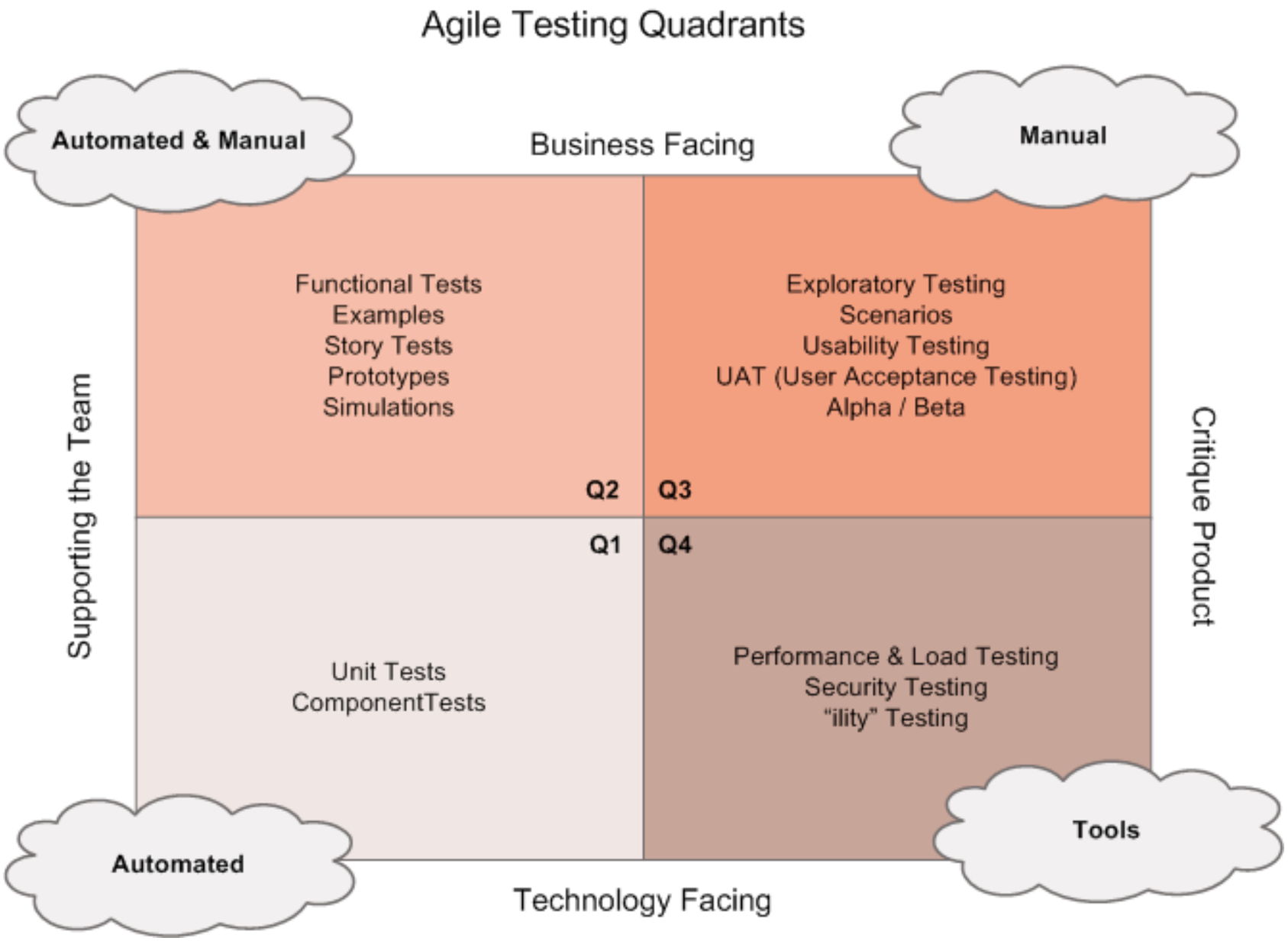


Automation Testing

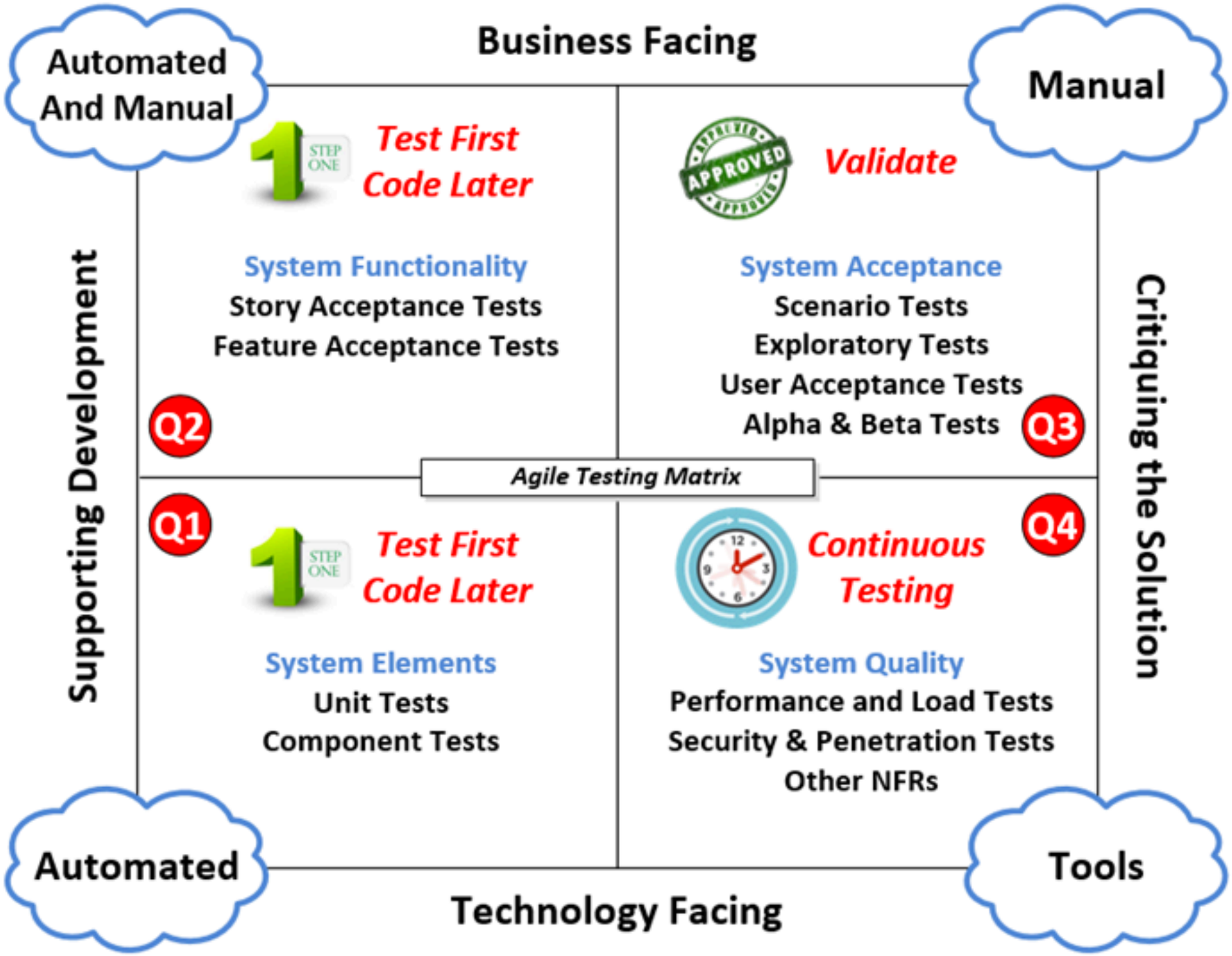
V Model



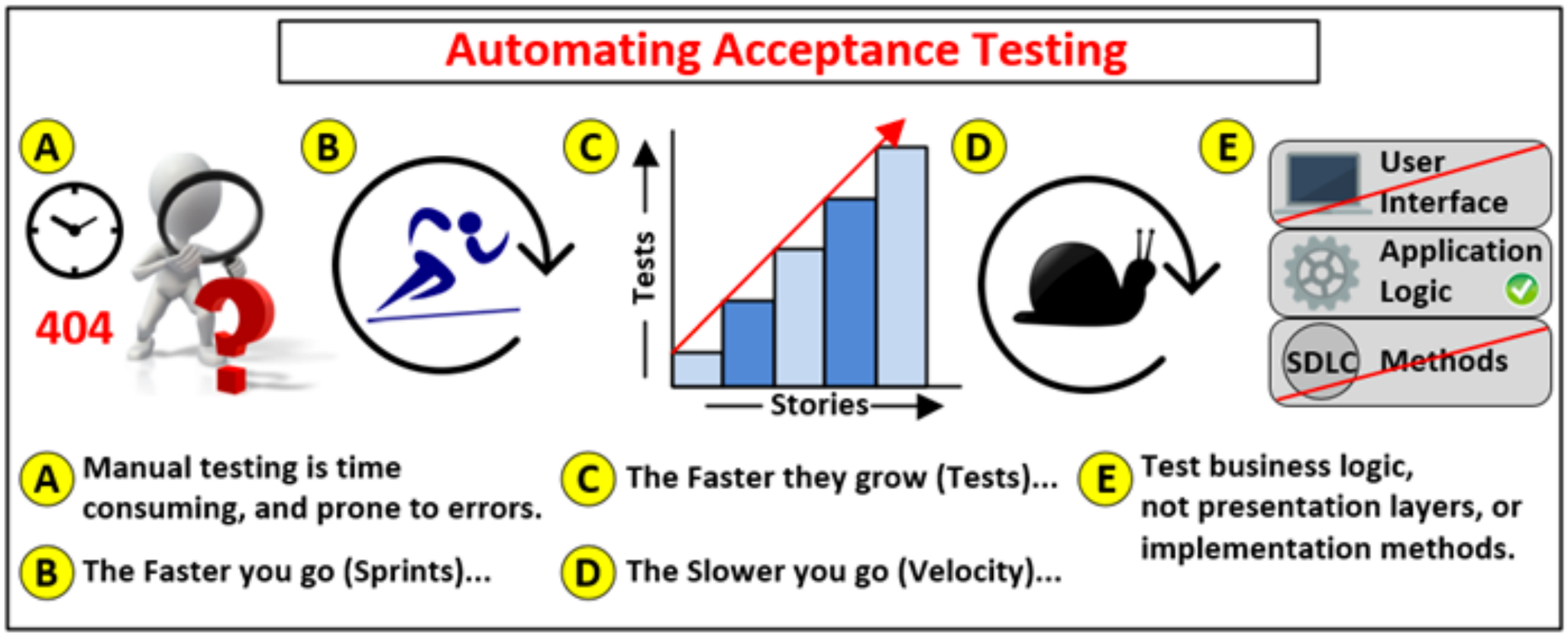
Agile Testing Quadrant



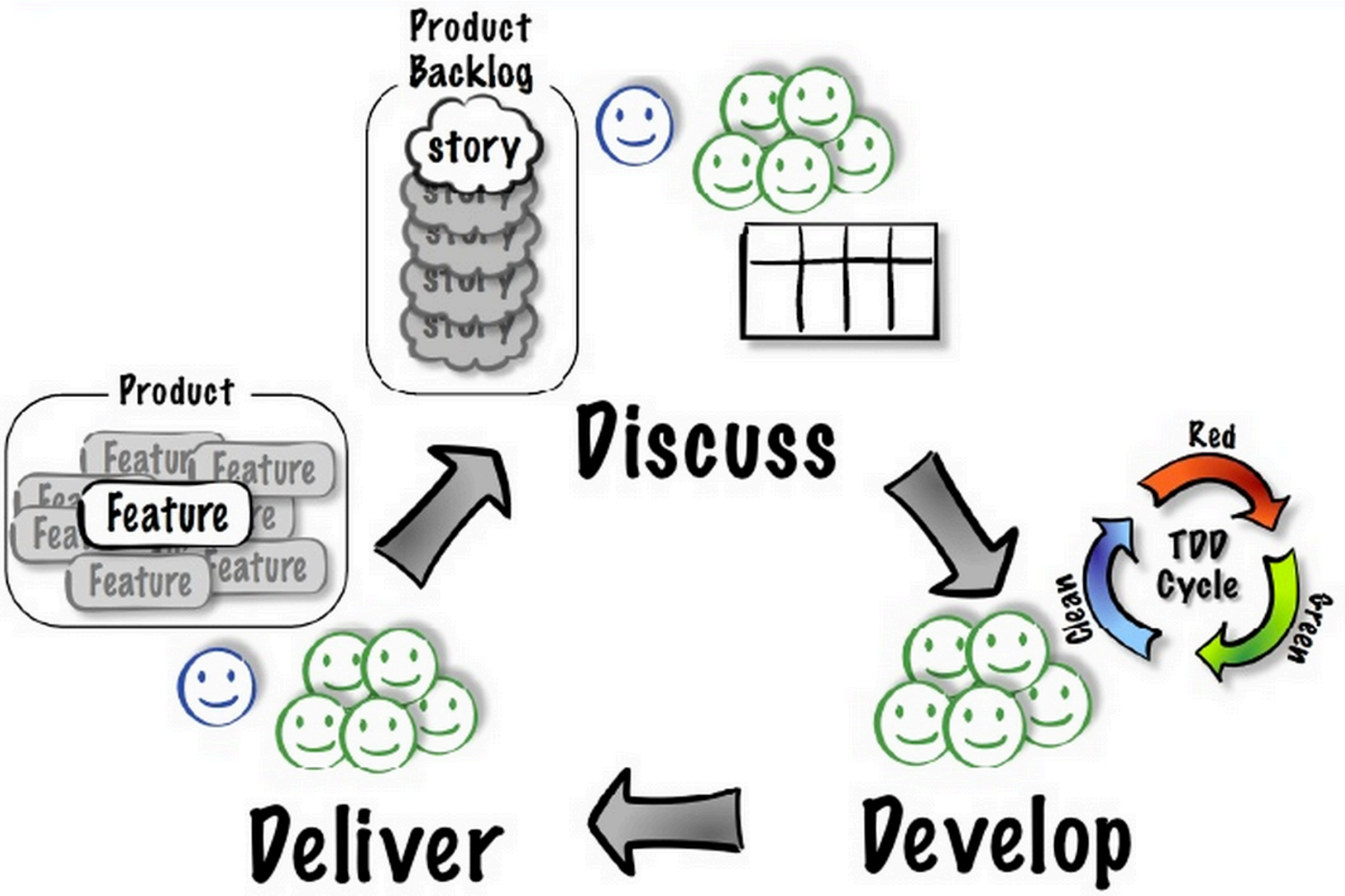
Agile Testing Quadrant



Test automation is essential



Acceptance Test-Driven Development



(Model developed with Pekka Klärck, Bas Vodde, and Craig Larman.)

ATDD

Common language
Common and share understanding
Executable requirements or examples
Living document

Acceptance Tests

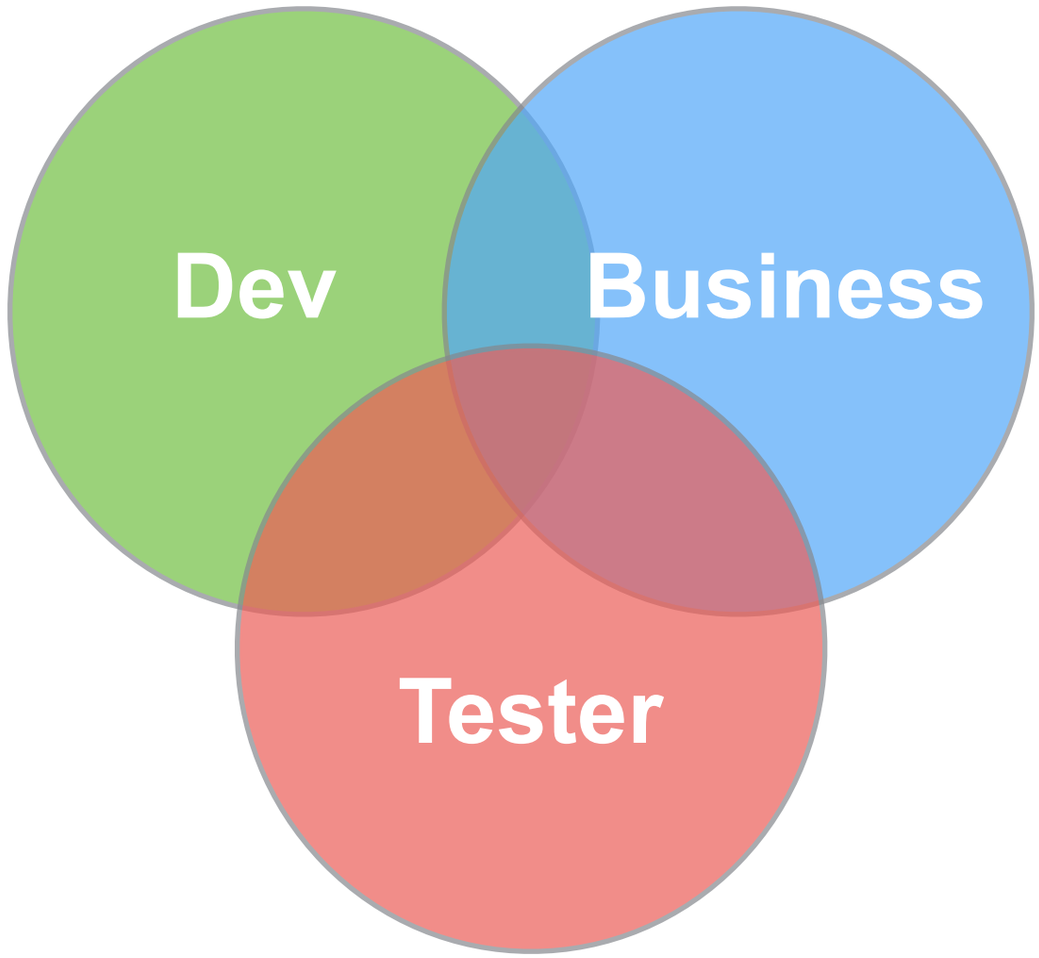
=

Business Criteria

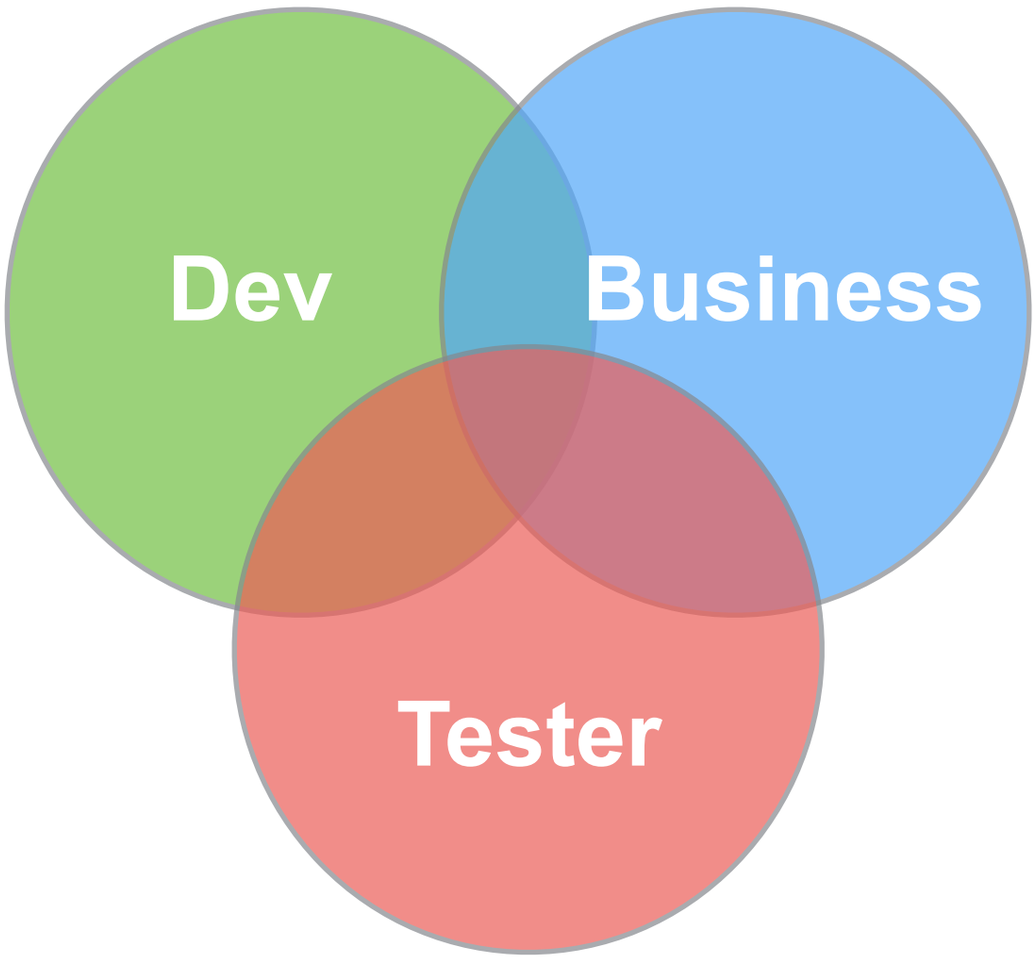
+

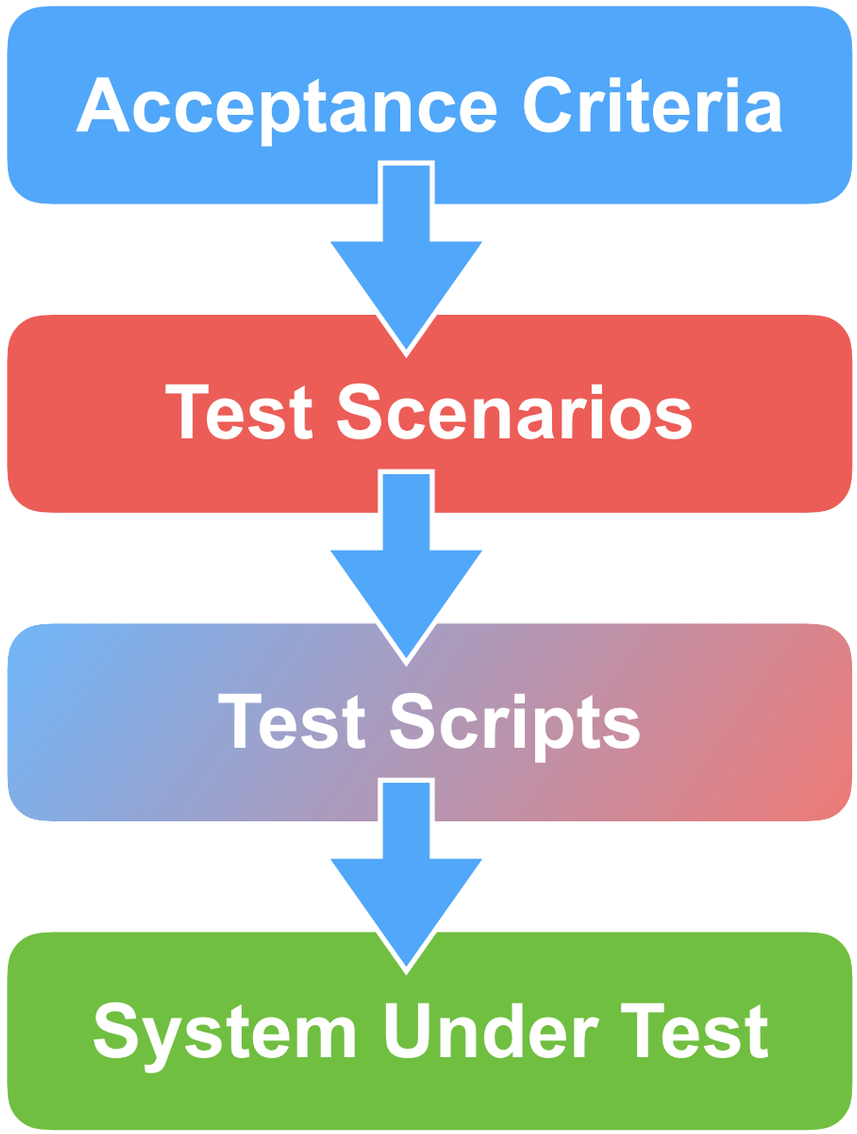
Examples (data)

Acceptance testing

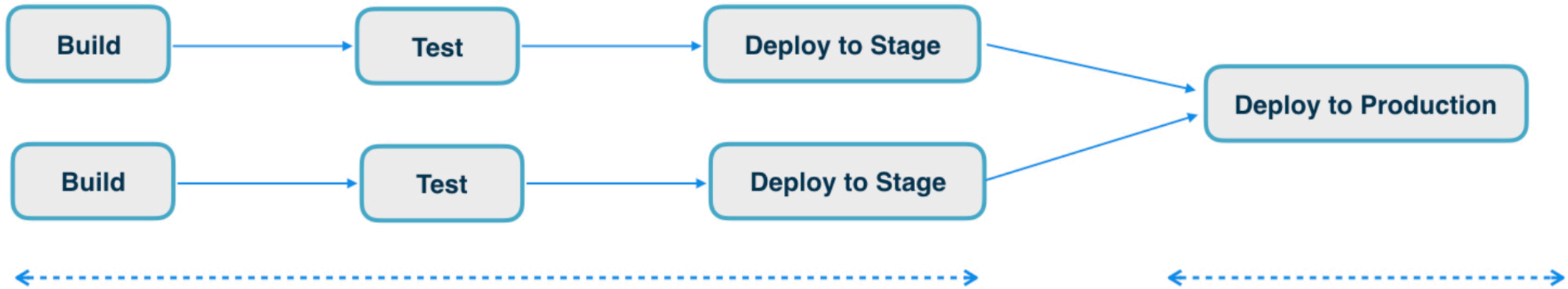


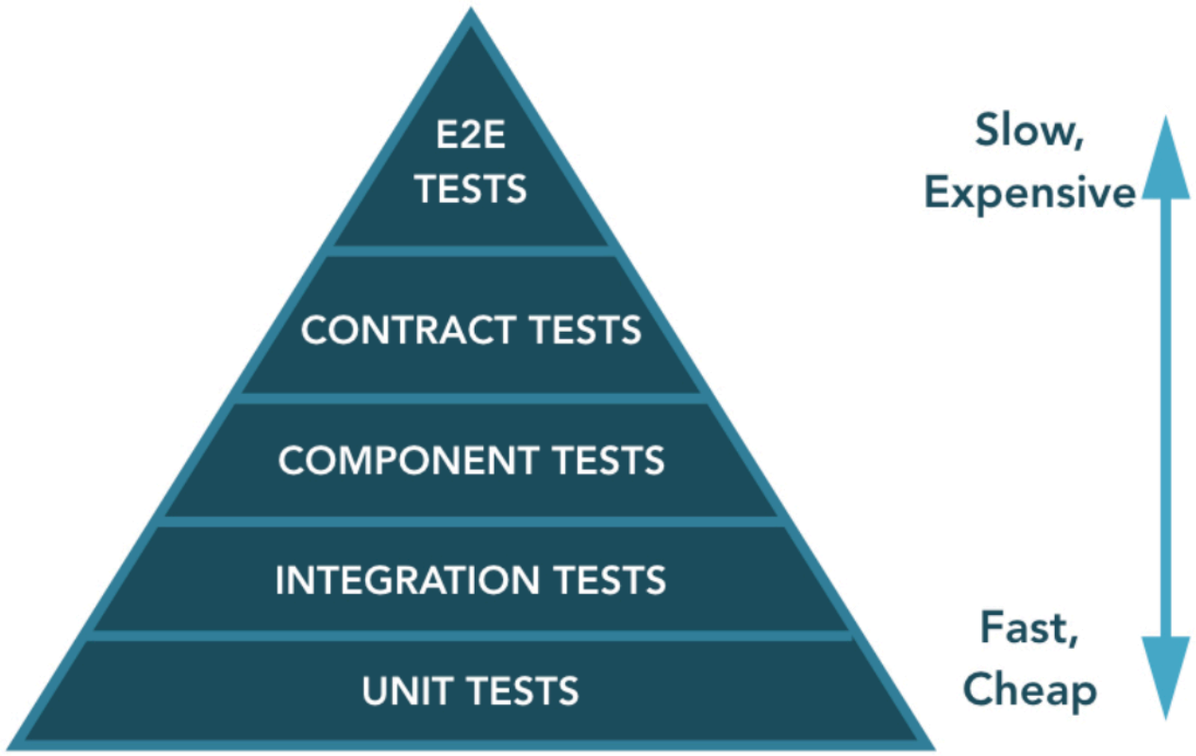
Acceptance testing





Test strategy

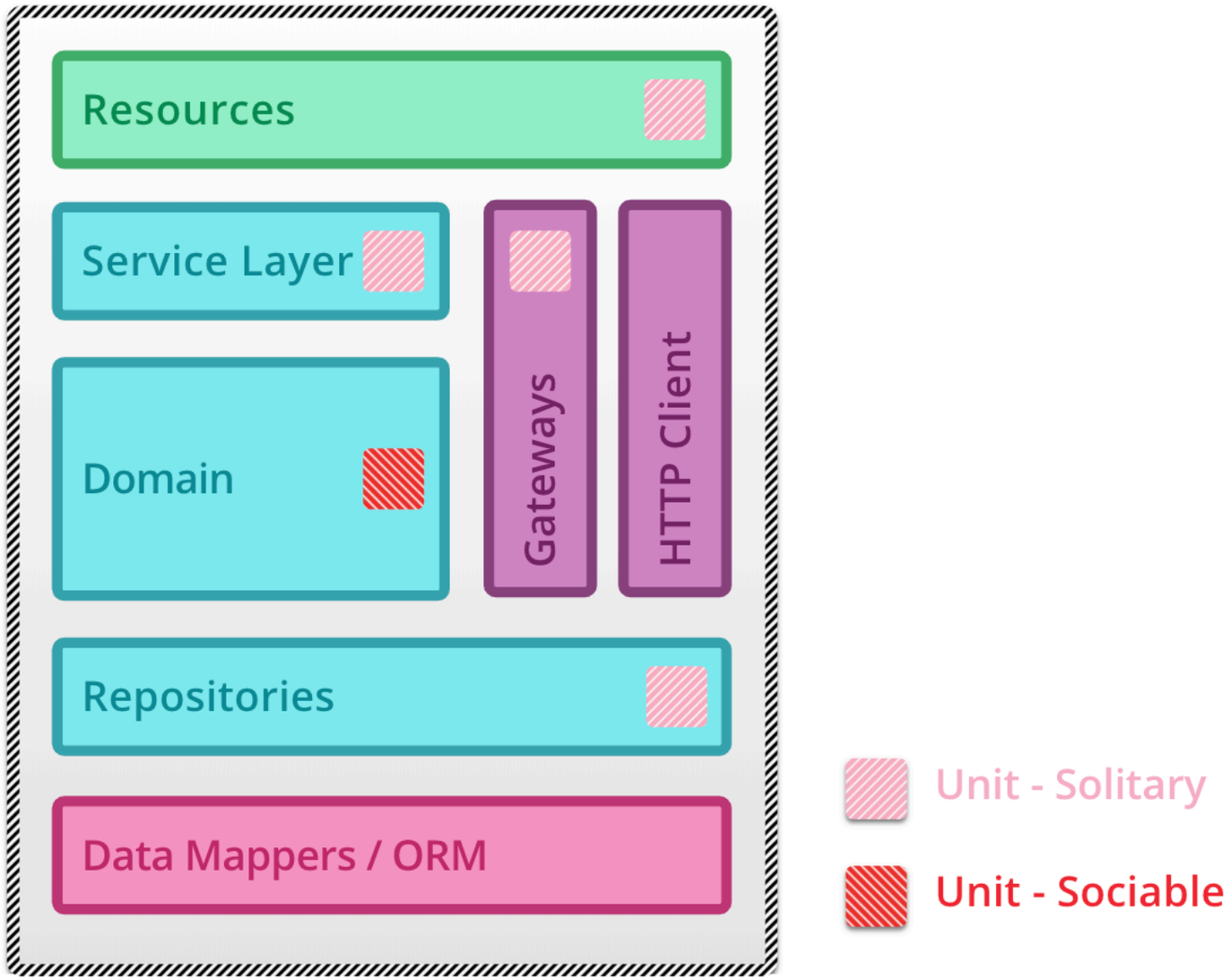




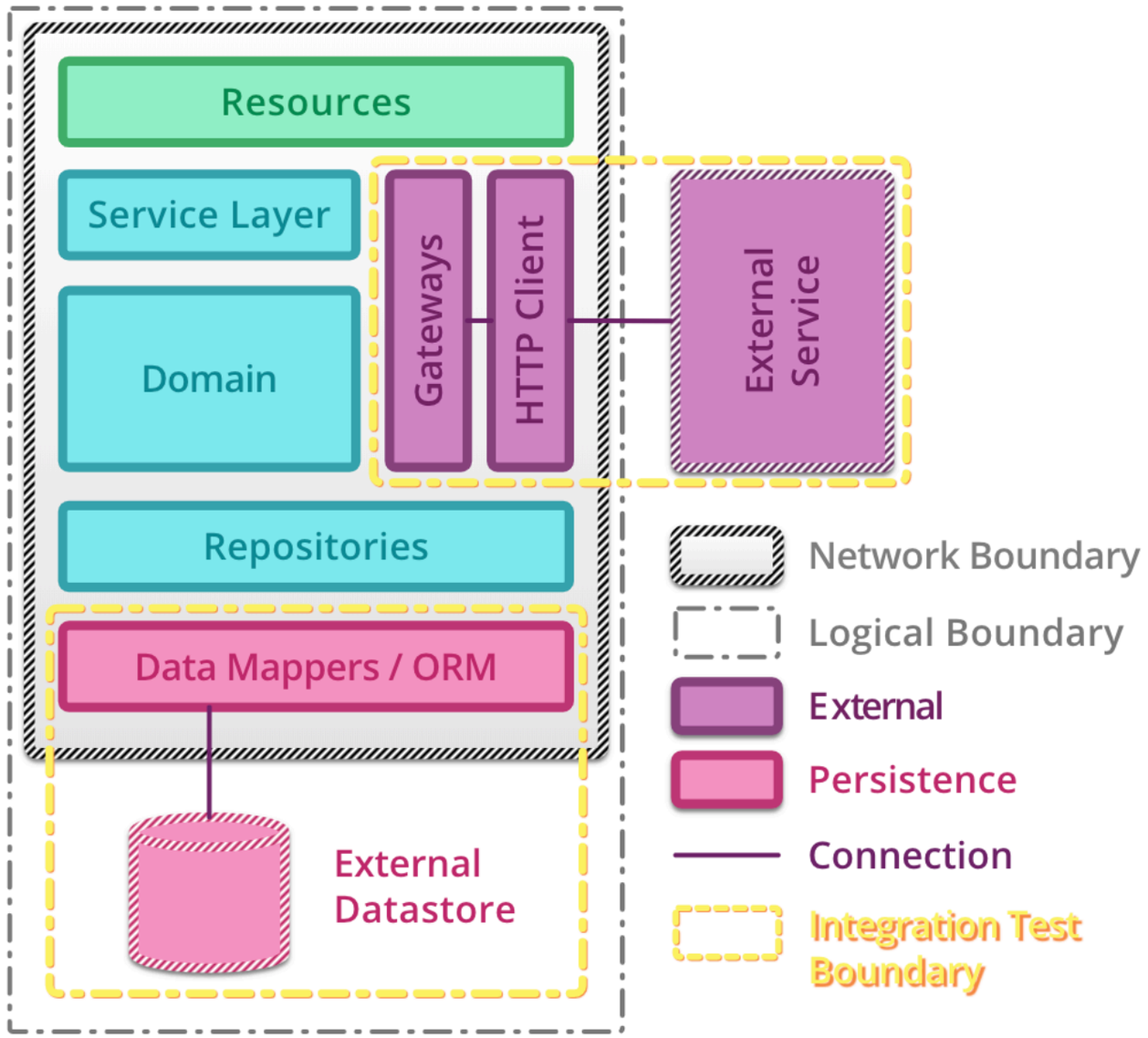
Monitoring
Distributed Tracing
Fault-injection Testing

Canary Deployments
Blue-Green Deployments
A/B Testing

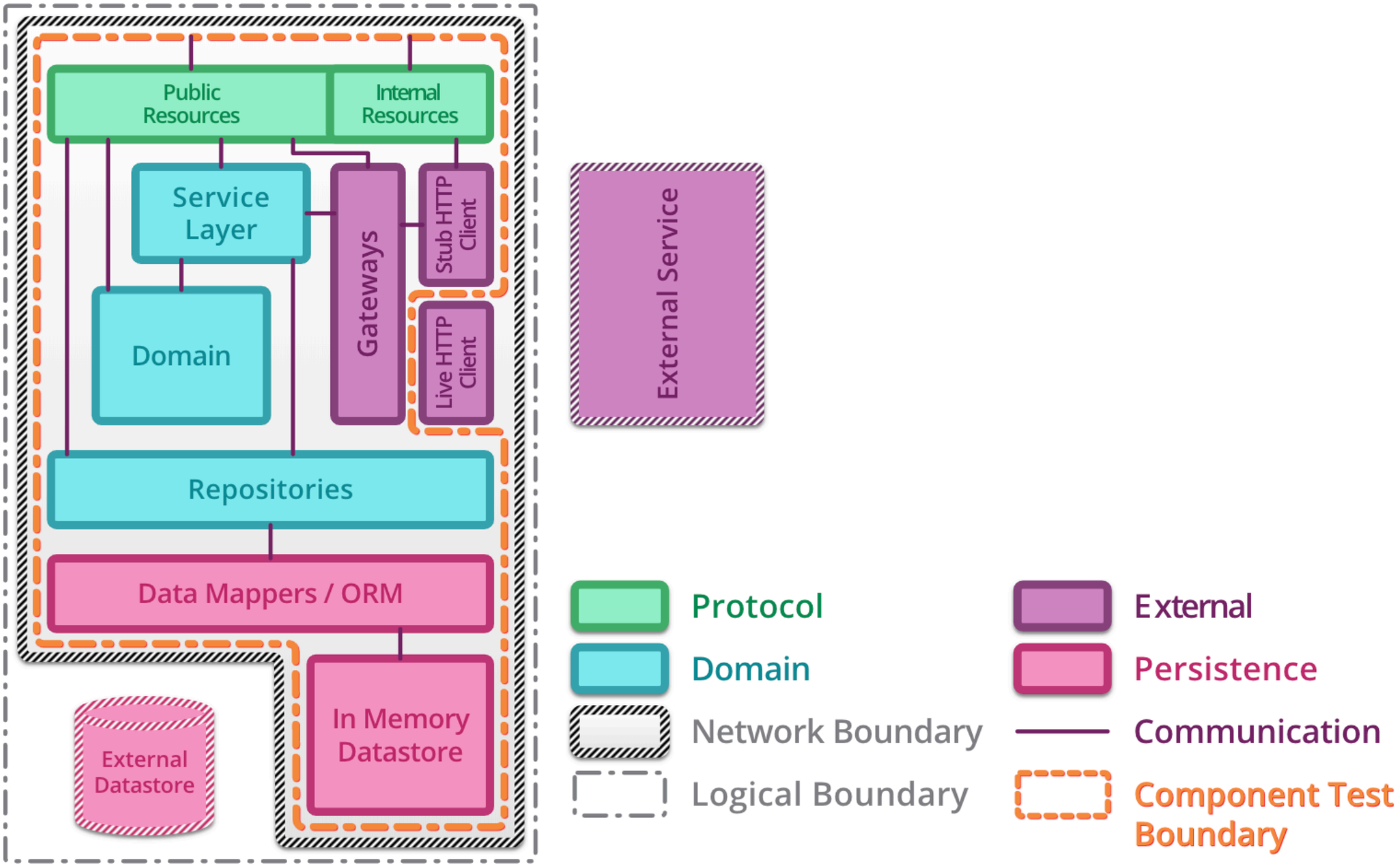
Unit testing



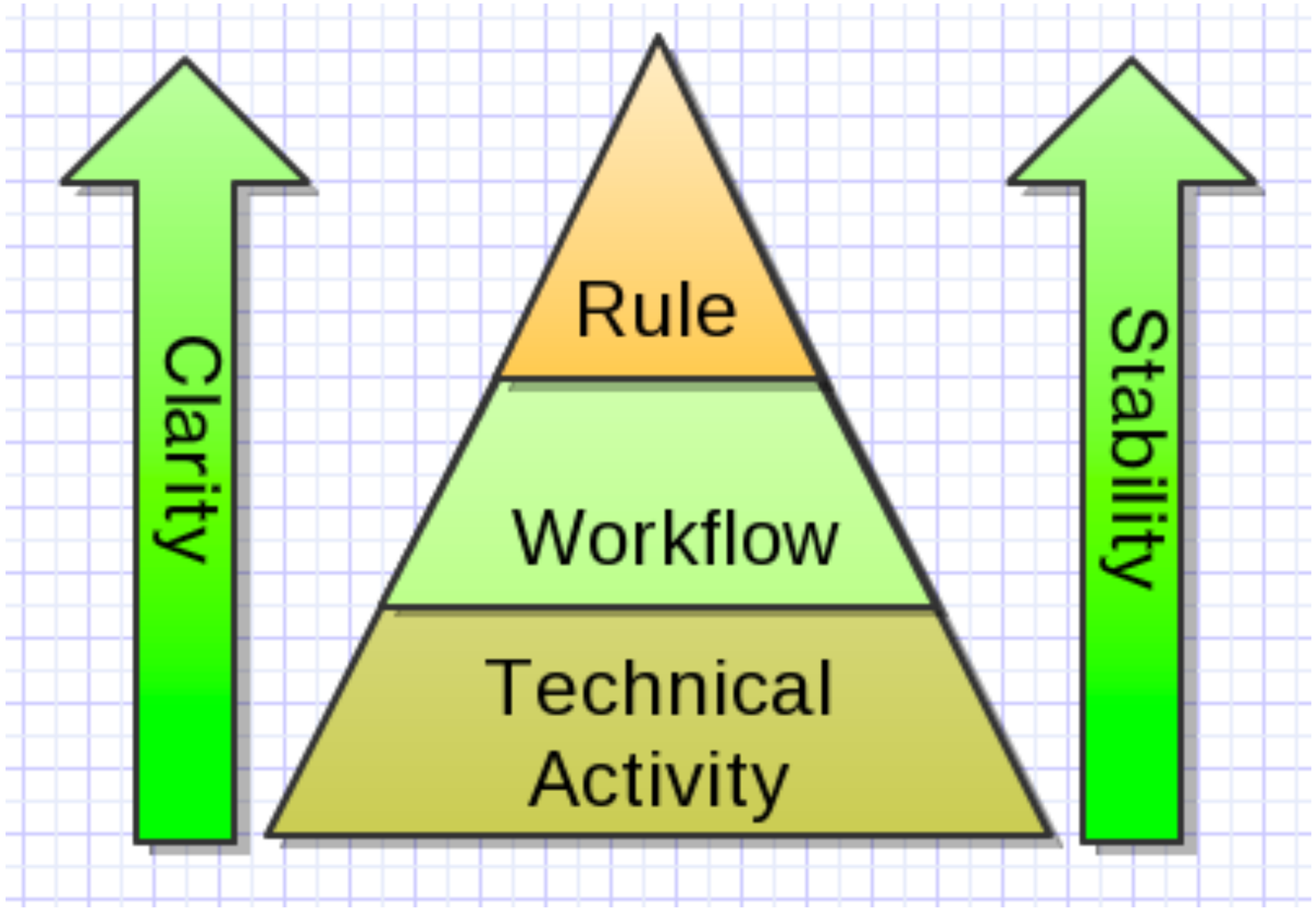
Integration testing



Component testing



3 levels of UI test automation



3 levels of UI test automation

Business rule/functionality level

what is this test demonstrating or exercising

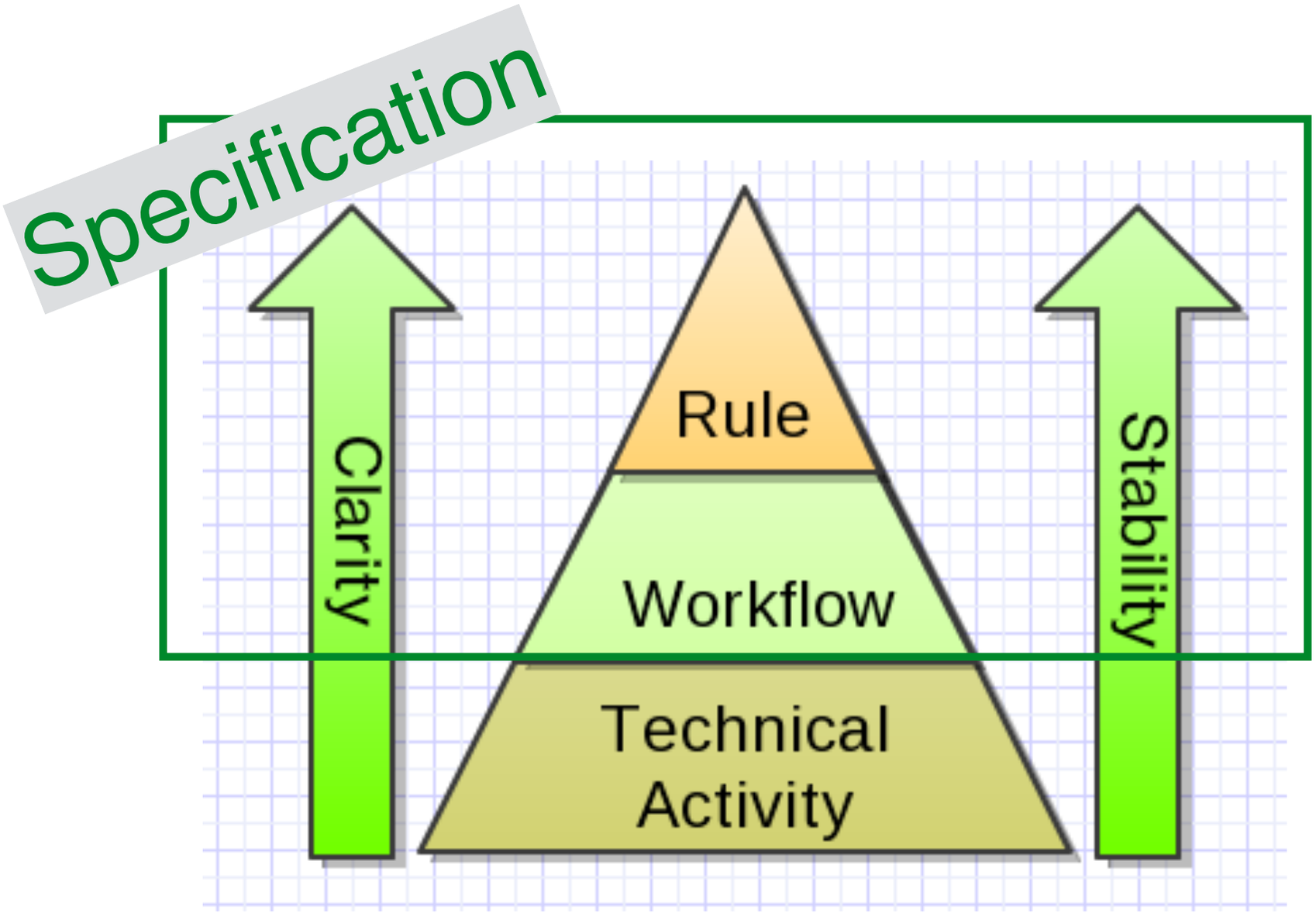
User interface workflow level

what does a user have to do to exercise the functionality through the UI

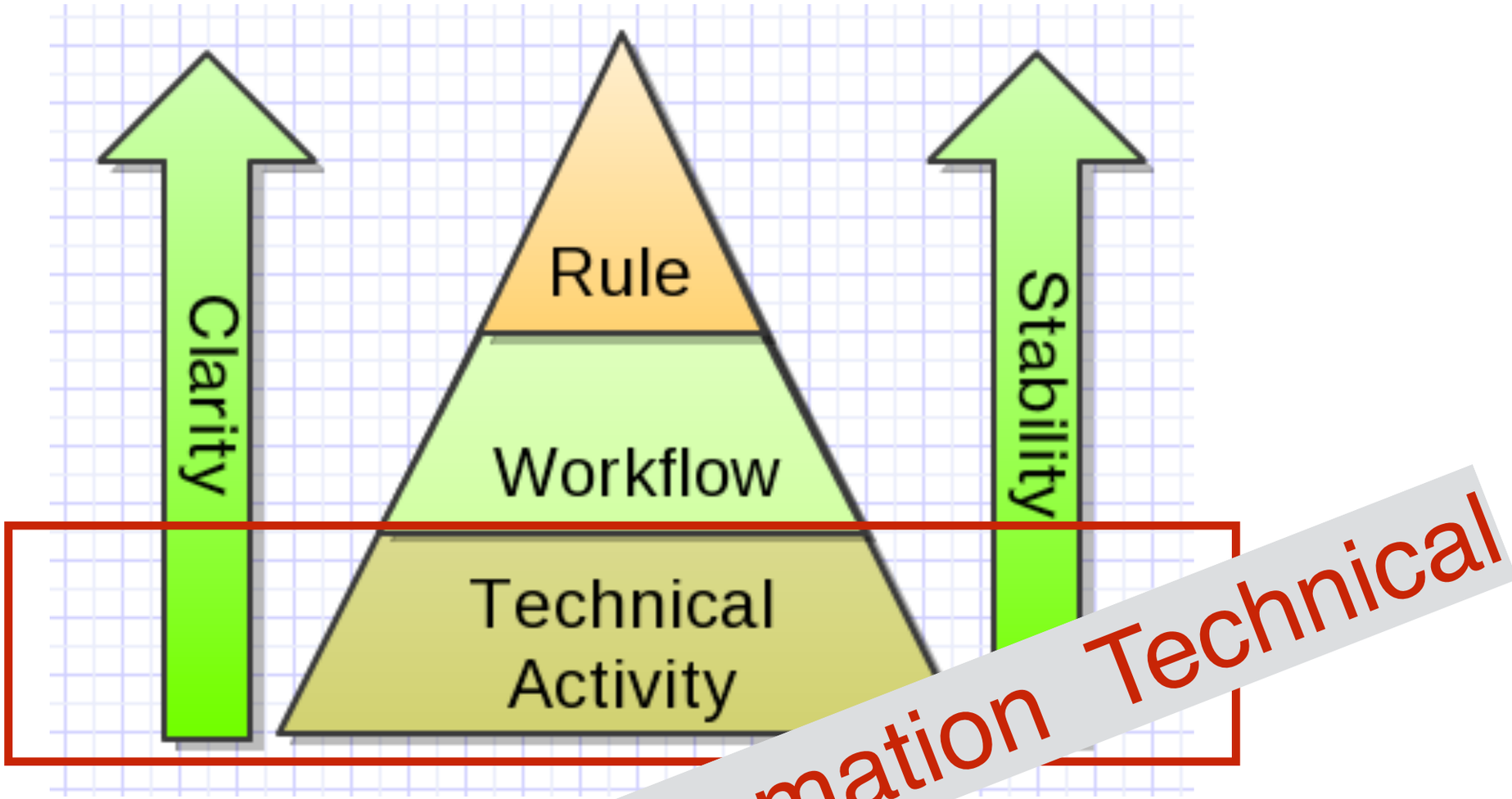
Technical activity level

what are the technical steps required to exercise the functionality

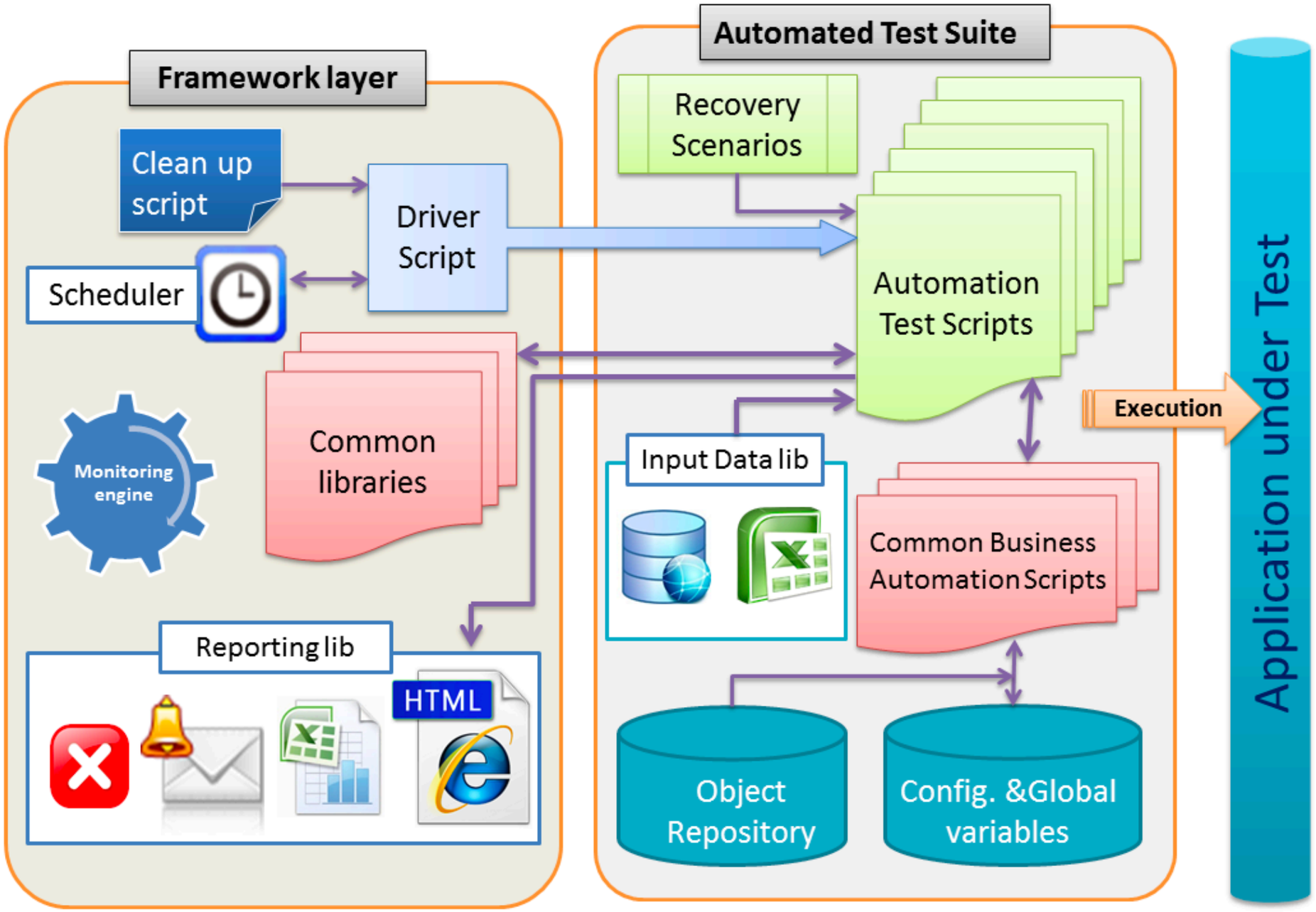
Good to start



Good to start

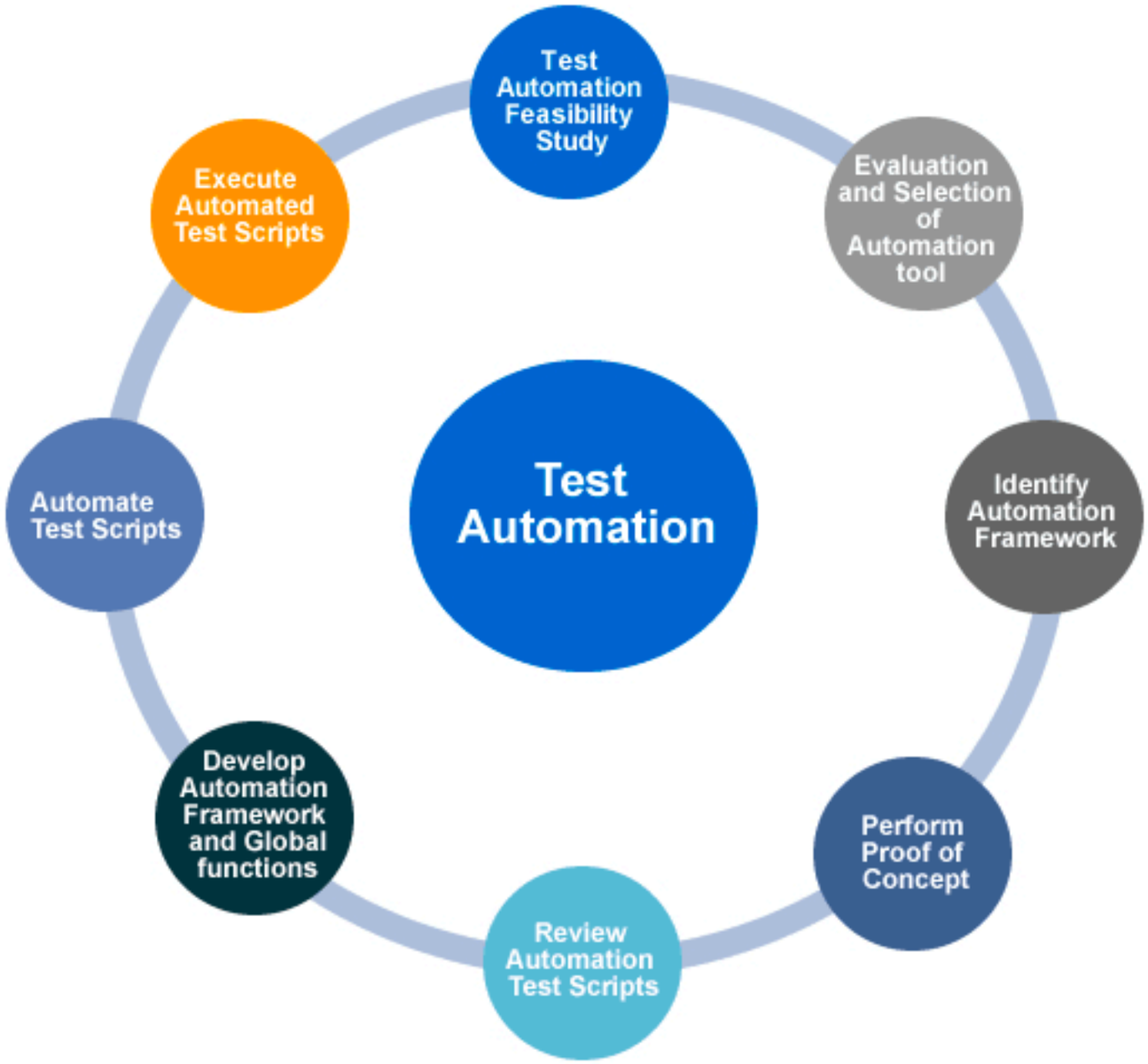


Test Framework



Test Automation strategy

Test Automation selection



Katalon

https://www.katalon.com/

Overview

Simple and powerful test automation
For web, API and Mobile testing
Easy to start project
Cross platforms support (Windows, mac, Linux)
Full featured to support small to enterprise team

Katalon features

Katalon Solutions

Authoring component

- Record and edit
- IDE
- Keyword
- Objects
- Data
- Test scripts

Execution component

- Test structure
- CI tools
- Selenium
- Appium
- Test results

Report and management

- Log and report engine
- Test Case Management (TCM)

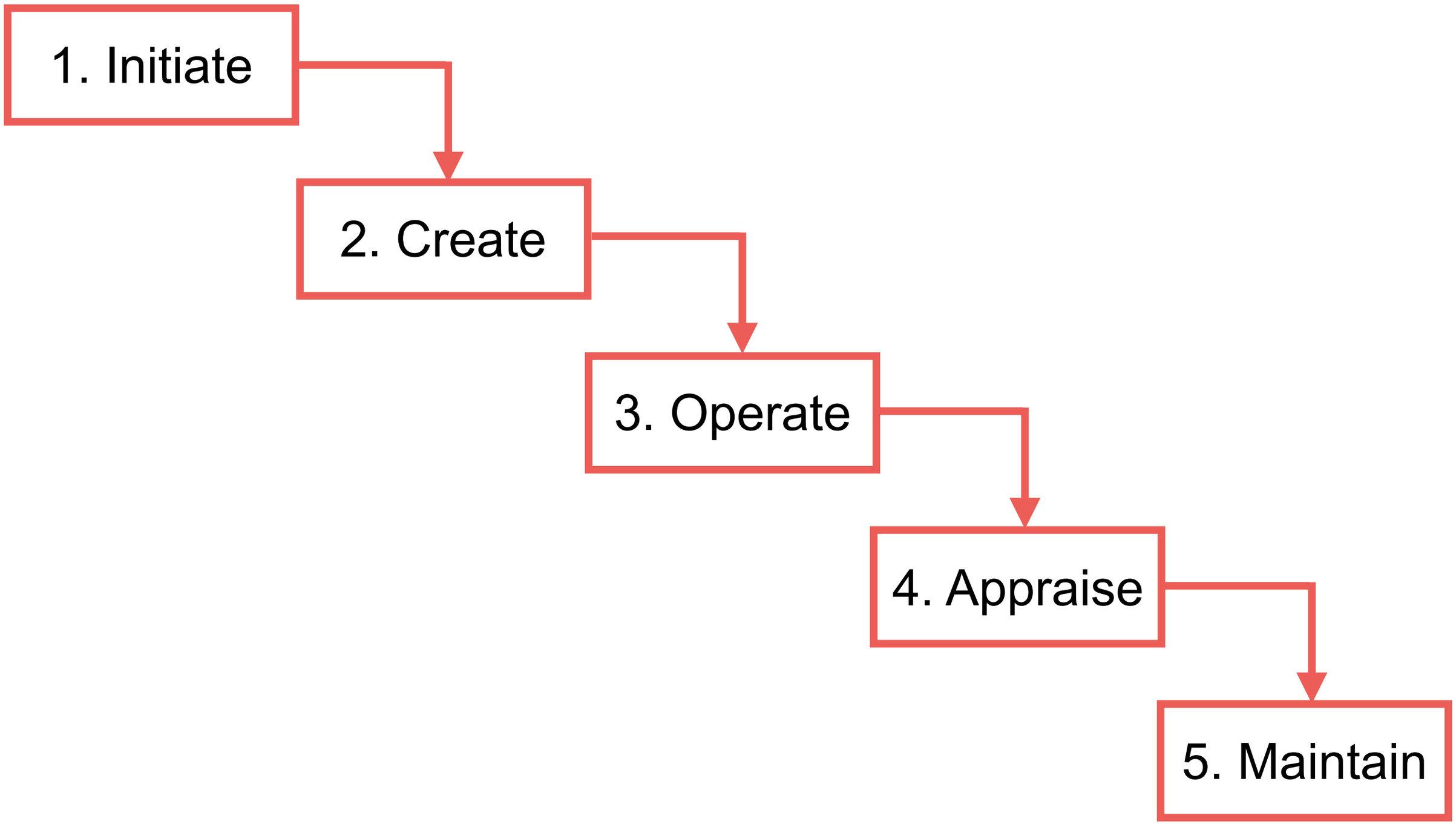
Key features

Test case management
Usability and functionality
Test data management and data-driven
Recording and test generation
Build-in keywords
Report and analytics
Integration (Git/CI/JIRA)

Automation painpoints

Complexity frameworks
Experience and expertise
Reusability of scripts

Katalon workflow



1. Initiate

Intrinsic project templates

Building templates for organising test case, object, keyword

Multiple capabilities

Fullt supports web, API and mobile testing

Hassle-free tool merging

Easy in integrate with Jenkins, Git, JIRA and plug-ins

2. Create

Automatic test generation

Record actions and generate scripts automatically

Hi-end scripting

Enable building advanced test scripts or customise keywords

Intuitive object capturing

Advance recorder detects object properties

3. Operate

Test execution made powerful

Run test case and test suite with multiple config and data sets

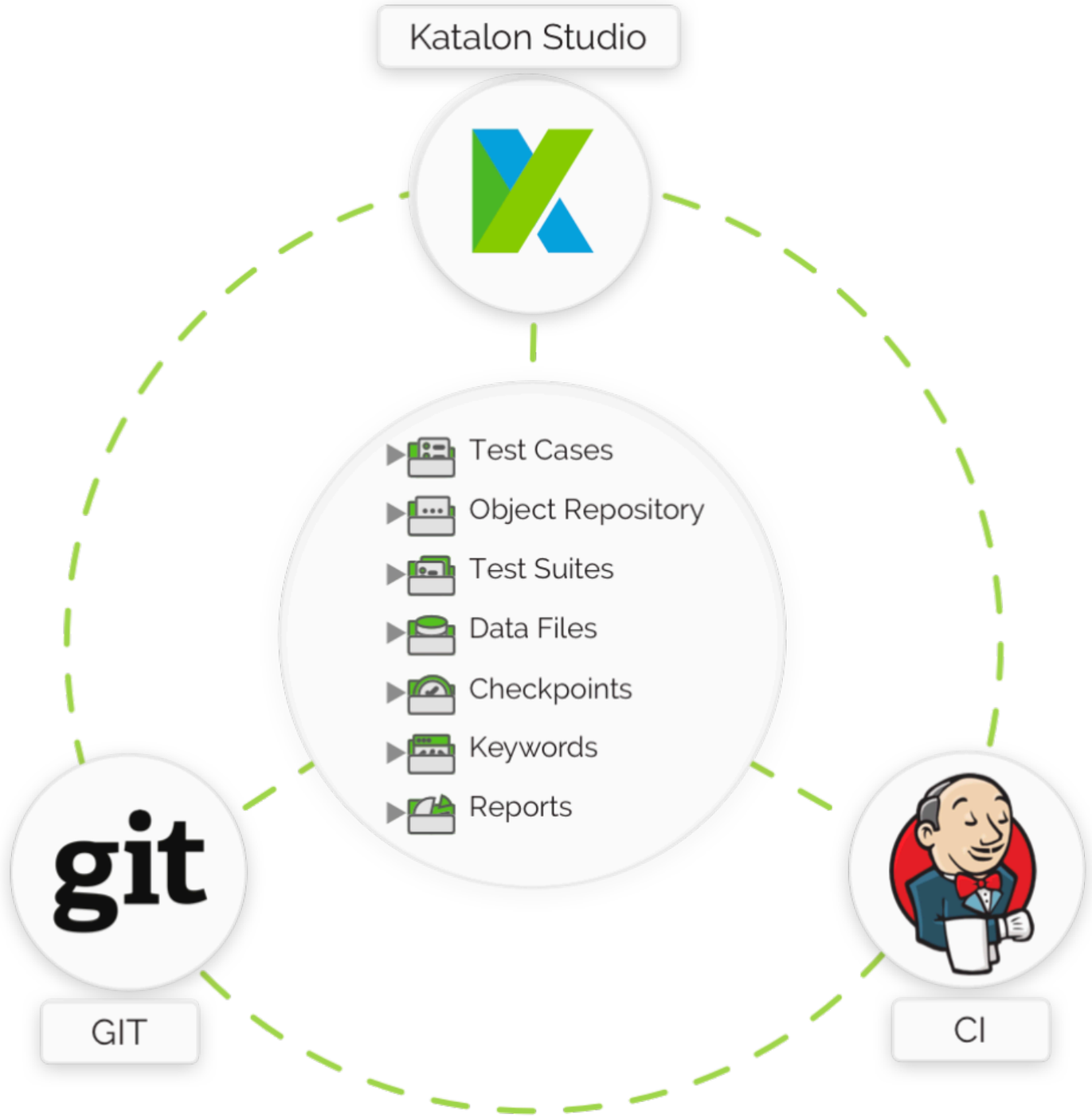
Versatility in execution

Provide CI integration console, remote execution

On-the-go failure handling and auto re-execution

Include runtime rules to handle complex execution flows

Katalon Integration



4. Appraise

Reports available in several formats

Advance logging, debug and screenshots

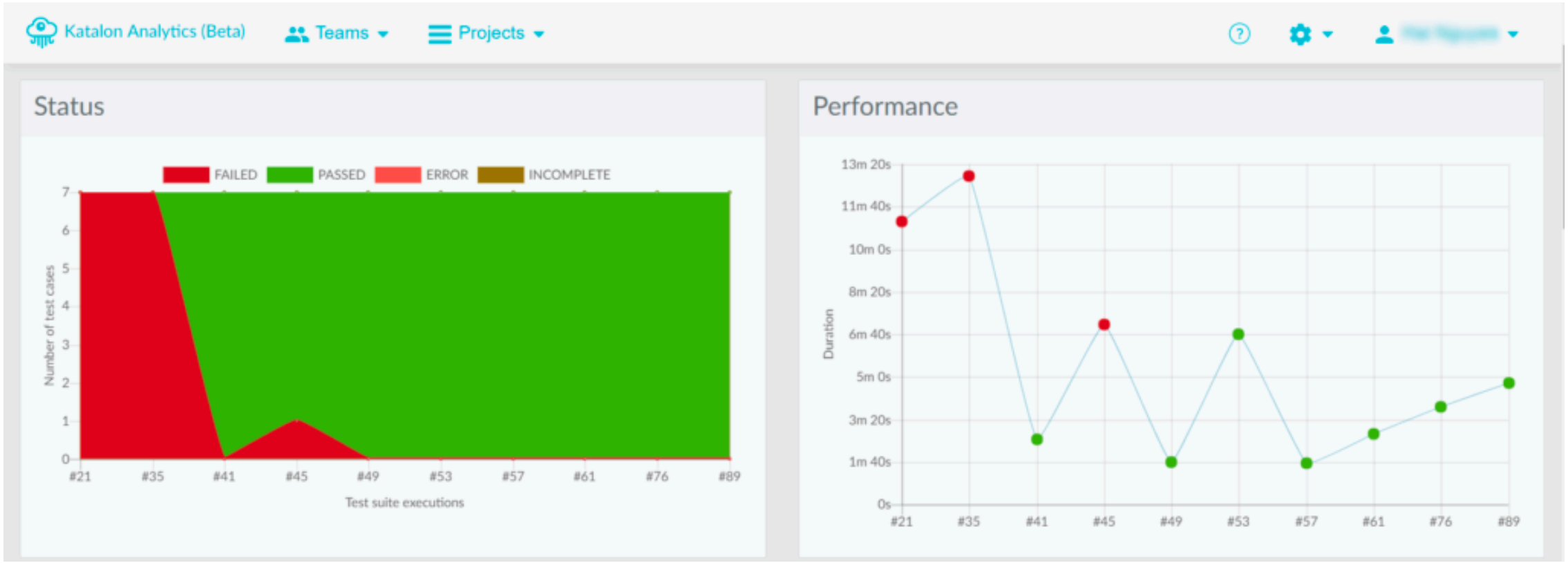
Bespoke execution reports

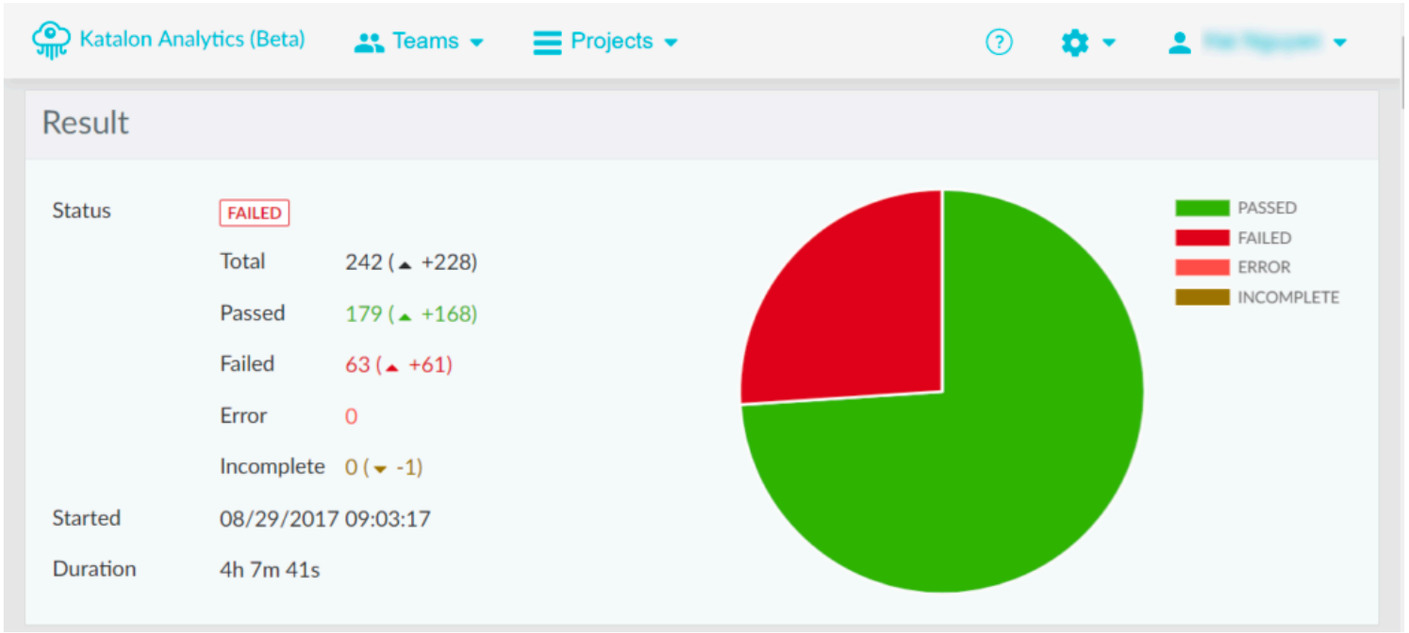
Integrated with your notification workflow

Enhanced Selenium and Appium logs

To improve analysis features to improve automation strategy

Katalon Analytic/Reporting





5. Maintain

Intelligent test object maintenance

Auto update all association test cases and suites
when object changed

Efficient test organization

Make easy to maintain and manage of tests, datas, keywords

Easy team collaboration

Integrate with Git to allow team members
to share artefacts and workload

Let’s start with Katalon

Katalon Studio

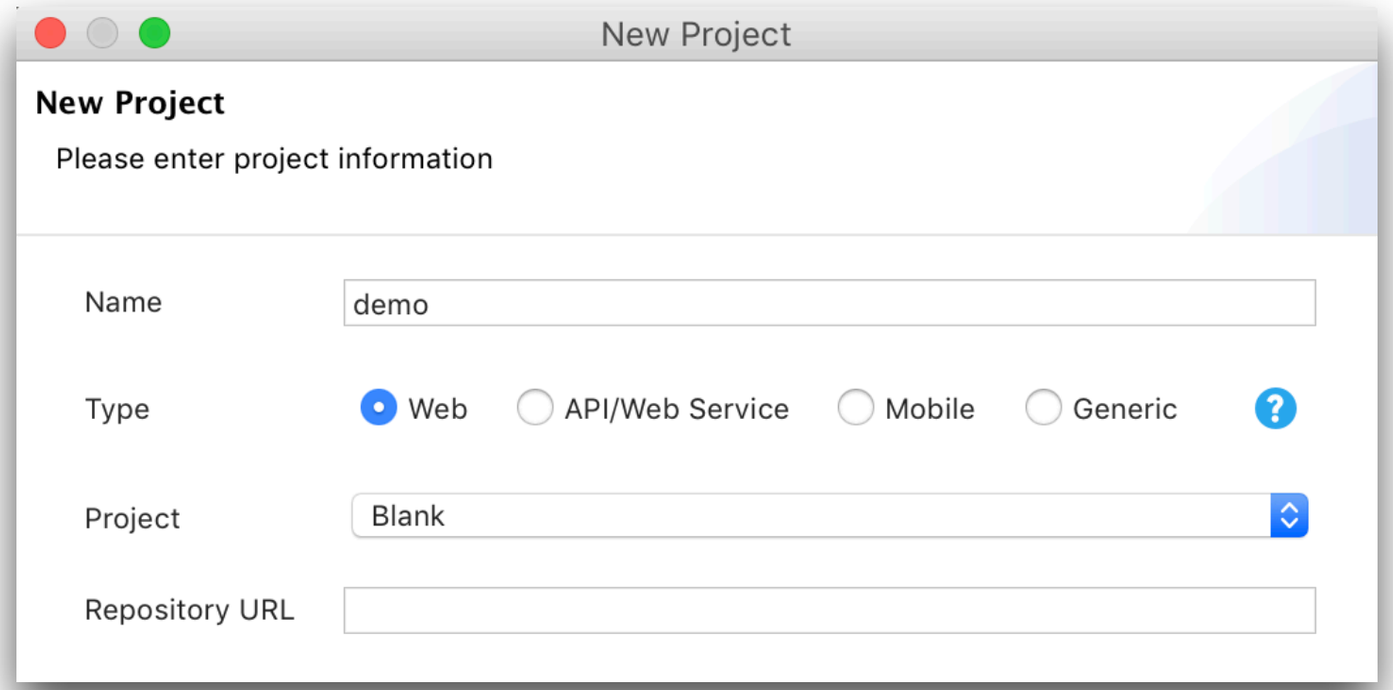
Java (JDK)
Android SDK
Web driver for web browsers
Plugins/dependencies
Appium for mobile testing (not included)

Katalon Studio

Helpful tool for automation teams
QA/Tester tools
Less effort to setup and execute tests

Types of project

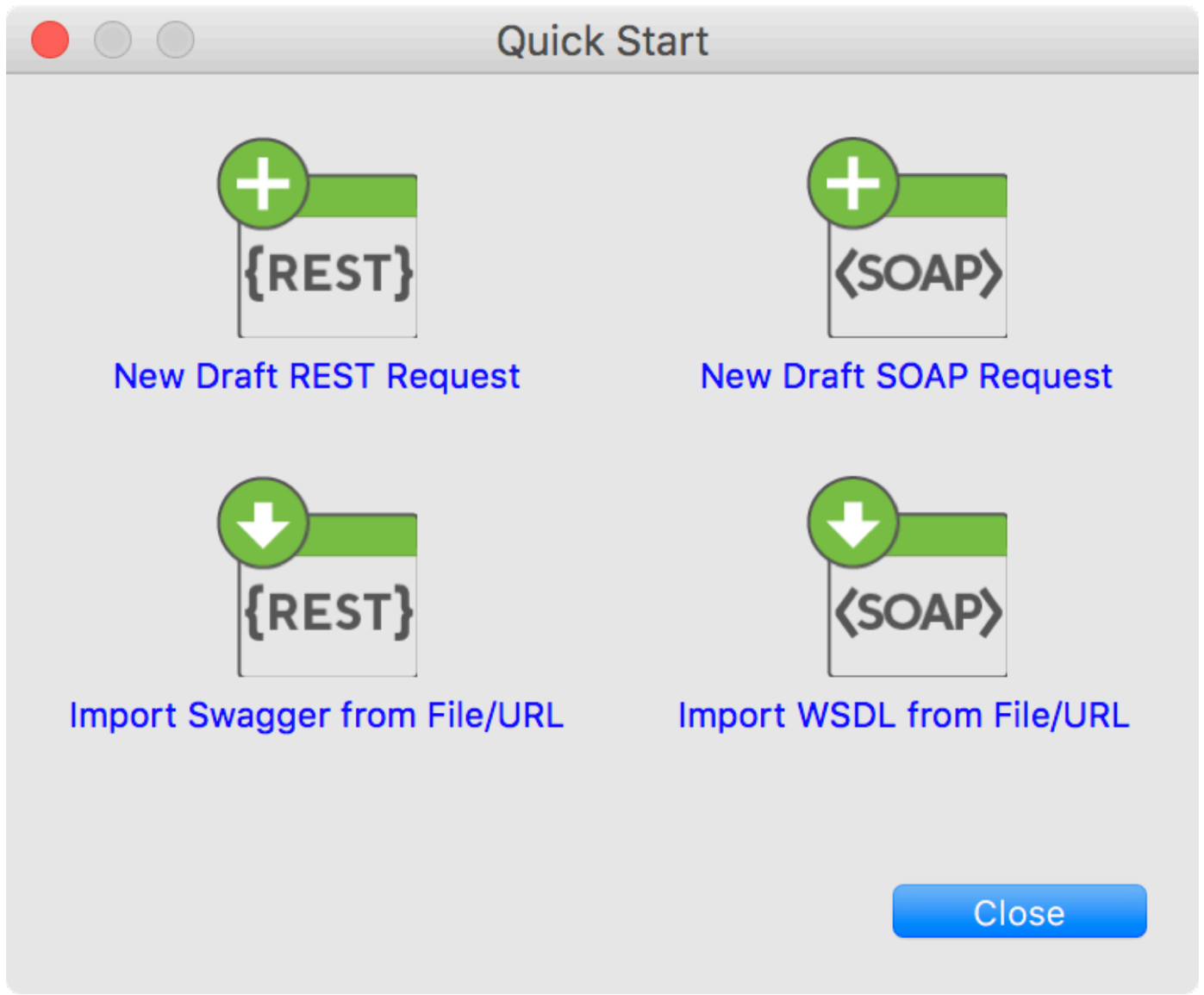
Web
API/Web Service
Mobile
Generic



Web

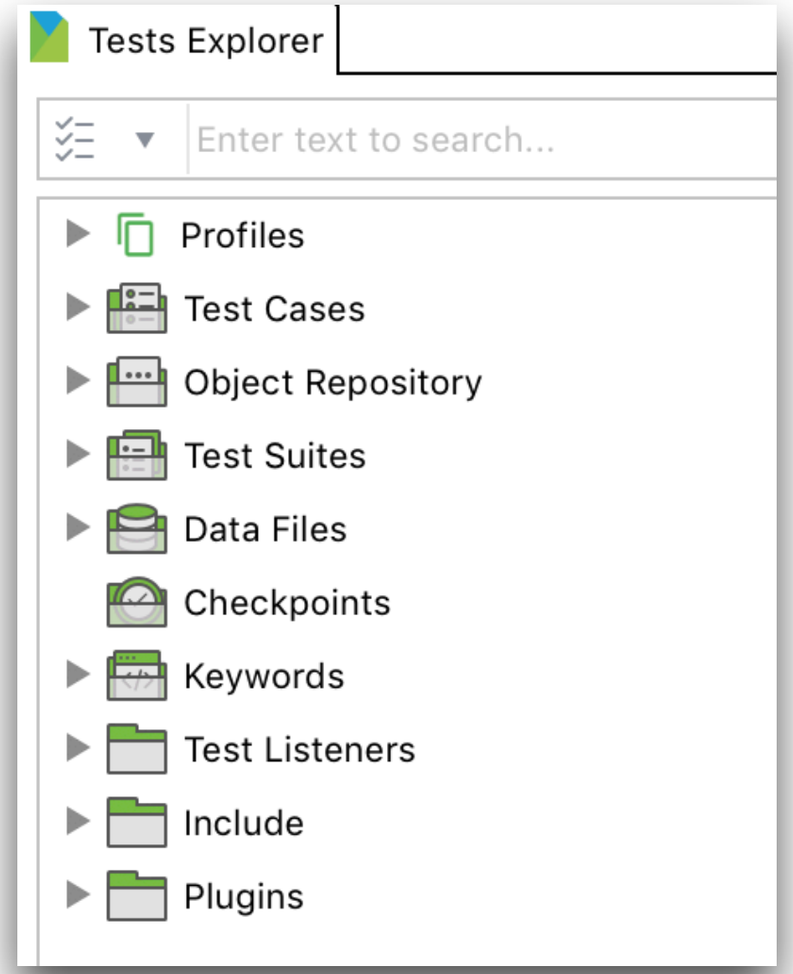
Record web UI test cases
Spy objects
Scripting

API/Web Service



Tests Explorer

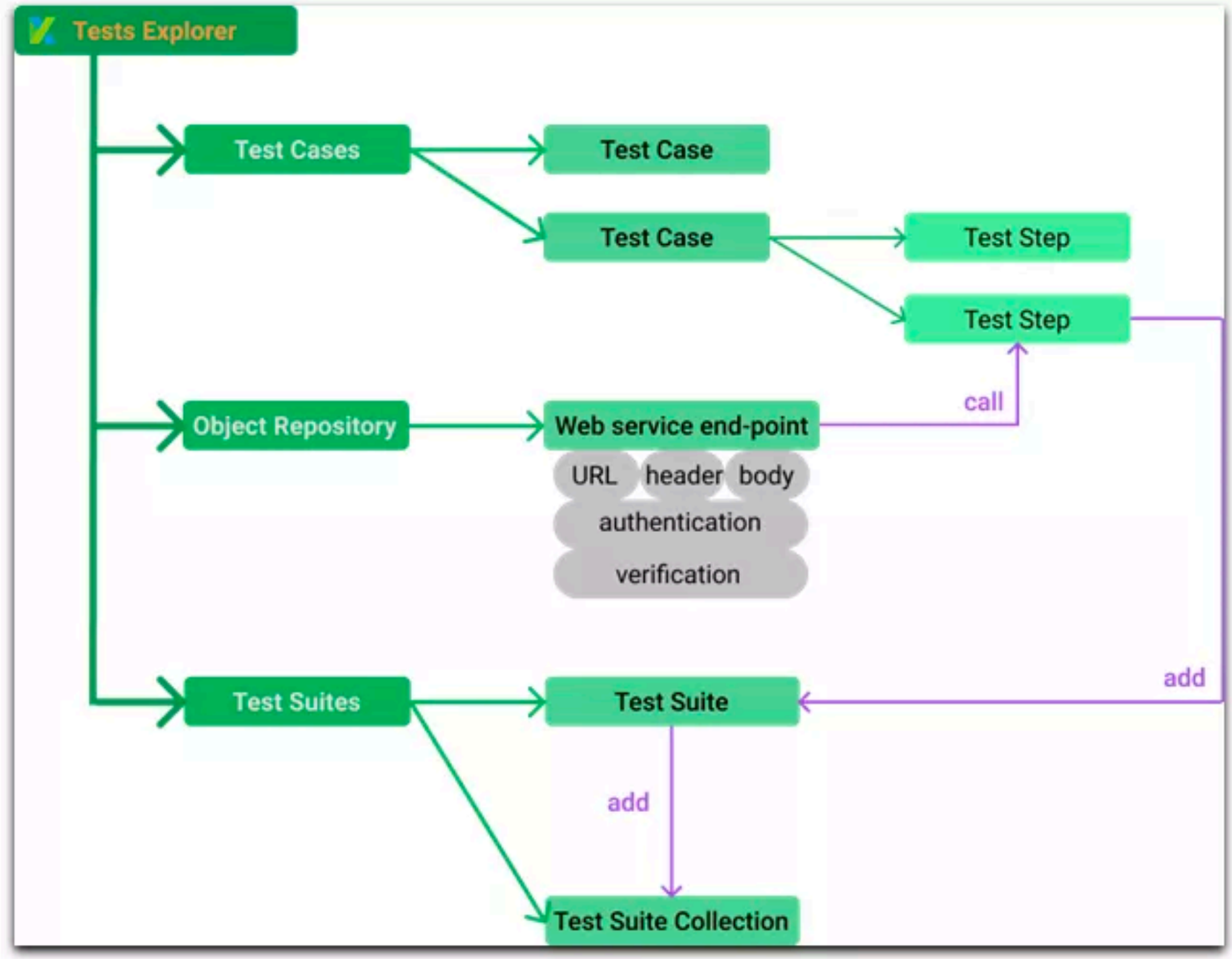
Allow to browse the structure of projects
Access all artefacts quickly



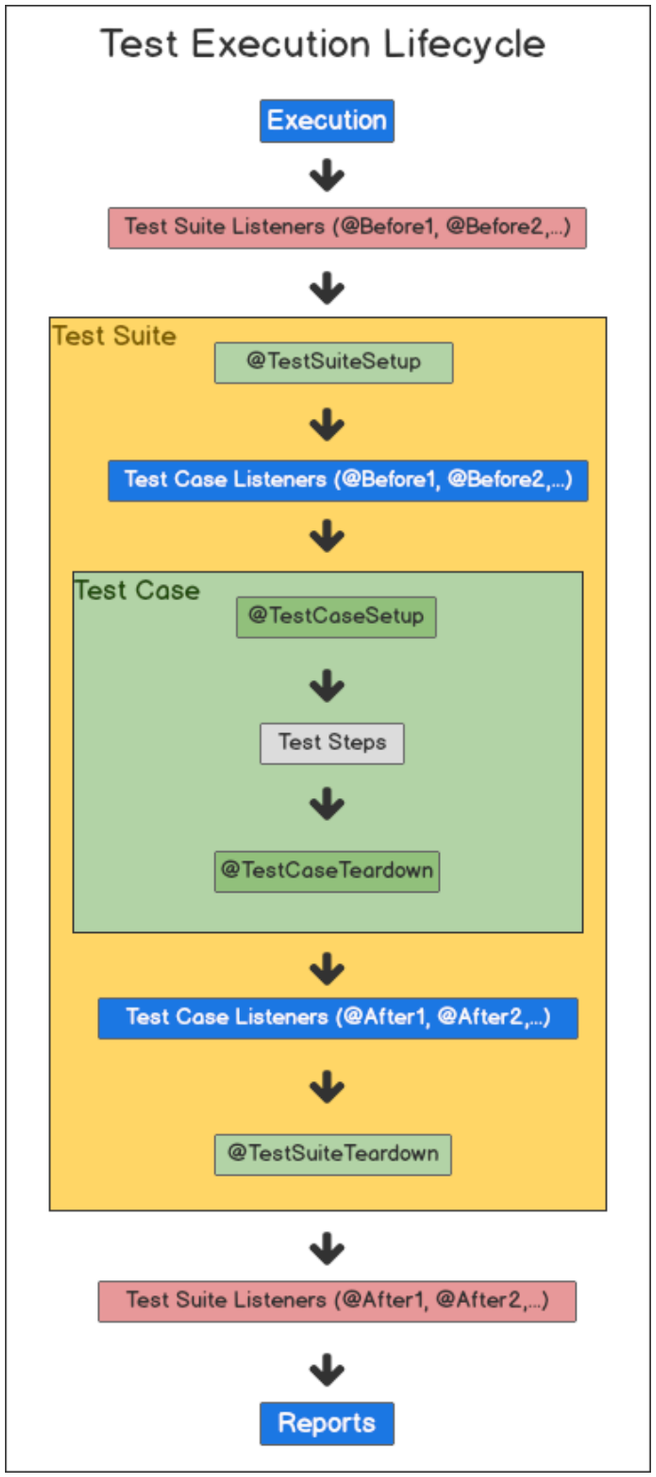
Vocabulary of Katalon

Test cases
Object repository
Test suites
Data files
Checkpoints
Keywords
Test listeners (life cycle)
Includes
Plugins

Project structure

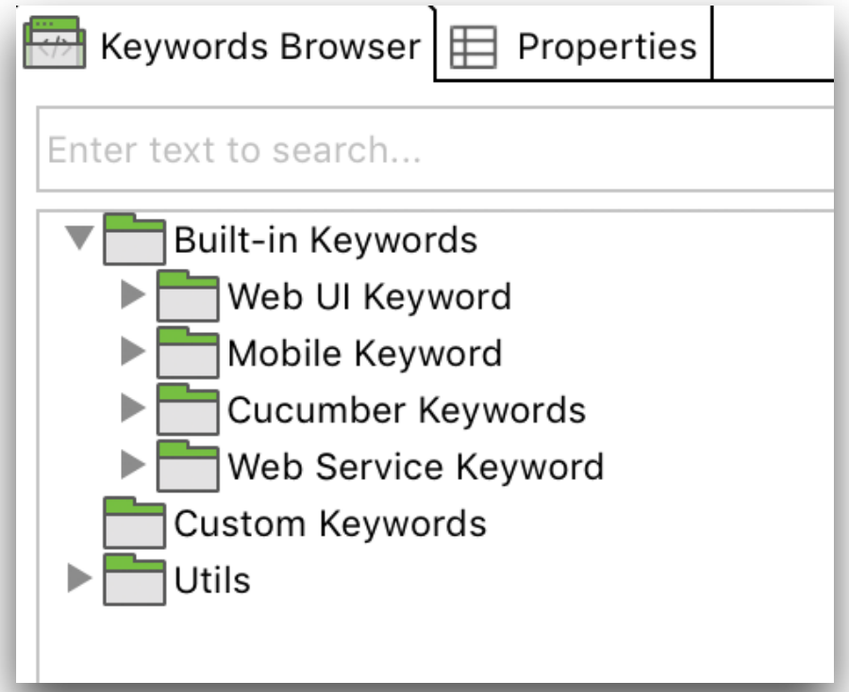


Test Life Cycle

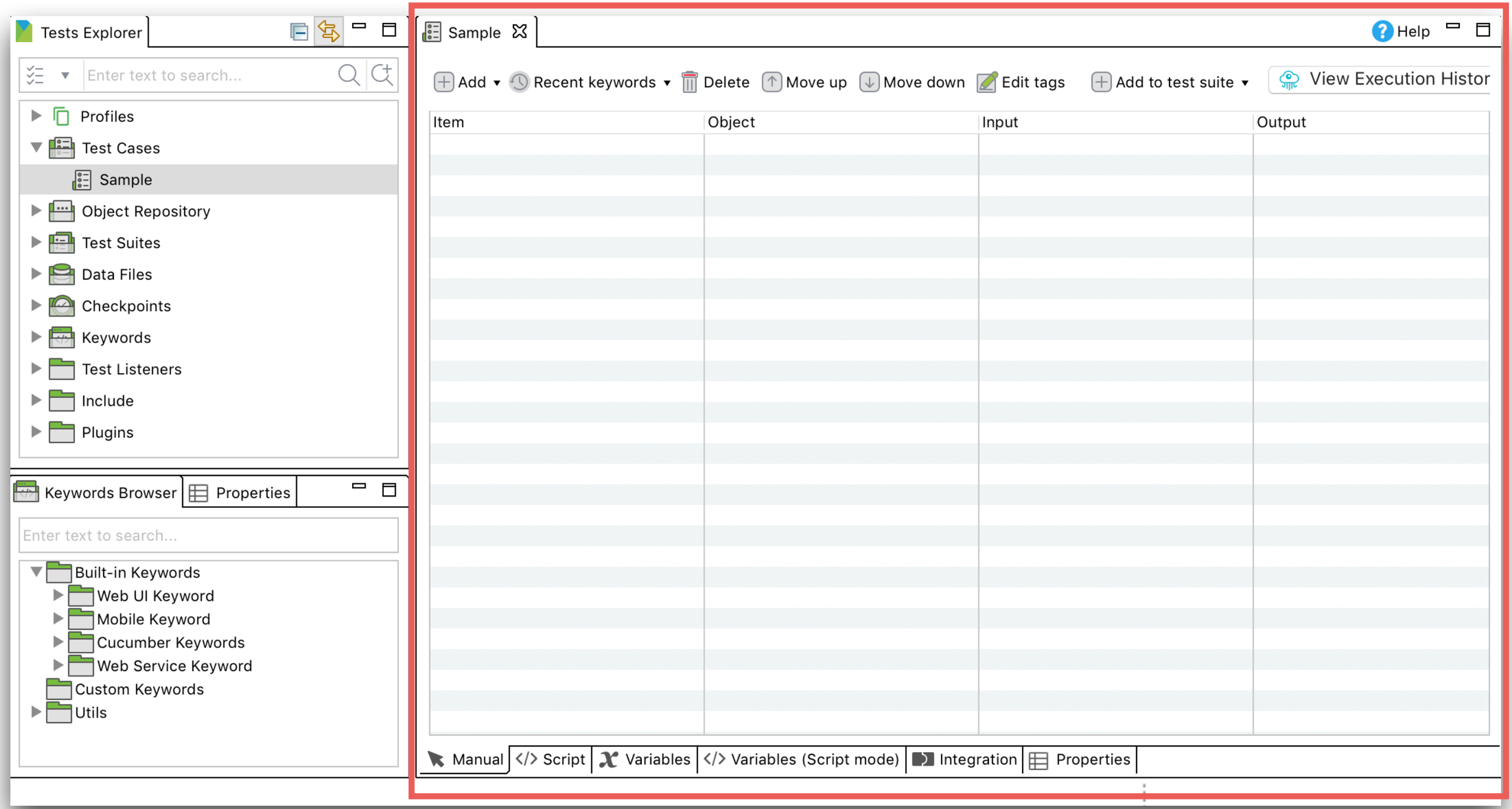


Keywords browser

Display all available keywords support by Katalon
Drag and drop the keywords to test case editor



Test case editor

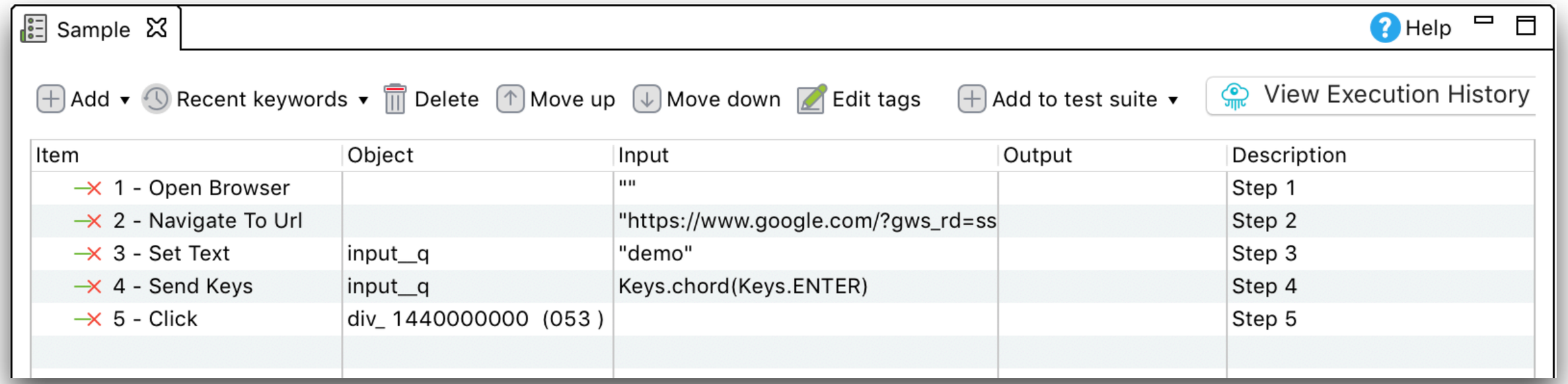


Test case editor

Manual tab
Script tab
Variables tab
Integration tab
Properties tab

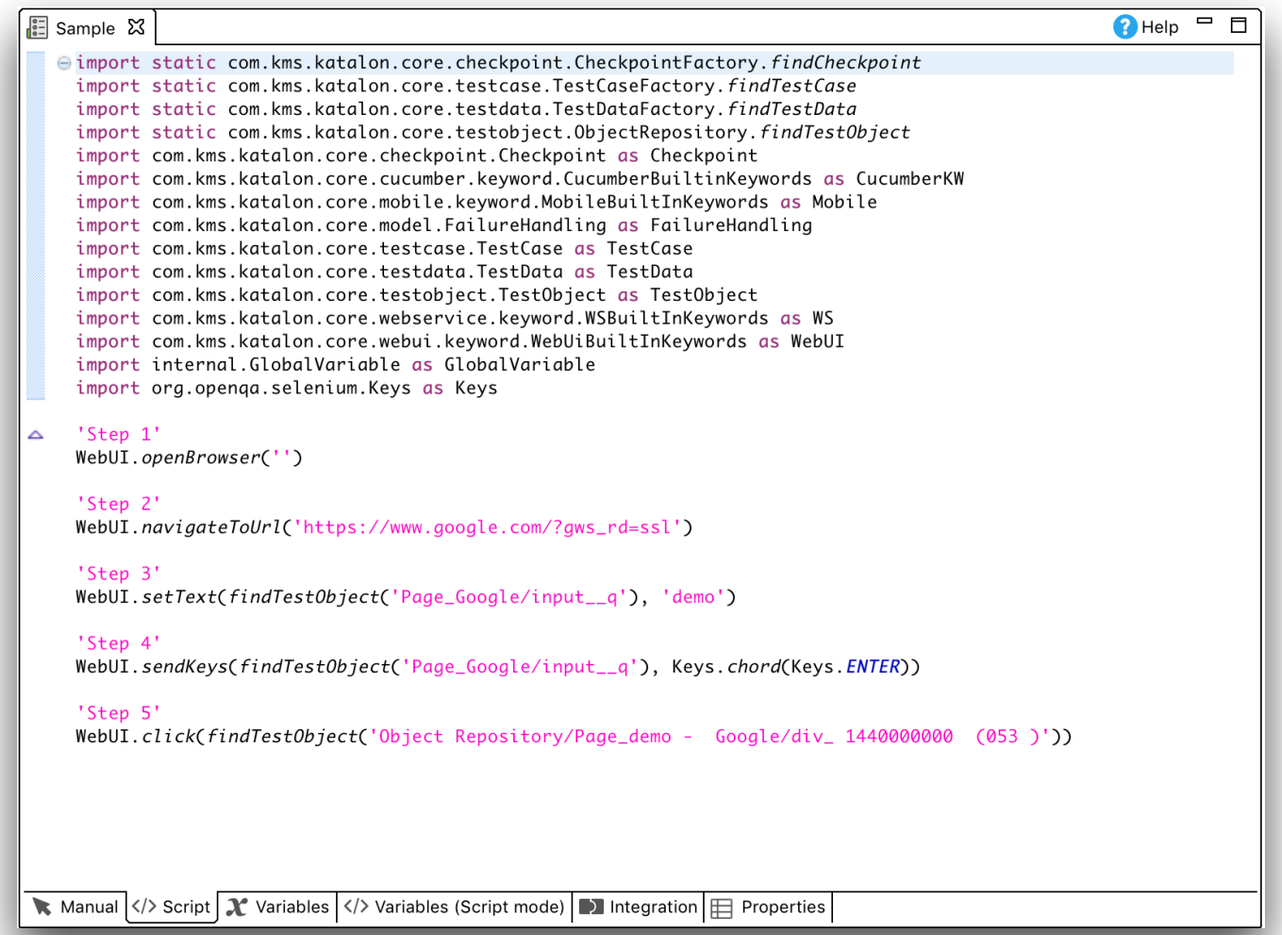
Manual tab

Display the manual view (basic keyword)
Allow user to create automation test effortlessly



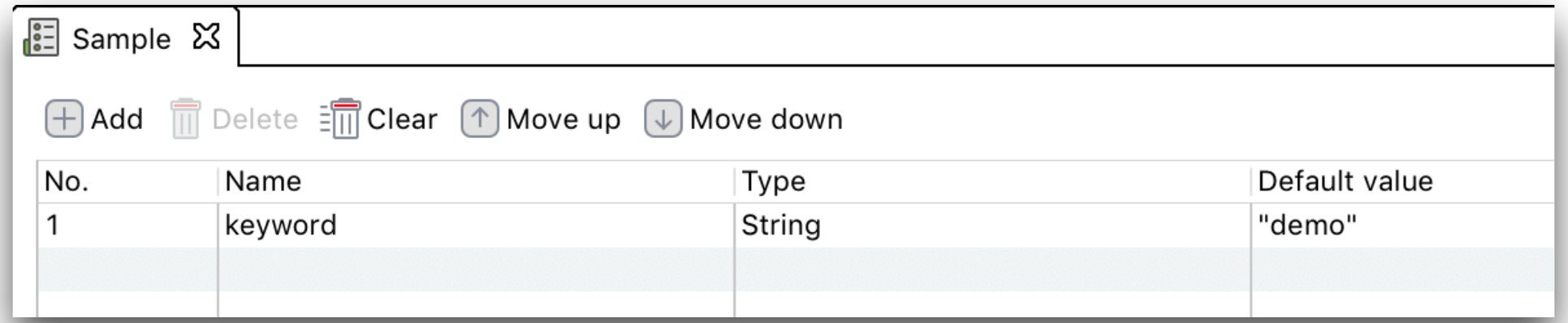
Script tab

Display the script view
Allow advance user to write code in test case
Write code with Java or Groovy



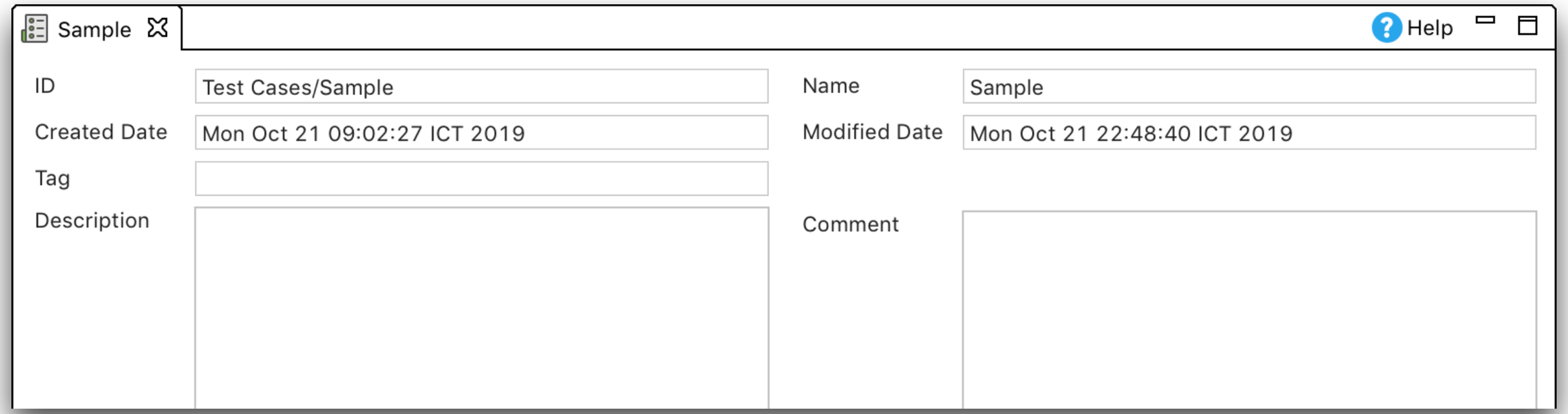
Variables tab

Show all variables defined for test case
Test case/ Groovy/ Global variables



Properties tab

General information about test cases

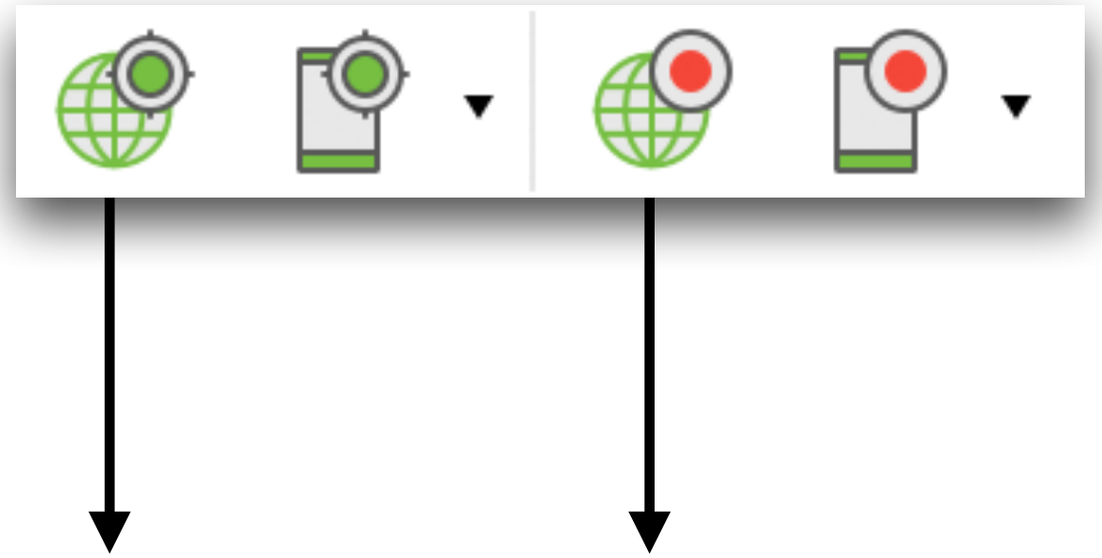


User Interface Testing

Working with web browser

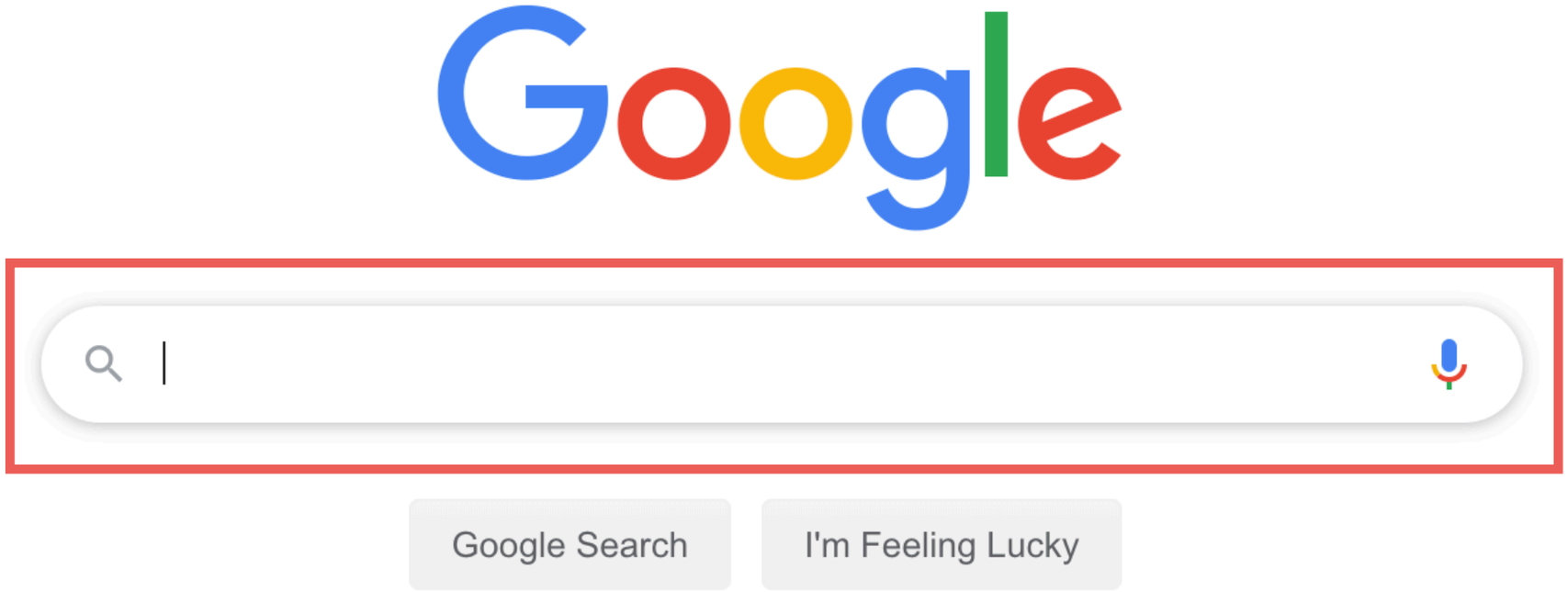
Spy web
Web recording
Run scripts
Report
Custom scripts

Toolbars



Spy web Record web

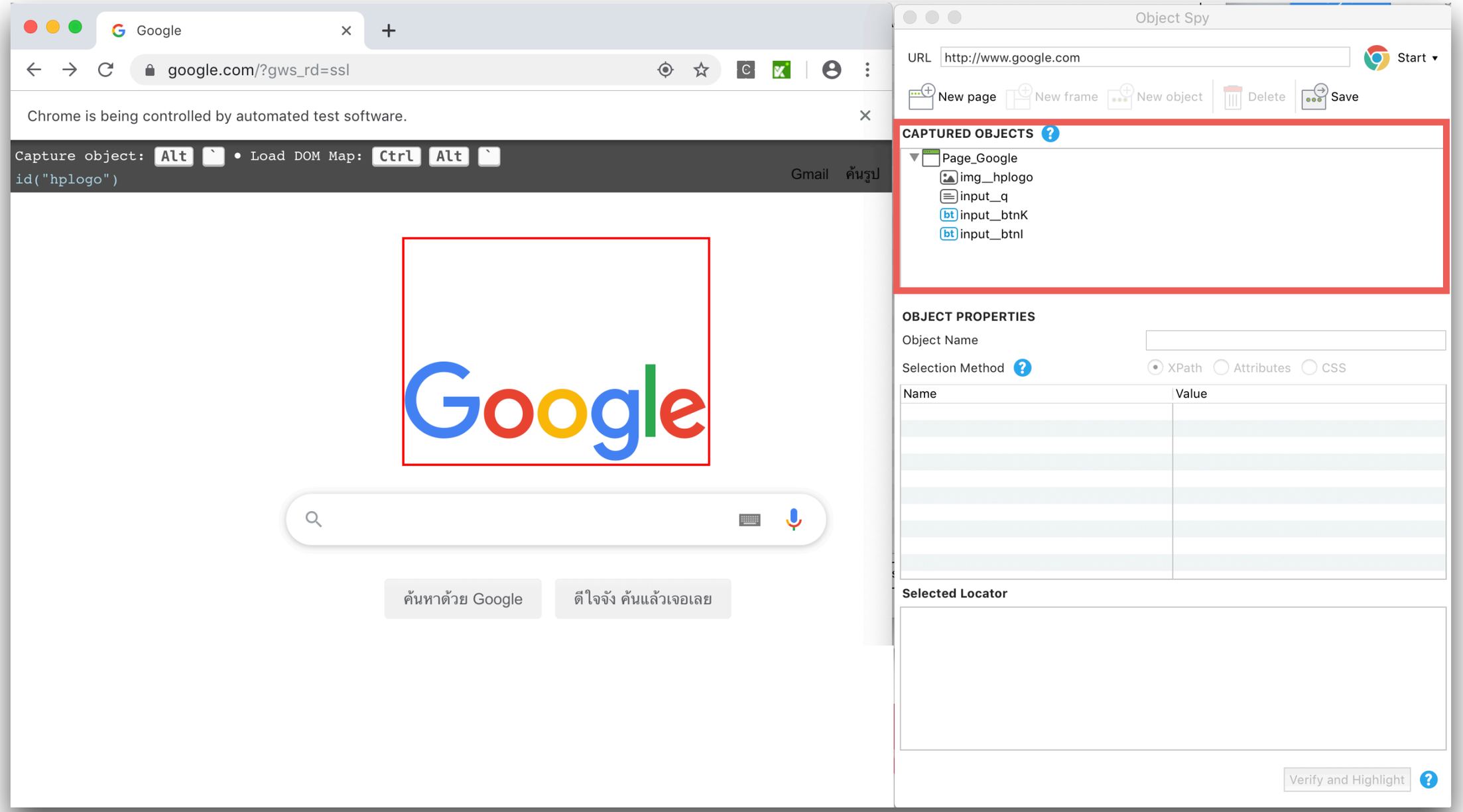
How to access element ?



Spy web

Capture objects from web browser
Save objects to object repository

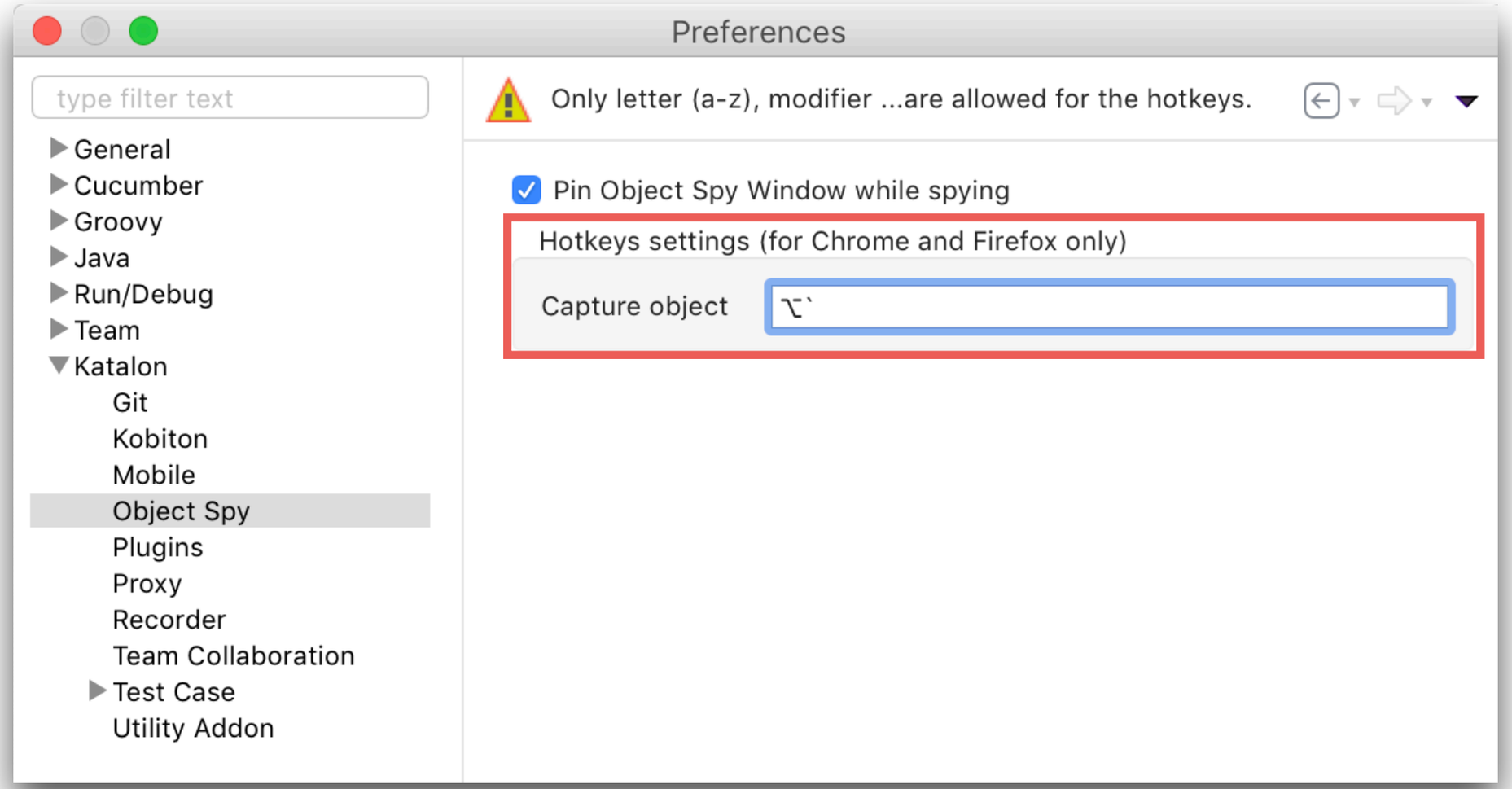
Spy web (1)



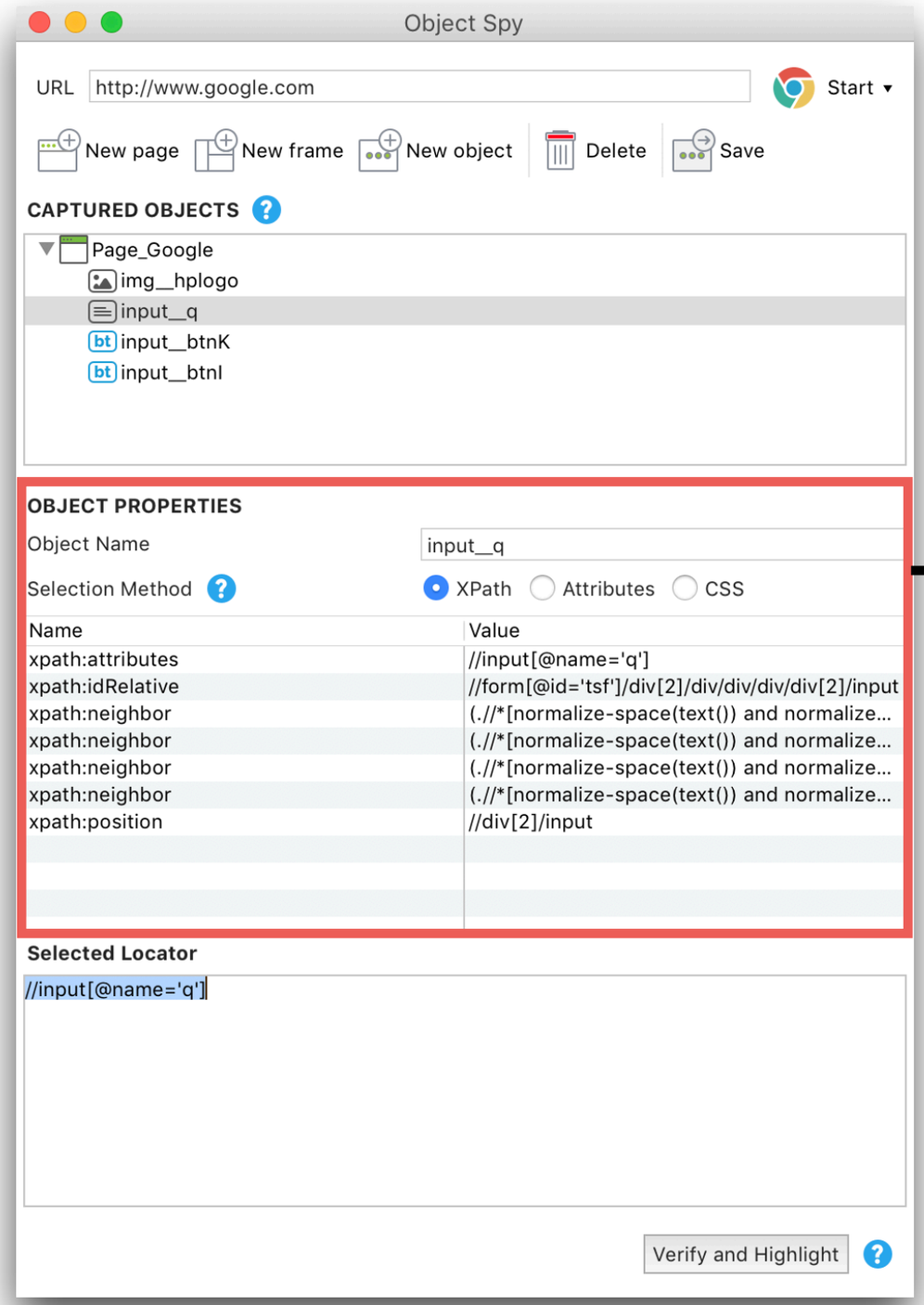
Capture object Alt + `

Change key to capture object

Preferences -> Katalon -> Object spy

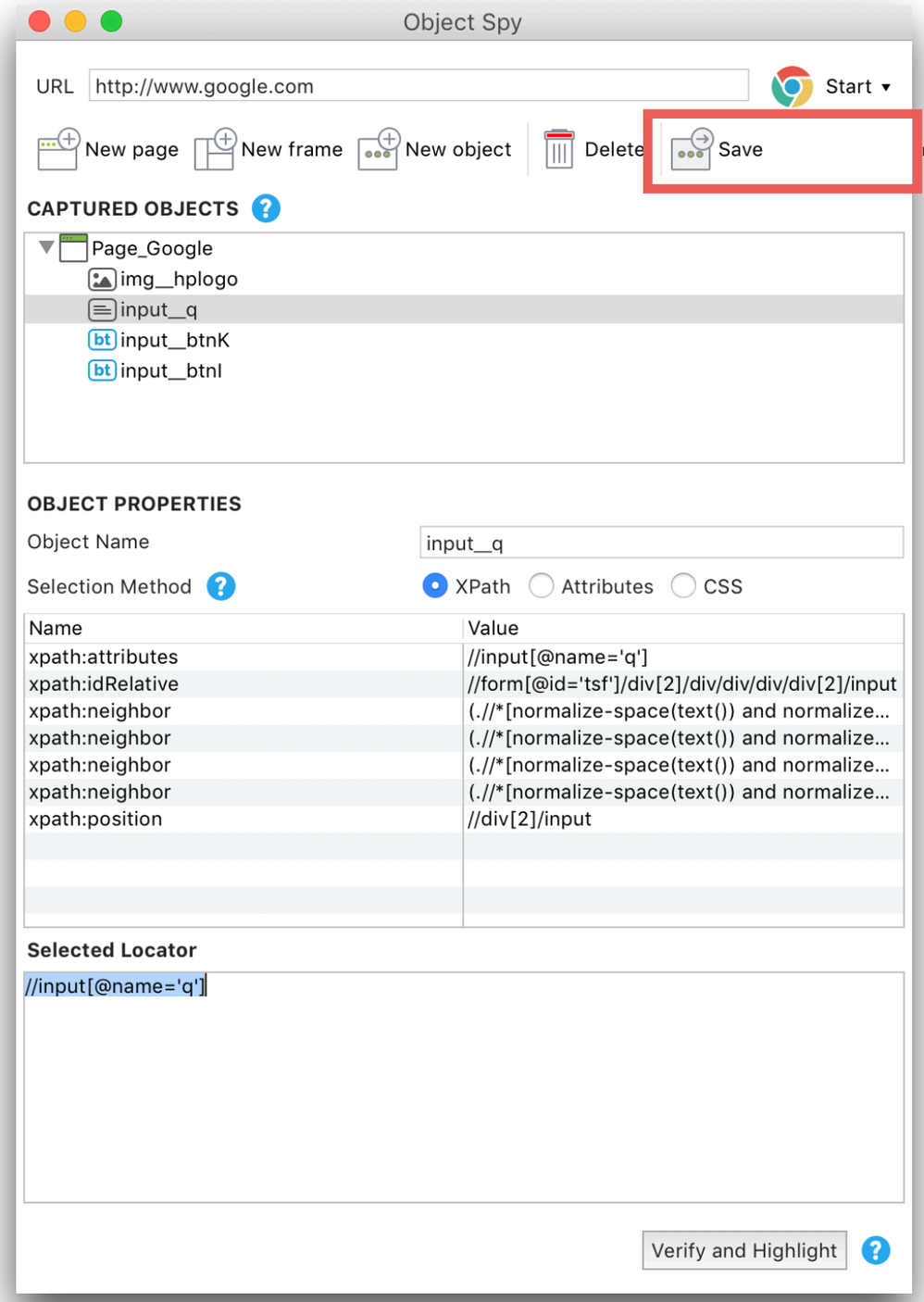


Spy web (2)



→ Object properties

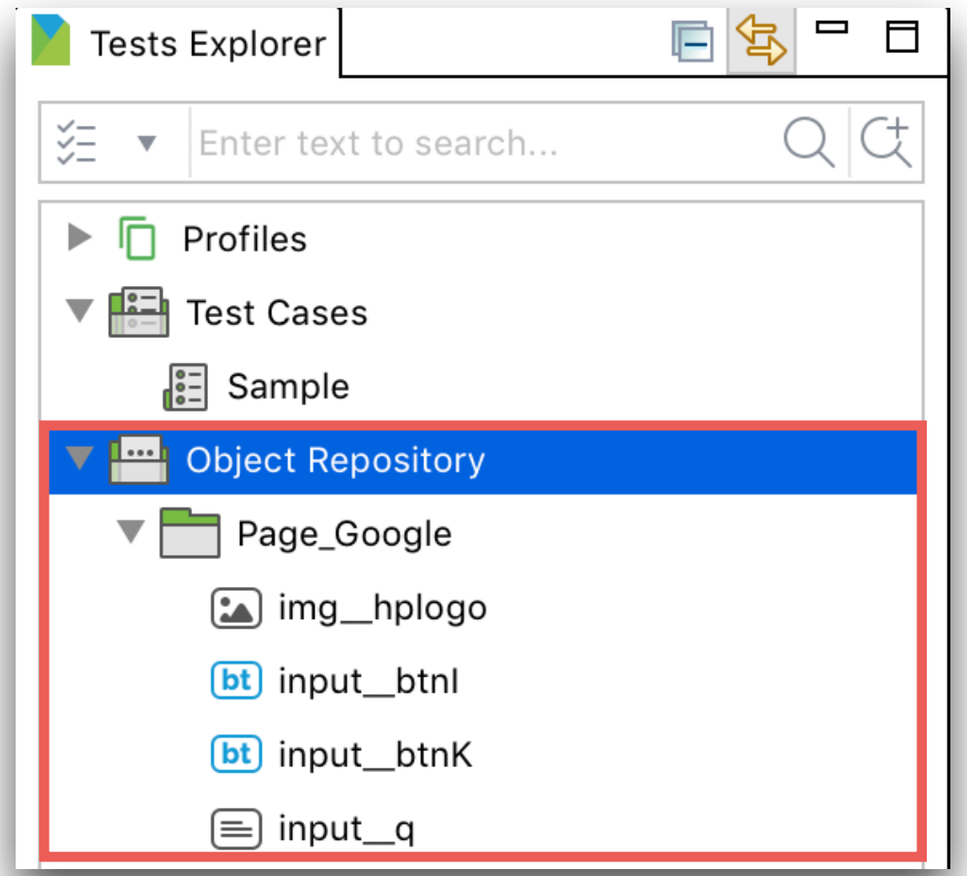
Spy web (3)



Save to object repository

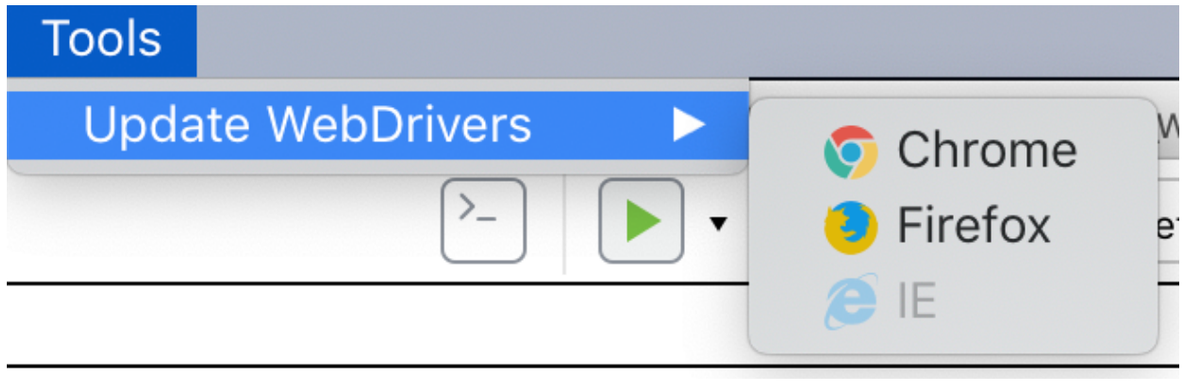
Spy web (4)

Save objects in folder/page object



Tips

Always Update Browser Driver !!



Element locators

Locator

A way to find a web element by selenium
Locator strategies (id, name, xpath, etc.)

Locators

Locator	Description
Id	Select element with the specified @id attribute
Name	Select first element with the specified @name attribute
Link text	Select link element which contains text matching the specified link text
Partial link text	Select link element which contains text matching the specified partial link text
Tag name	Locate Element using a Tag name
Class name	Locate Element using a class name
CSS	Select the element using CSS selectors
Xpath	Locate an element using a XPath expression

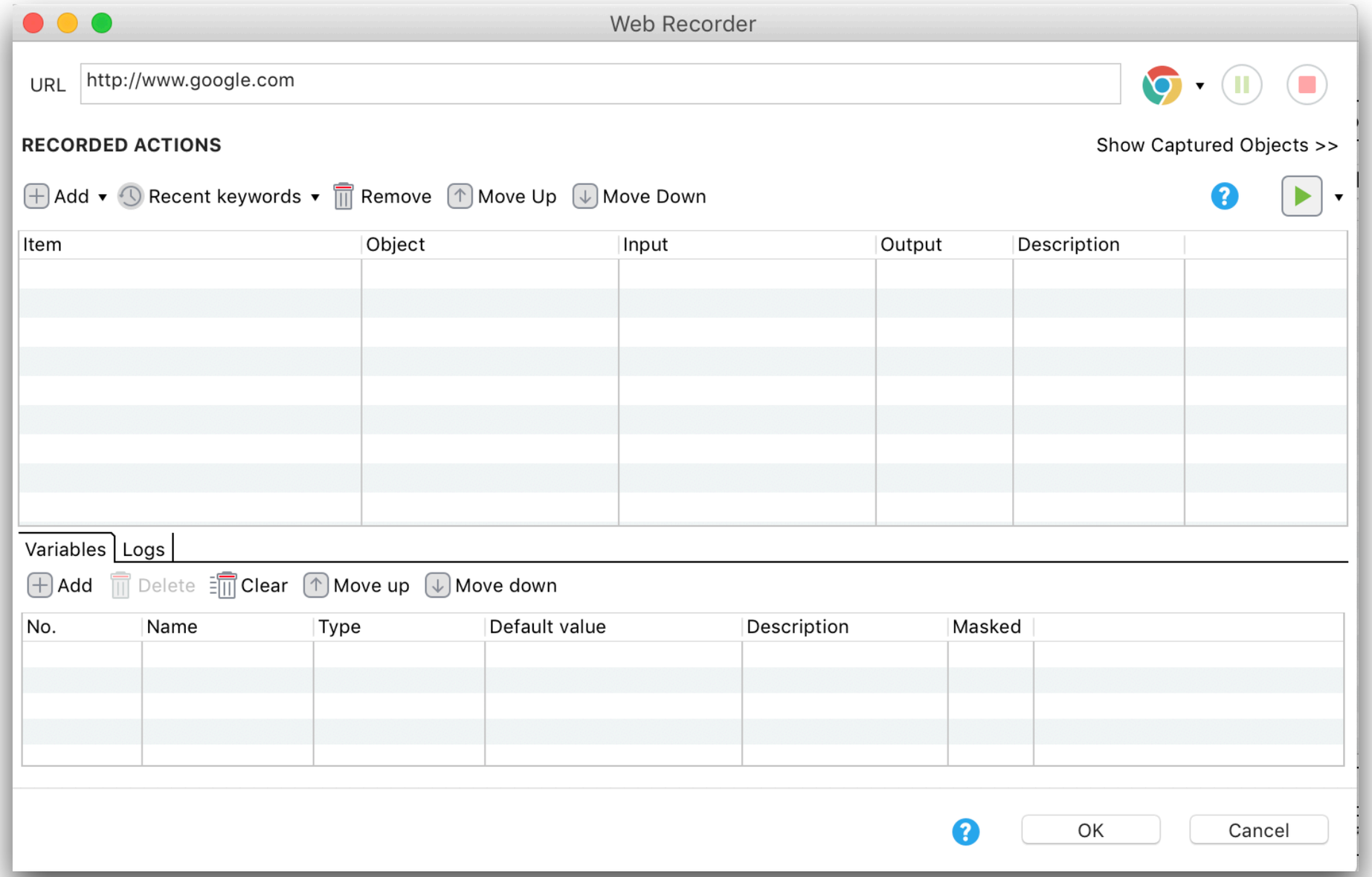
Challenges

Unable to find elements
Multiple elements matched
Hard to maintain scripts
Change in implementation

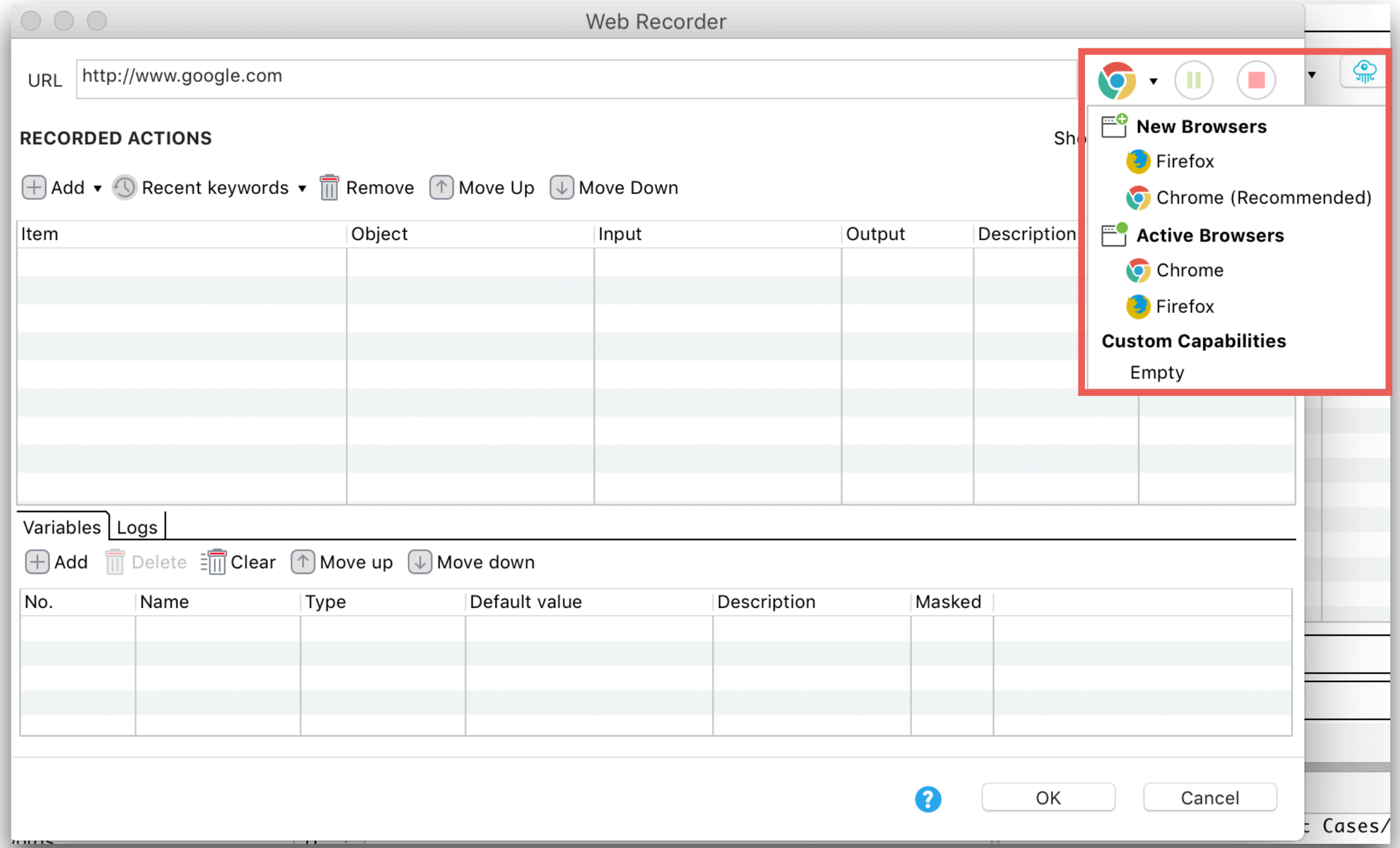
Web recording

Web recording

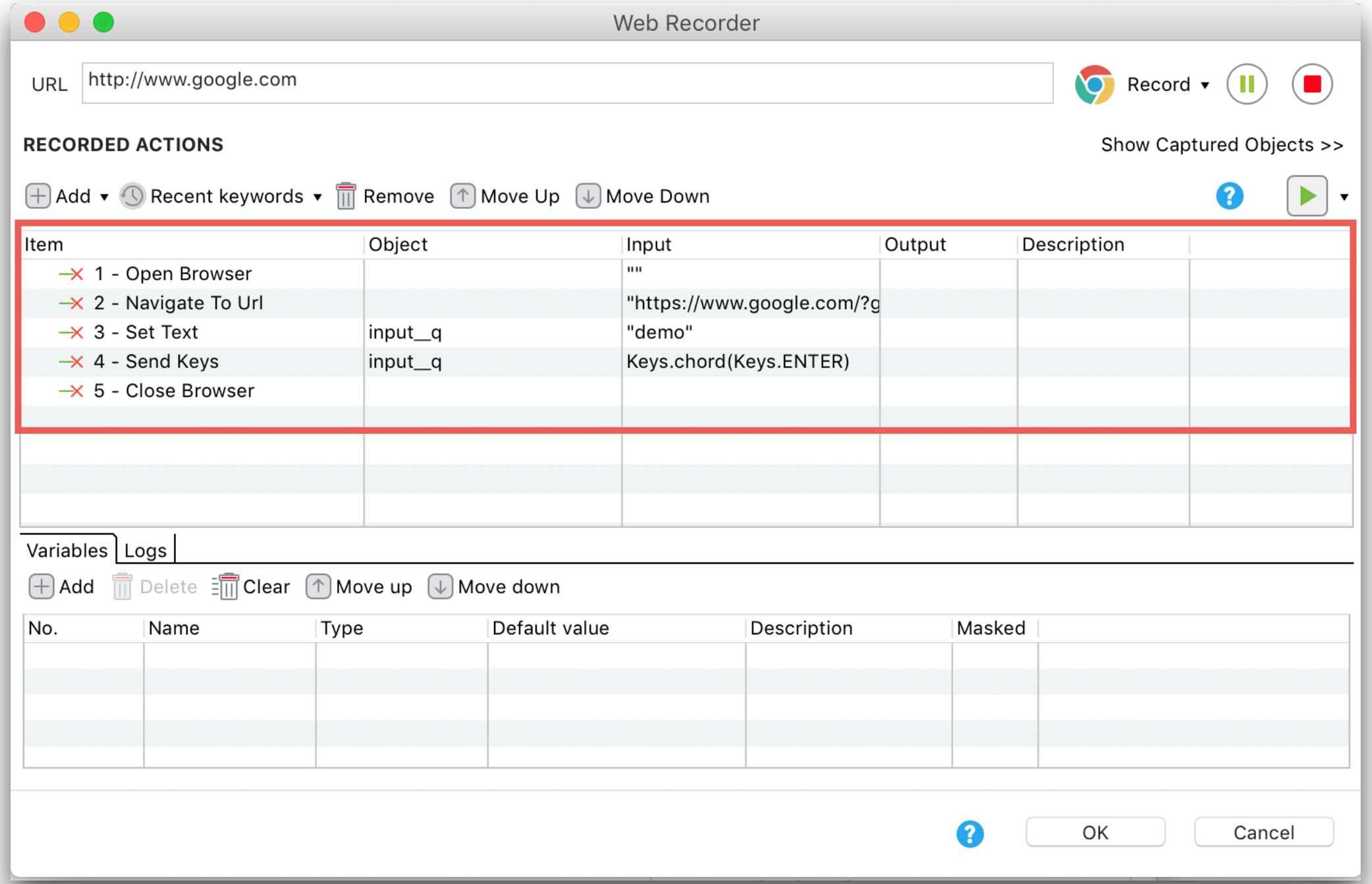
Record all actions from web browser



Try to record (choose browser)

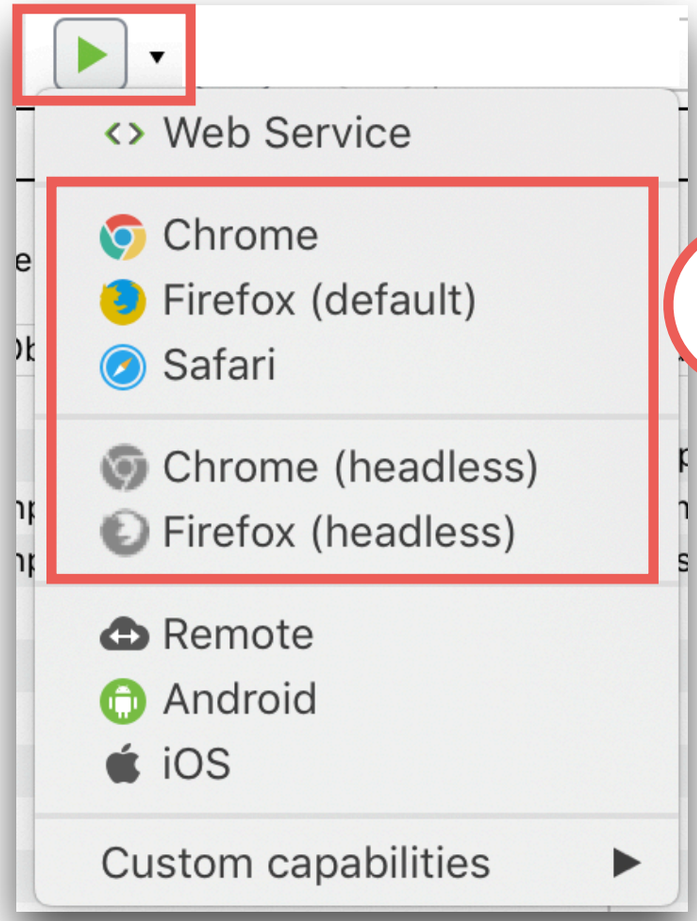


Let’s record



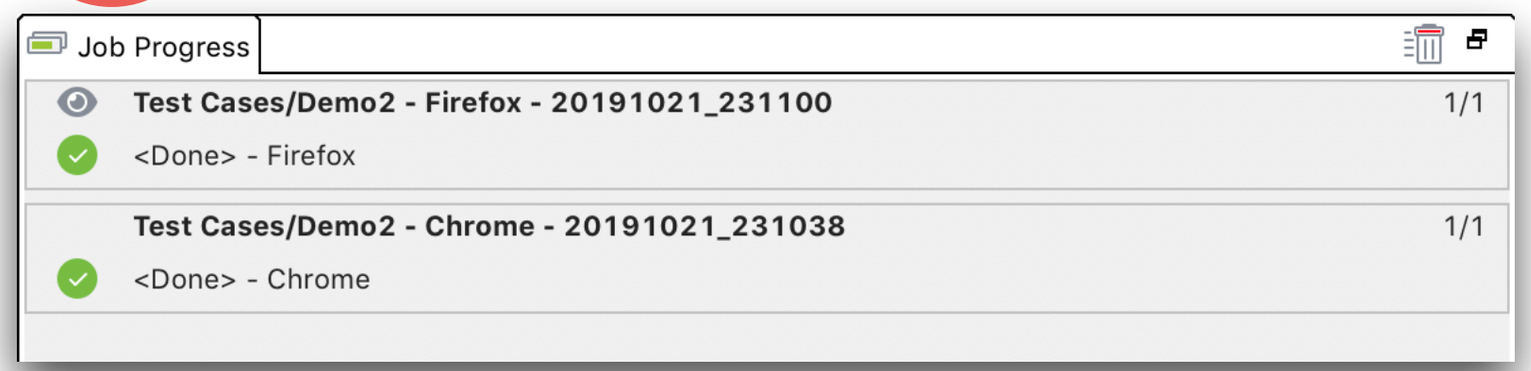
Run test cases

1 Click run test case

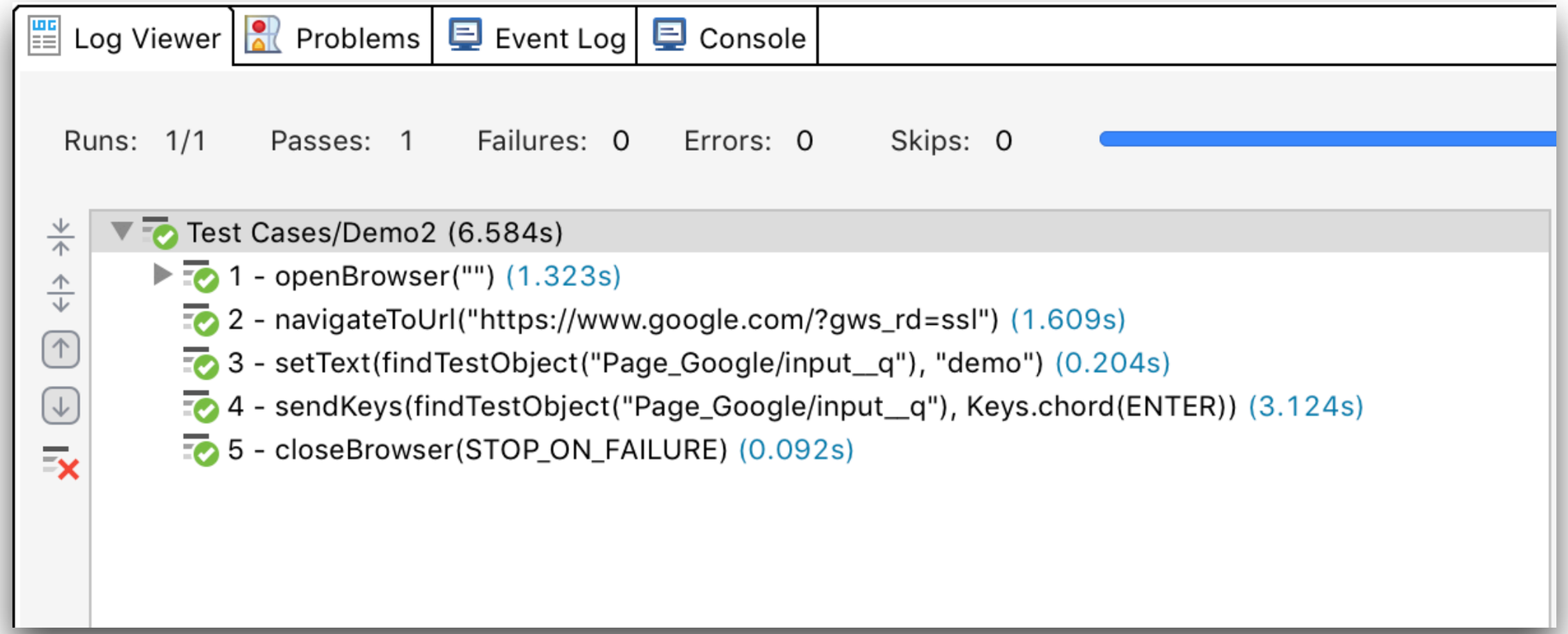


2 Choose target/browser to run tests

3 Test results



Test logs

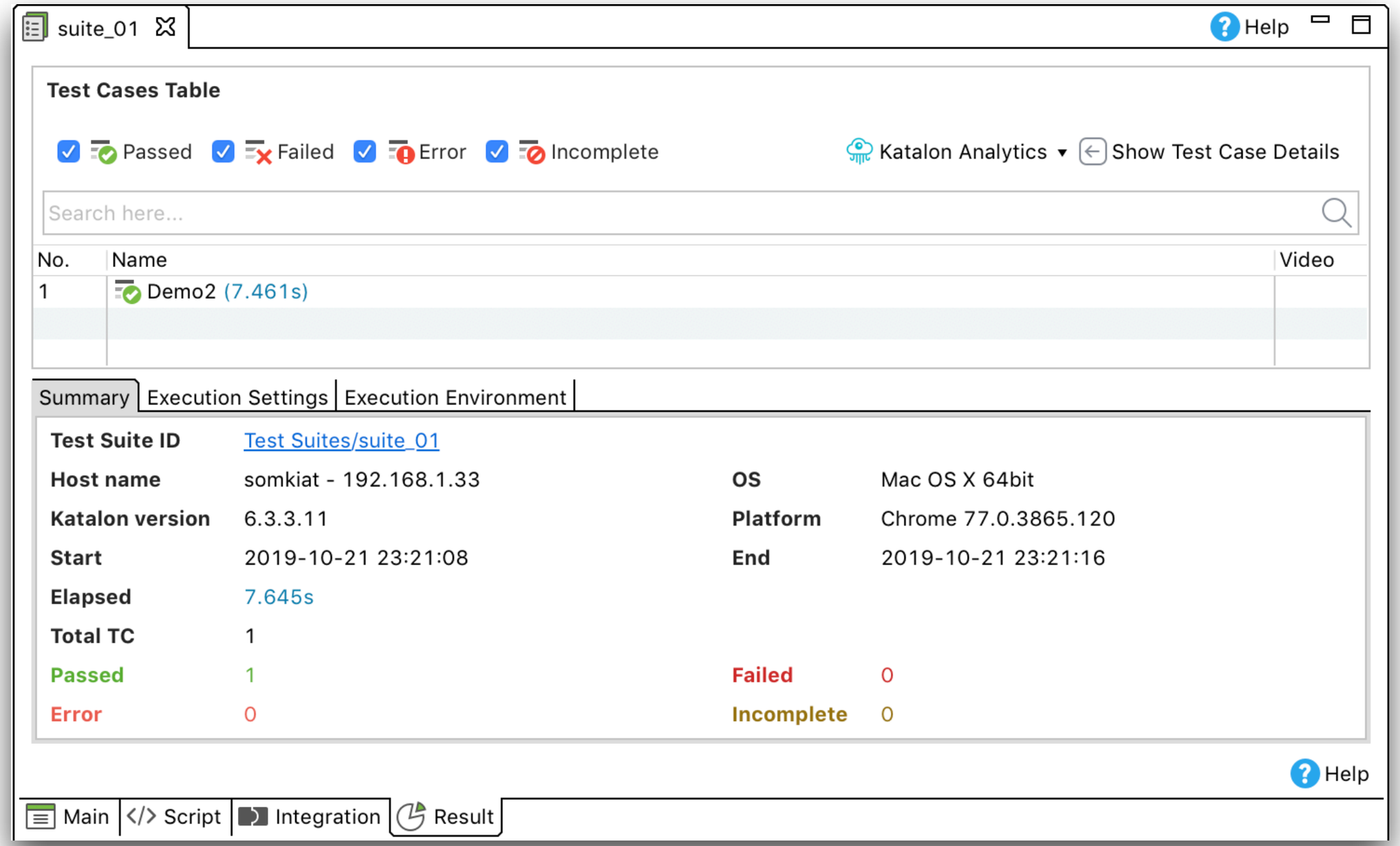


https://docs.katalon.com/katalon-studio/docs/test-suite-report.html

More reports

Create new test suite -> add test cases

Run suite and see report



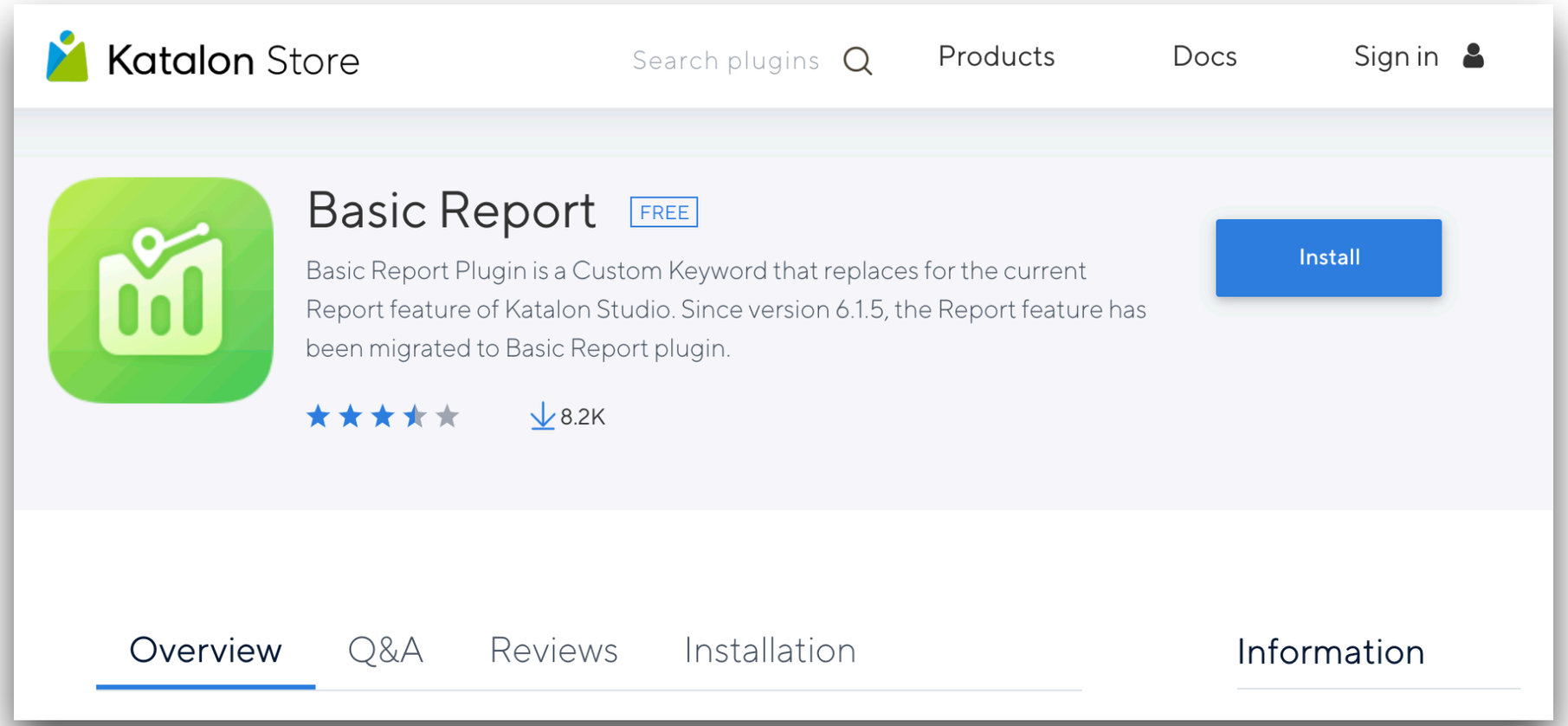
Export report to other formats

Since Katalon 6.1.5, report feature migrated to Basic report plugin

Support formats (HTML, CSV, Junit, PDF)

Export report to other formats

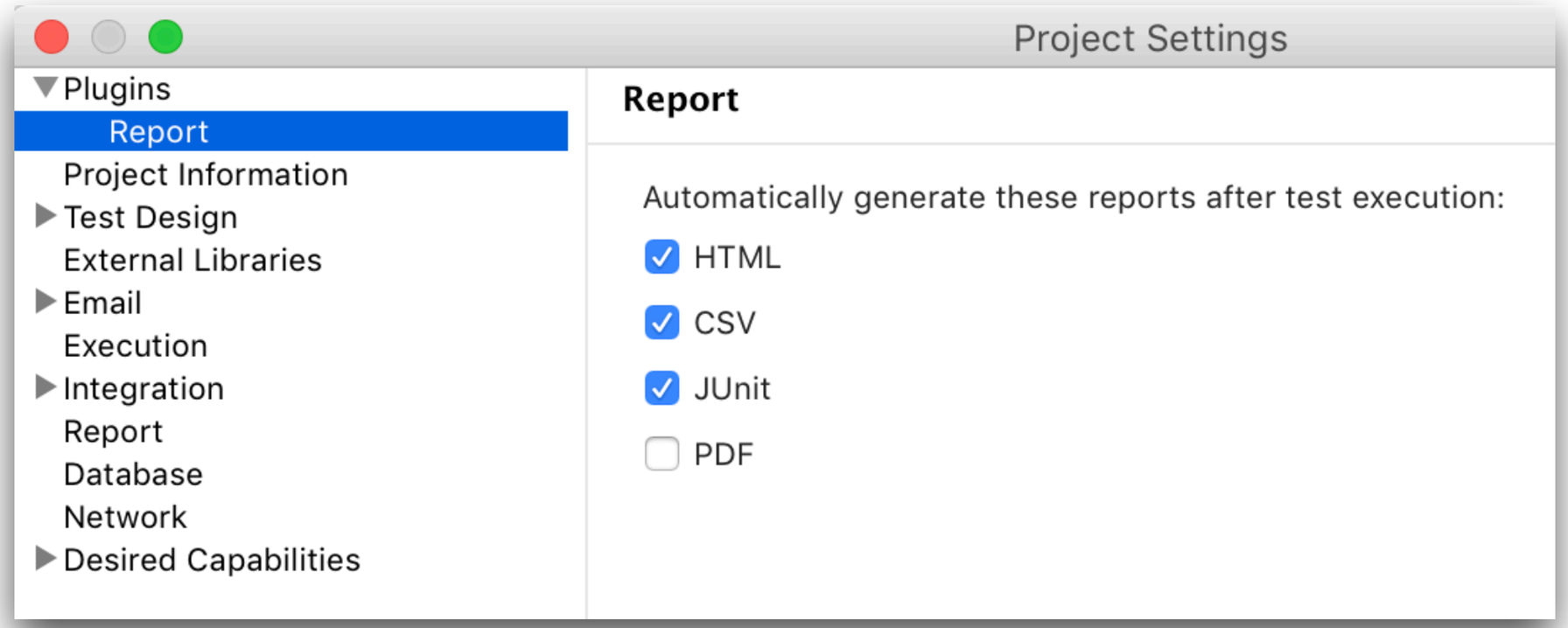
Try to install Basic report plug-in and **reload !!**



https://store.katalon.com/product/59/Basic-Report

Setting report plug-in

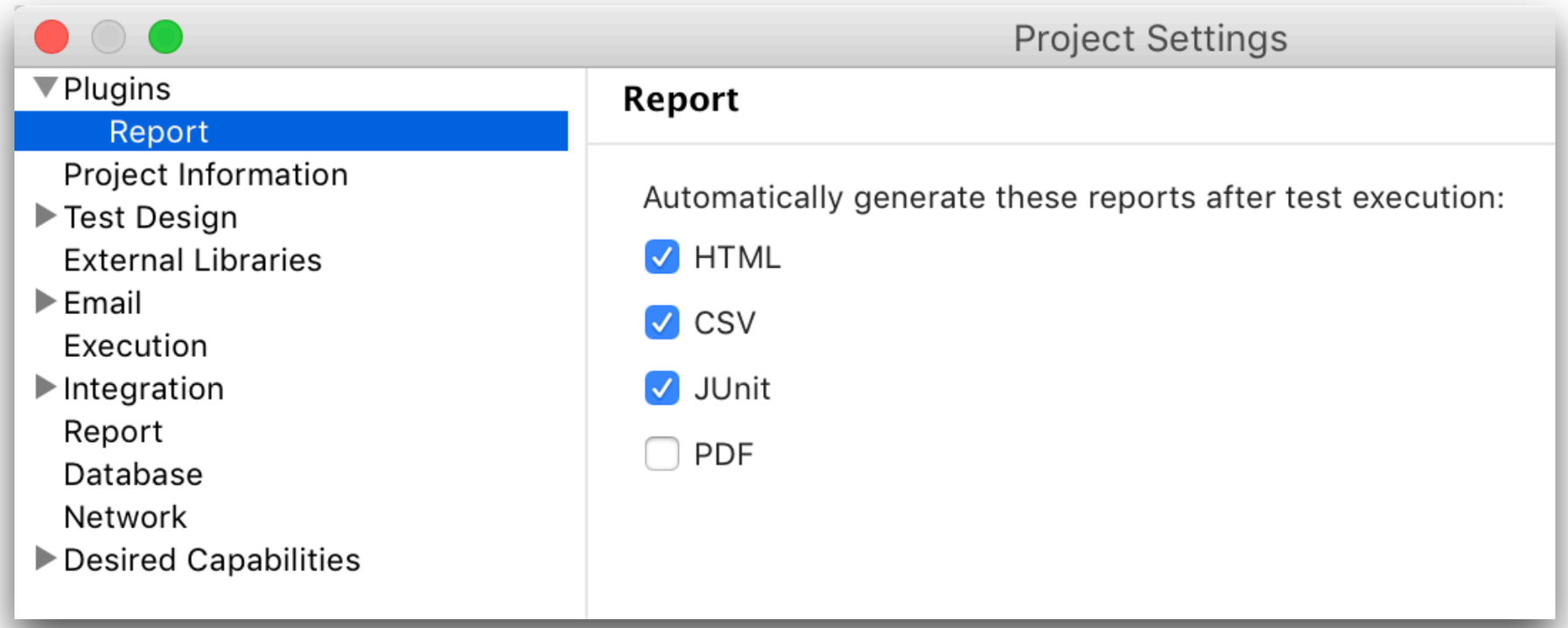
Goto Project -> Settings



https://docs.katalon.com/katalon-studio/docs/basic-report.html

Setting report plug-in

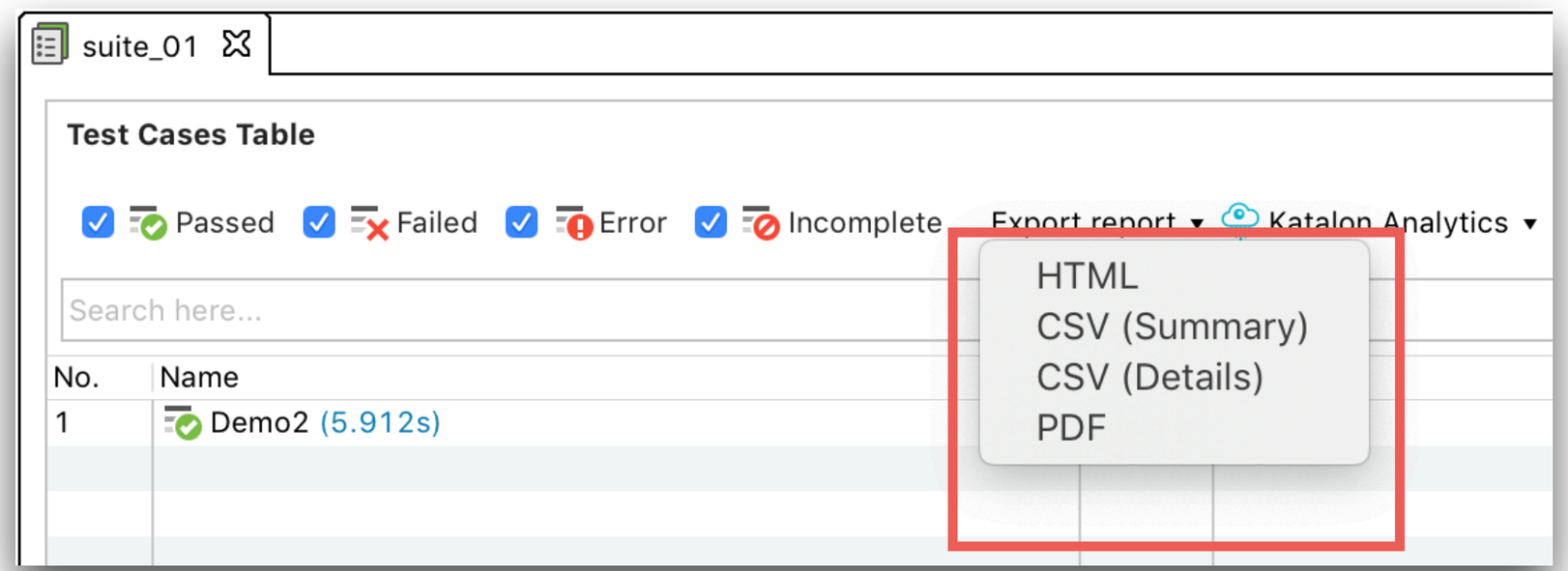
Goto Project -> Settings



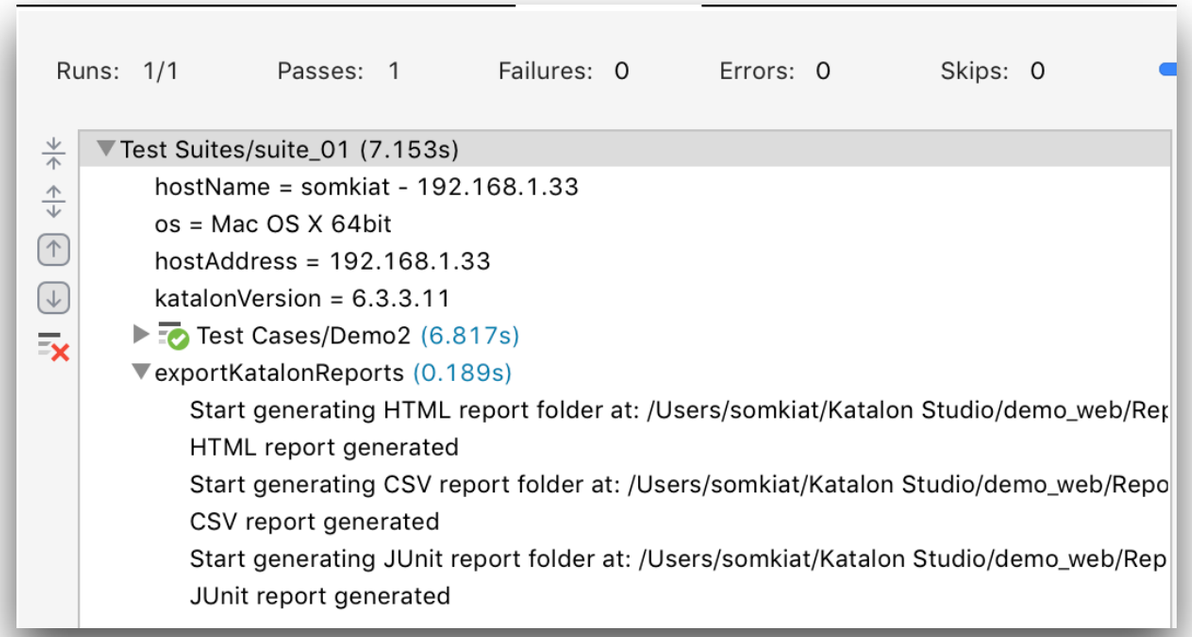
https://docs.katalon.com/katalon-studio/docs/basic-report.html

Run tests and see reports

Test suite



Log viewer



HTML report

Execution Environment

Host name:	somkiat - 192.168.1.33
OS:	Mac OS X 64bit
Katalon version:	6.3.3.11
Browser:	Firefox 69.0

Test Execution Log

TEST SUITE: **suite_01** Expand All

Full Name: suite_01
Start / End / Elapsed: 2019-10-23 02:19:33.097 / 2019-10-23 02:19:39.130 / 00:00:06.033
Status: 1 test total, 1 passed, **0 failed**, **0 error**, 0 incomplete

TEST CASE: **Test Cases/Demo2** Expand All

Full Name: suite_01/Test Cases/Demo2
Start / End / Elapsed: 2019-10-23 02:19:33.218 / 2019-10-23 02:19:39.130 / 00:00:05.912
Status: PASSED

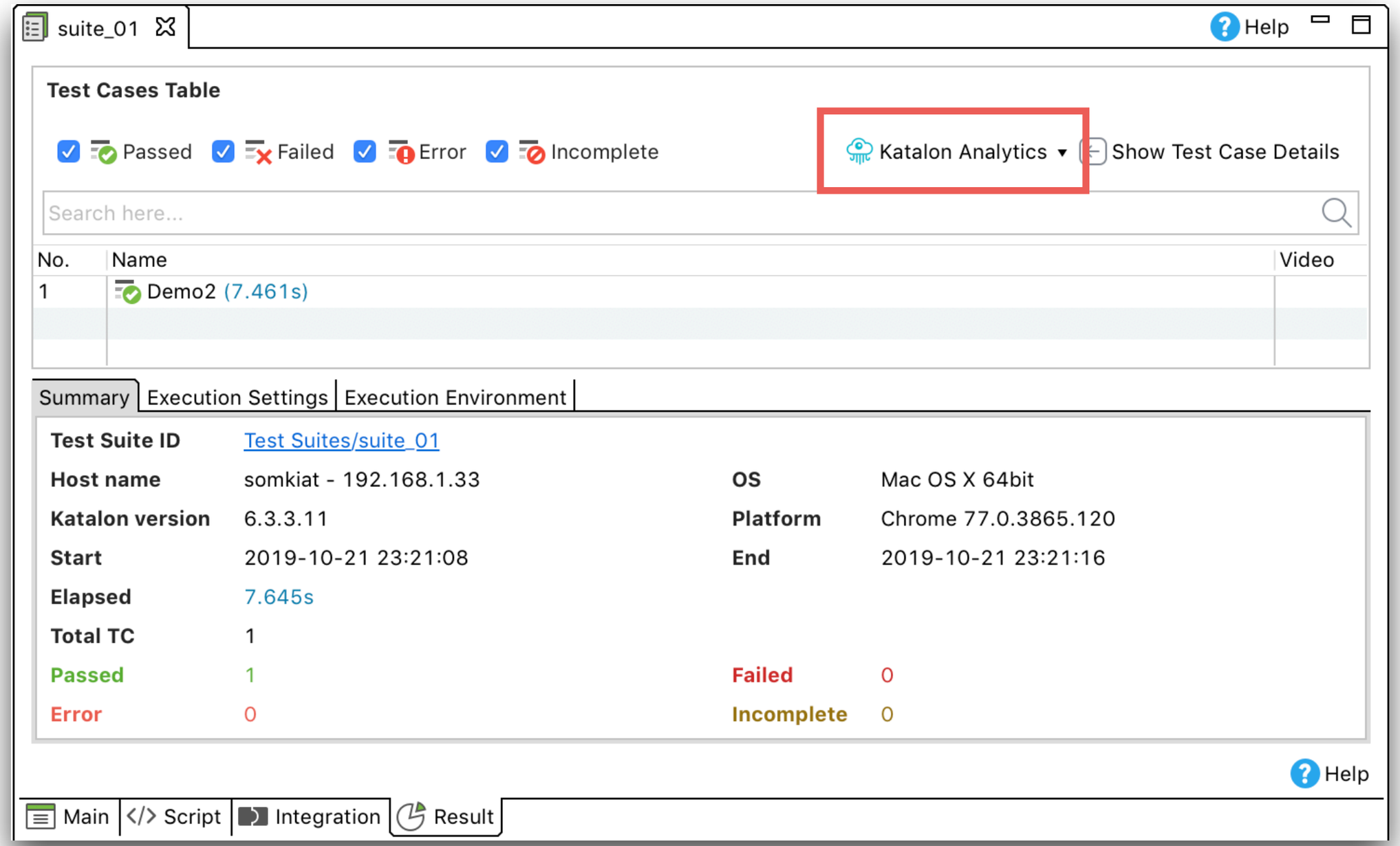
TEST STEP: *openBrowser("https://www.google.com/?gws_rd=ssl")*

TEST STEP: *setText(findTestObject("Page_Google/input__q"), "demo")*

TEST STEP: *sendKeys(findTestObject("Page_Google/input__q"), Keys.chord(ENTER))*

TEST STEP: *closeBrowser(STOP_ON_FAILURE)*

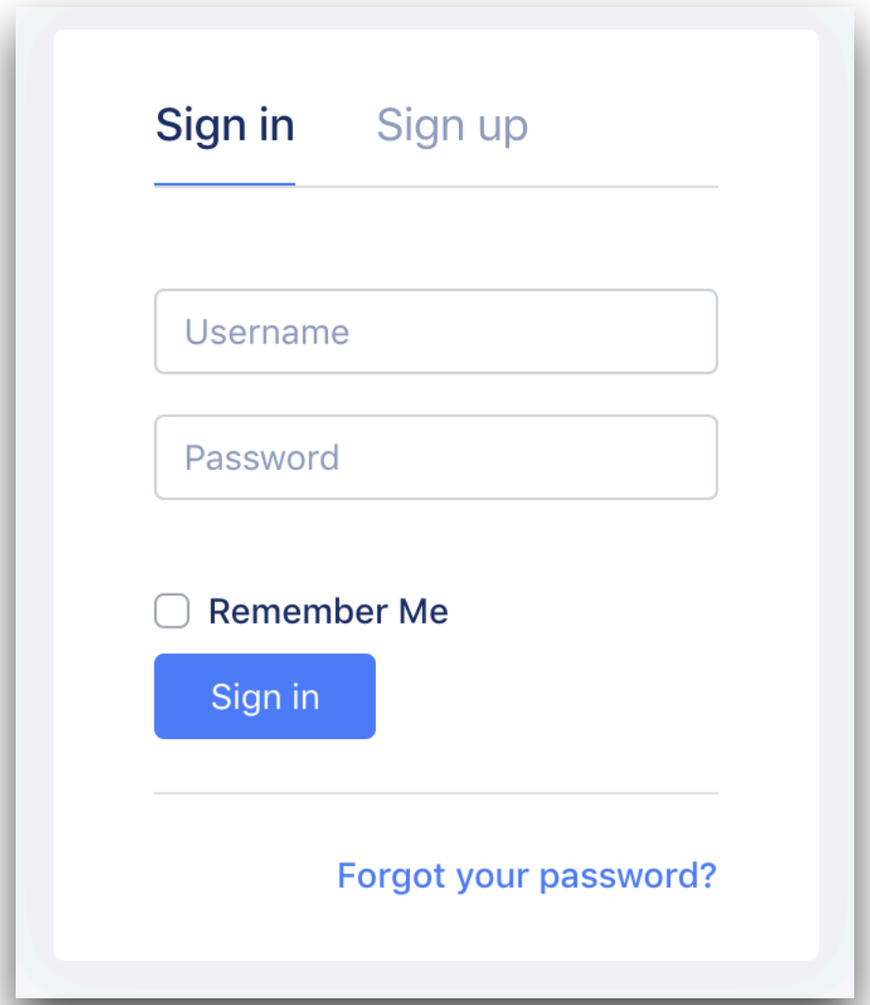
Integrate with Katalon analytics



https://docs.katalon.com/katalon-analytics/docs/view-reports.html

Katalon analytic service

Cloud service from Katalon

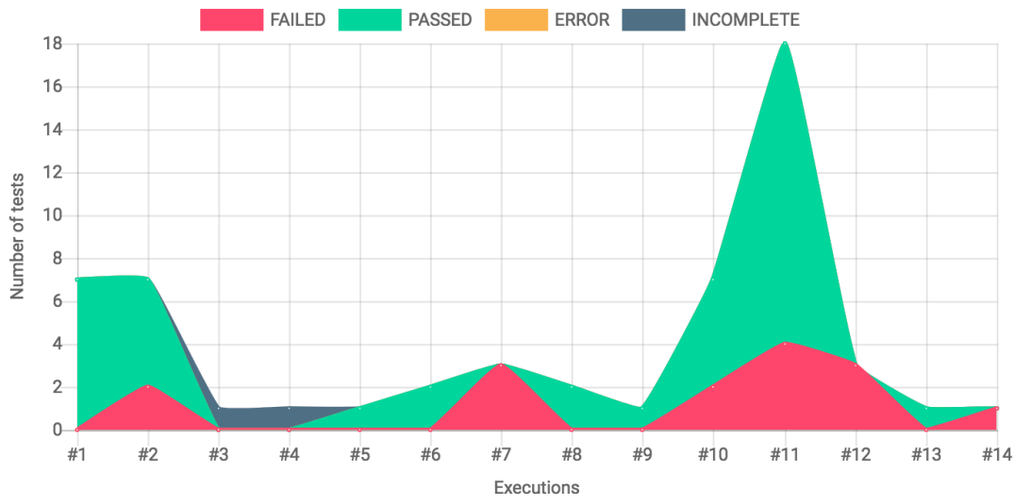


https://analytics.katalon.com/login

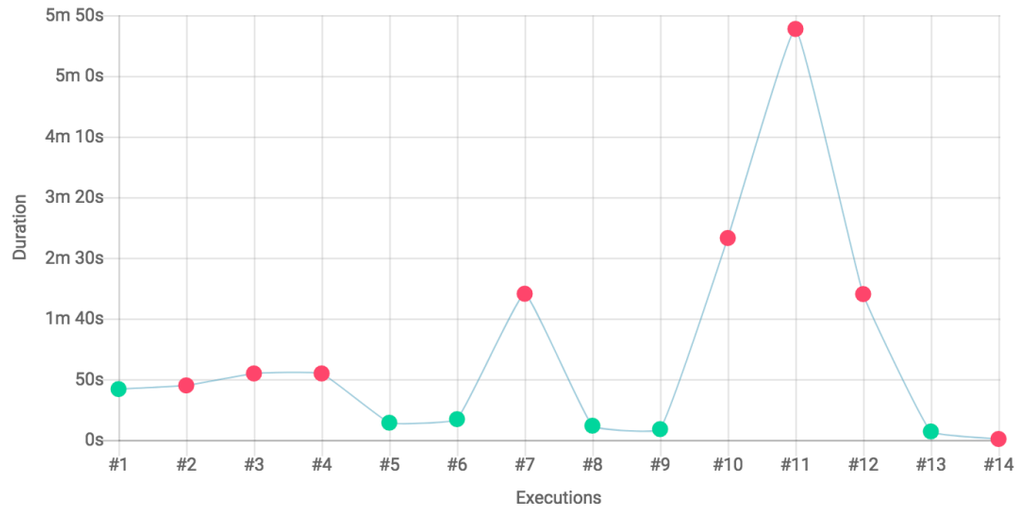
Katalon analytic service

Project: Katalon Demo Project

Status



Performance



Executions

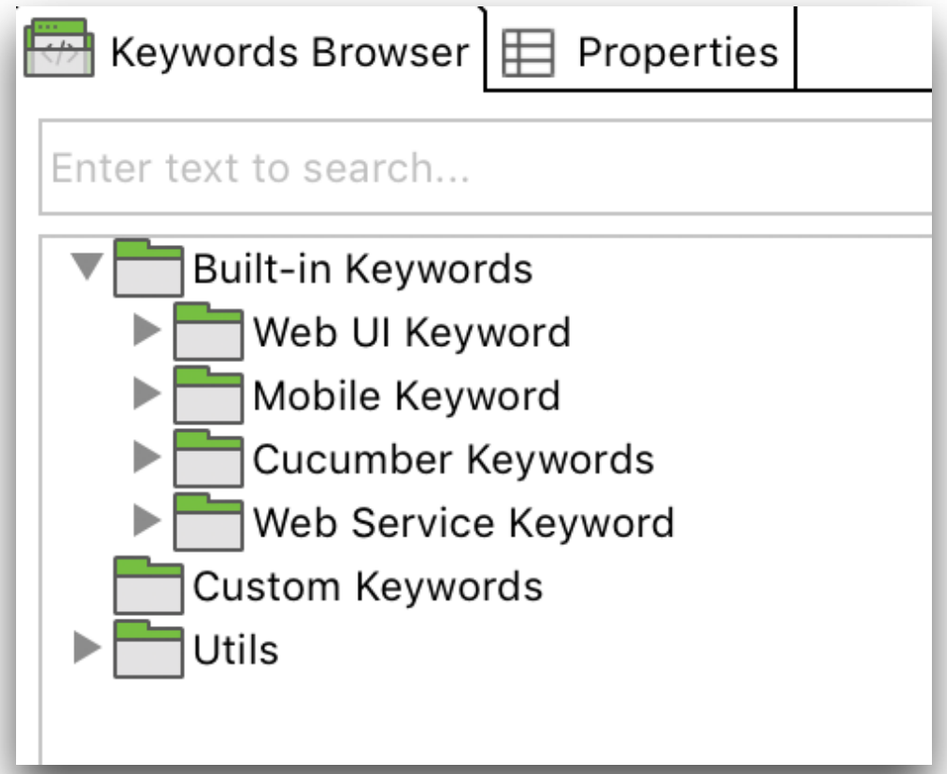
Filter

Sort

ID	Status	Test Suite	Started	Duration	Total	Passed	Failed	Error	Incomplete	Action
#14	FAILED	New Test Suite	08/01/2018 13:36:42	750ms	1	0	1	0	0	
#13	PASSED	Katalon Recorder Samples	03/19/2018 15:03:10	7s	1	1	0	0	0	
#12	FAILED	TS_Image Keywords	03/02/2018 16:31:54	2m 0s	3	0	3	0	0	

Build-in keywords

Build-in keywords



Web UI keywords

Working with browser
Working with elements
Working with text
Upload files
Frame/Form/Checkbox/Combo box/ Alert

Working with browser

Keyword	Description
Open browser	Open a browser and navigate to the specified URL
Navigate to Url	Navigate to the specified web page
Close browser	This action will close all windows of the browser
Maximize Window	Resize the current window to take up the entire screen
Refresh	Simulate users clicking "refresh" button on their browser

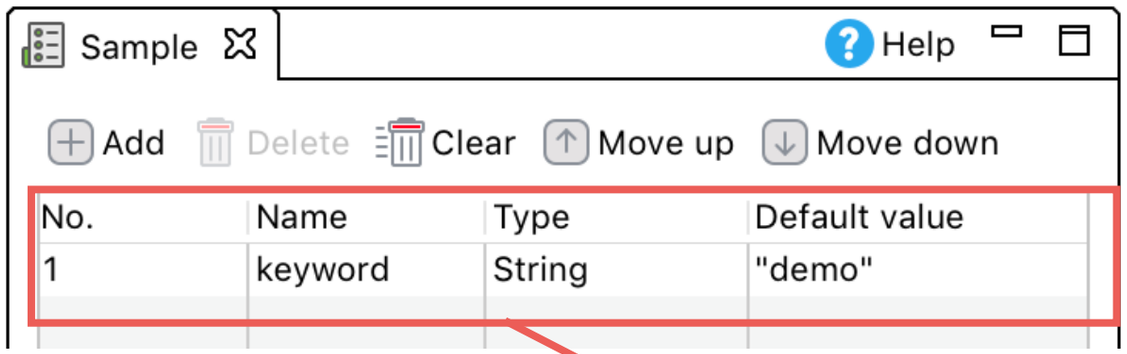
https://docs.katalon.com/katalon-studio/docs/webui-open-browser.html

Working with textbox

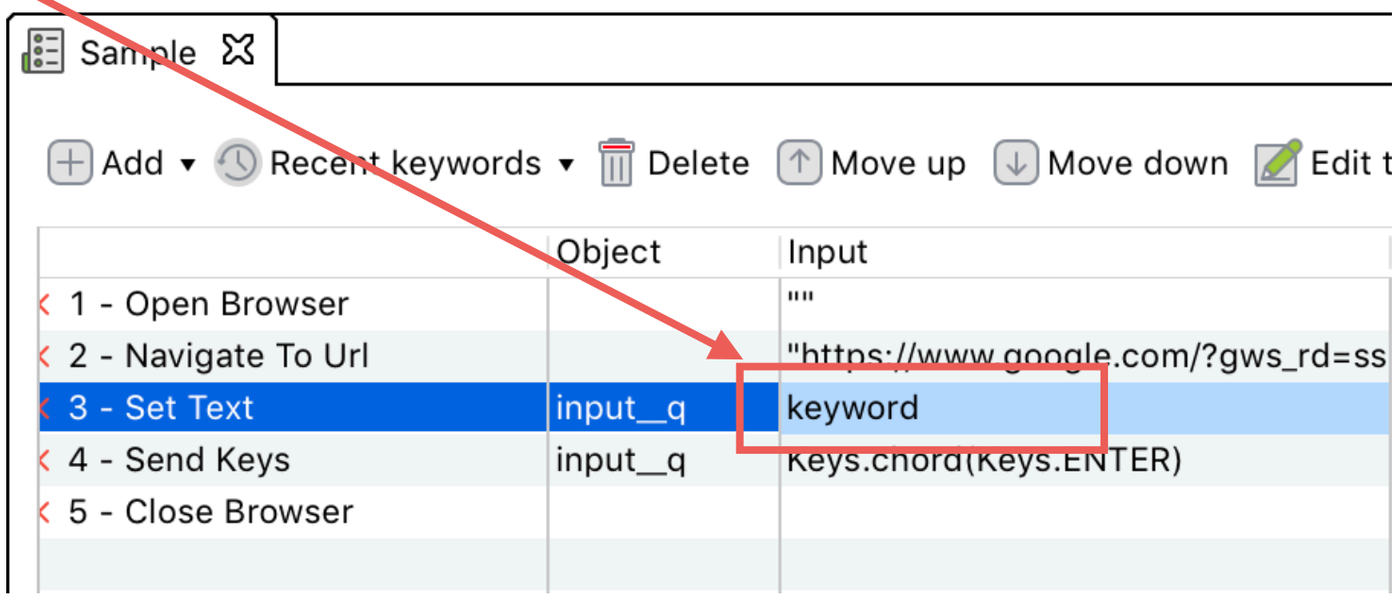
Keyword	Description
Set text	Set the value of an input field, as though you type it in
Send keys	Simulates keystroke events on the specified element
Clear text	Clear the value of an input field
Set encrypted text	Set encrypted text into an input field

Working with variables

Variables tab



Using in manual tab



Verify elements in browser

Verify Elements

Verify alert
Verify text
Verify element
Verify option

Wait For

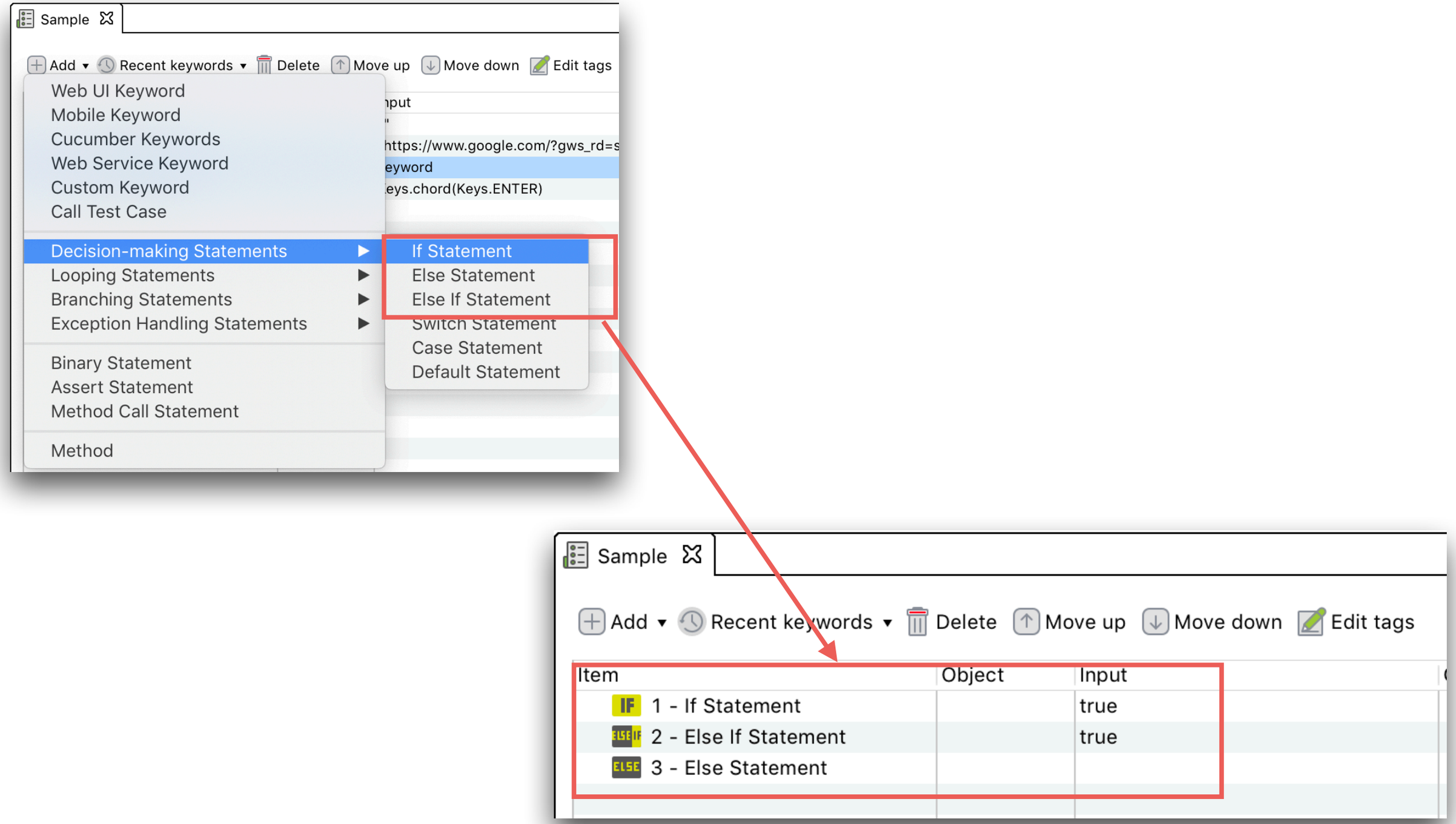
Wait for alert
Wait for element
Wait for image
Wait for page loaded

Conditional execution of steps

Conditional of steps

If only
If and Else
If Else-If Else
Switch-case-default

Conditional of steps



Data-driven testing

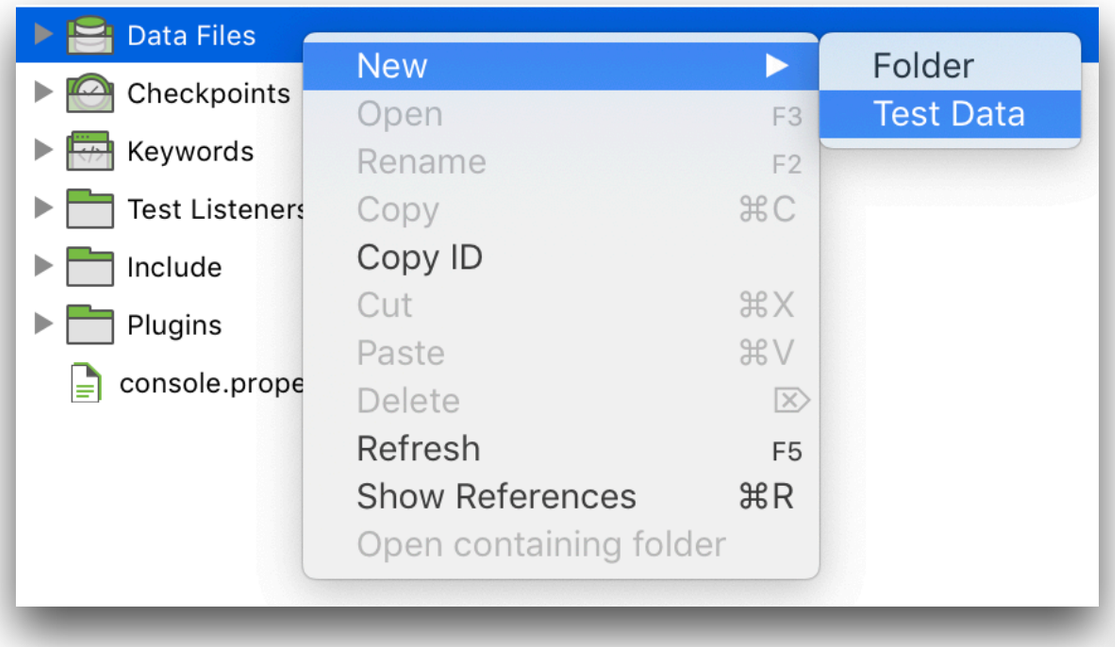
Data-driven testing

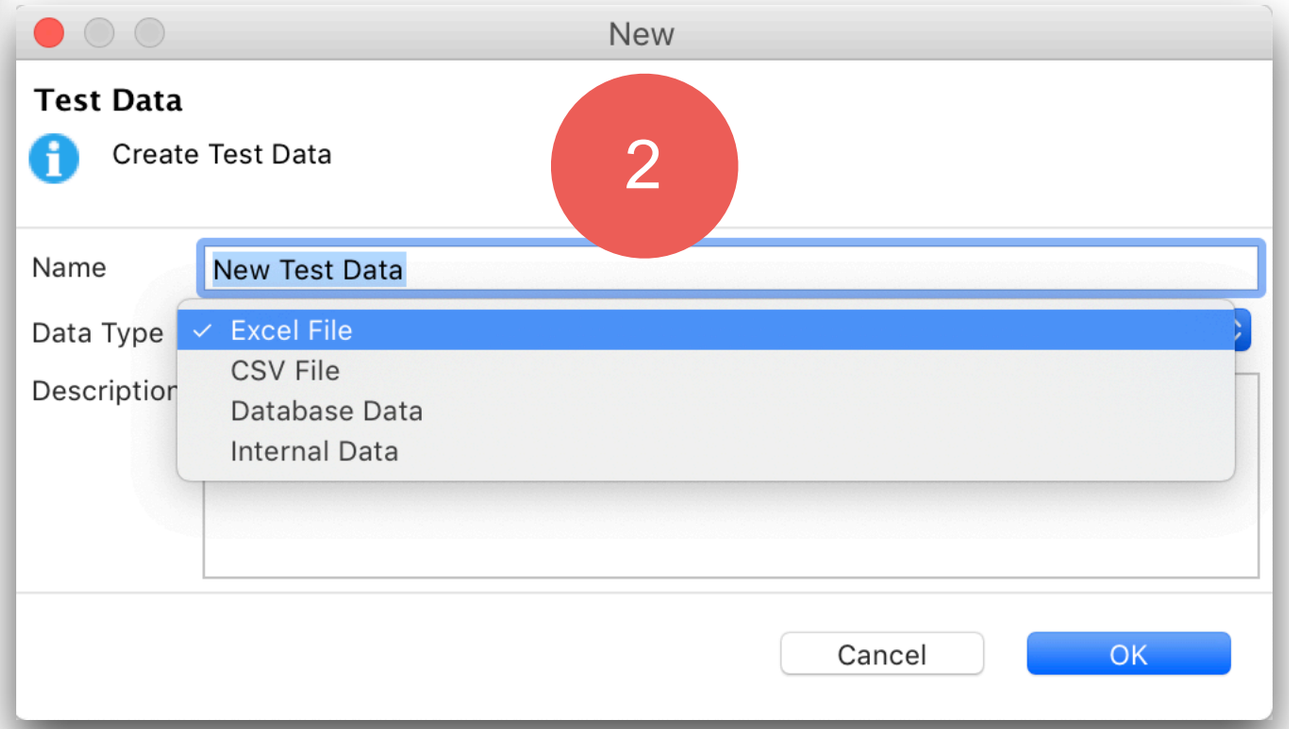
1. Create data test files
2. Define variables and add to test case
3.1 Create test suites
3.2 Pick data from data test files
4. Execute test suites

Data files

Data/ Datasource files (test data)
Different type of files
Microsoft Excel, CSV, Internal, Database

1. Create data files (1)





1. Create data files (2)

File keywords.csv

3

keywords.csv

```
1   User,Password
2   user01,pass
3   user02,pass
4   user03,pass
5   user04,pass
```

4

Read data from CSV file

keyword

Help

File's Information

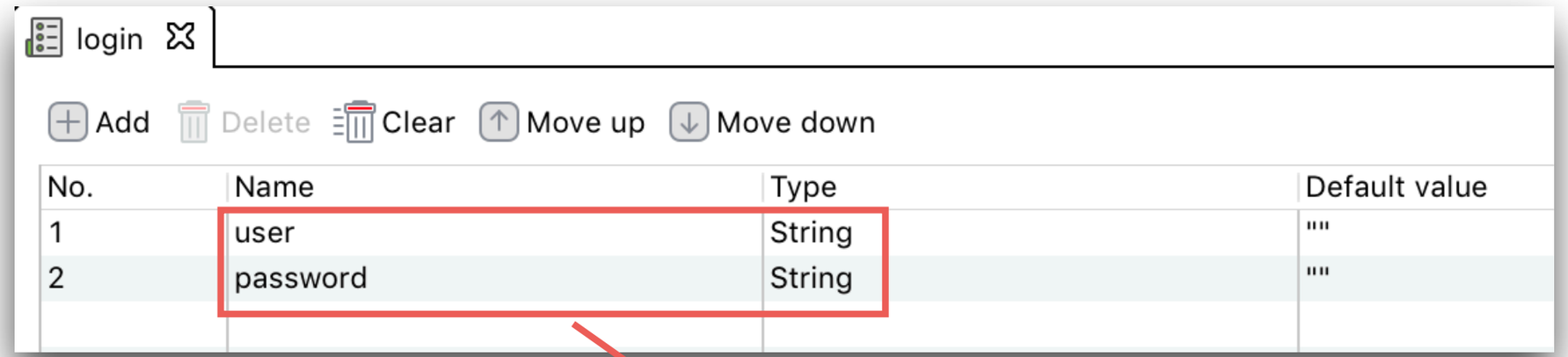
File Name: /Users/somkiat/data/sli [Browse...] Separator: COMMA

☑ Use first row as header ☐ Use relative path

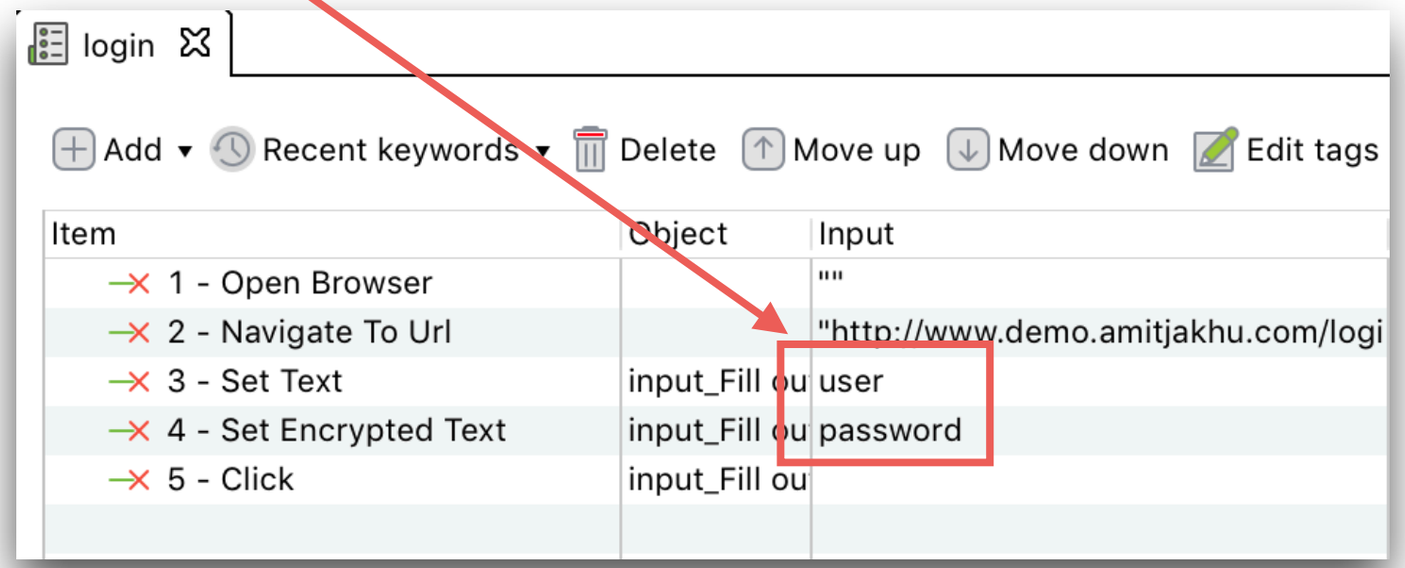
No.	User	Password
1	user01	pass
2	user02	pass
3	user03	pass
4	user04	pass

2. Create test case and variables

Variables tab



Manual tab :: using variable in input



3. Create test suite (1)

Add test cases and **show data binding**

LoginSuite

Help

1 Add data file

Execution Information

Add | Delete | View Execution History | Hide Data Binding

Enter text to search...

	No.	ID	Description	Run
	1	Test Cases/login		☑

Test Data

Add ▾ | Delete | Move Up | Move Down | Map All

	No.	ID	Data Iteration	Type
	1	Data Files/keyword	All	One

Variable Binding

Set Type ▾ | Set Test Data ▾

	No.	Name	Default value	Type	Test Data	Value
	1	user	''	Data Column	1 - Data Files/keyword	User
	2	password	''	Data Column	1 - Data Files/keyword	Password

Main | Script | Integration | Result

3. Create test suite (2)

Add test cases and **show data binding**

LoginSuite

Help

Execution Information

Add | Delete | View Execution History | Hide Data Binding

Enter text to search...

No.	ID	Description	Run
1	Test Cases/login		☑

Test Data

Add | Delete | Move Up | Move Down | Map All

No.	ID	Data Iteration	Type
1	Data Files/keyword	All	One

Variable Binding

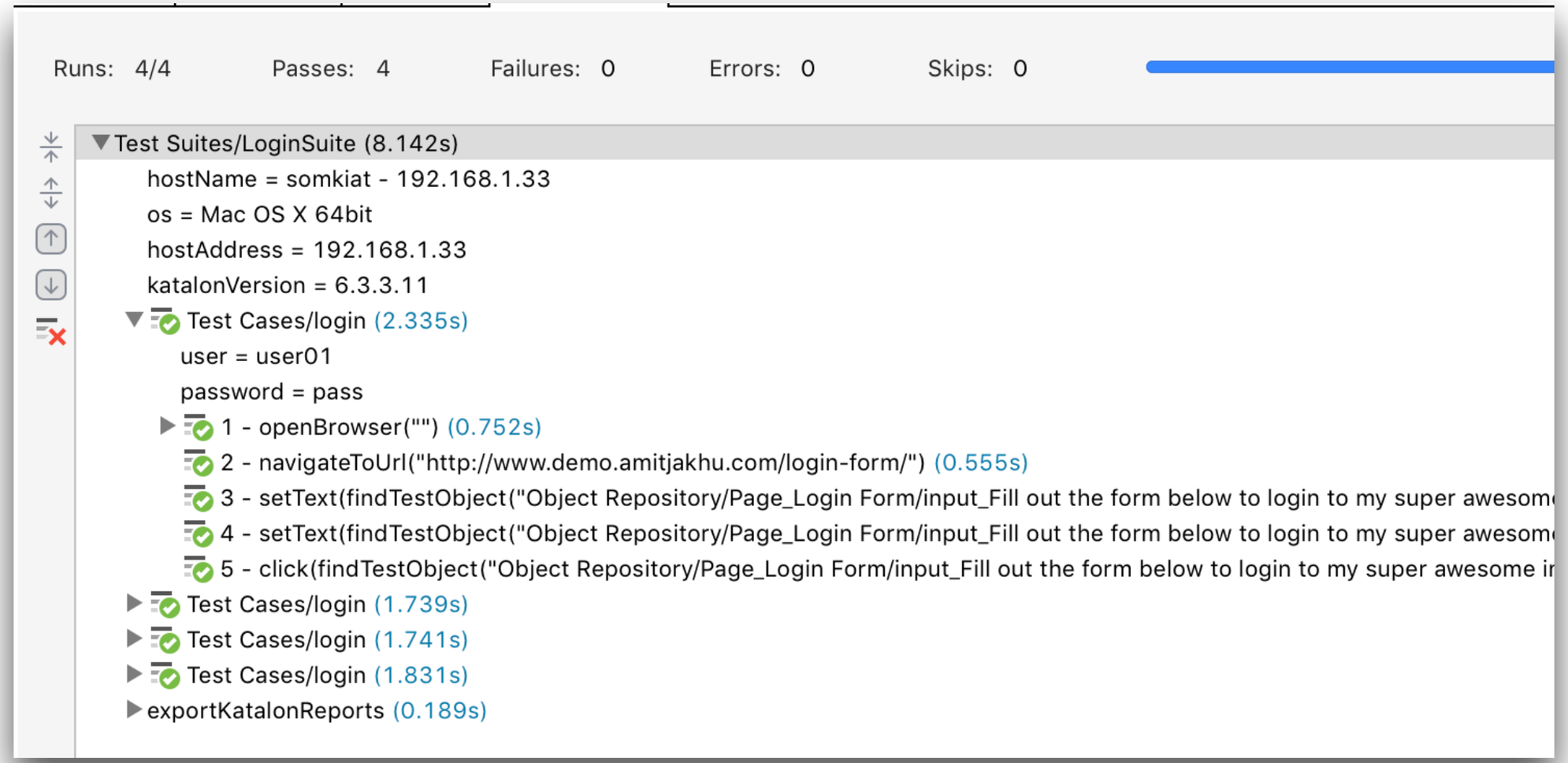
Set Type | Set Test Data

No.	Name	Default value	Type	Test Data	Value
1	user	''	Data Column	1 - Data Files/keyword	User
2	password	''	Data Column	1 - Data Files/keyword	Password

2 Binding data variables with test data

Main | Script | Integration | Result

4. Execute test suite



4. Execute test suite

TEST SUITE: **LoginSuite**

Full Name: LoginSuite
Start / End / Elapsed: 2019-10-23 03:14:00.898 / 2019-10-23 03:14:09.040 / 00:00:08.142
Status: 4 test total, 4 passed, **0 failed**, **0 error**, 0 incomplete

TEST CASE: **Test Cases/login**

Full Name: LoginSuite/Test Cases/login
Start / End / Elapsed: 2019-10-23 03:14:01.079 / 2019-10-23 03:14:03.414 / 00:00:02.335
Status: PASSED

TEST STEP: *openBrowser("")*

TEST STEP: *navigateToUrl("http://www.demo.amitjakhu.com/login-form/")*

TEST STEP: *setText(findTestObject("Object Repository/Page_Login Form/input_Fill out the form below to login to my super awesome imaginary control panel_username"), user)*

TEST STEP: *setText(findTestObject("Object Repository/Page_Login Form/input_Fill out the form below to login to my super awesome imaginary control panel_password"), password)*

TEST STEP: *click(findTestObject("Object Repository/Page_Login Form/input_Fill out the form below to login to my super awesome imaginary control panel_submit"))*

TEST CASE: **Test Cases/login**

Full Name: LoginSuite/Test Cases/login
Start / End / Elapsed: 2019-10-23 03:14:03.430 / 2019-10-23 03:14:05.169 / 00:00:01.739
Status: PASSED

TEST STEP: *openBrowser("")*

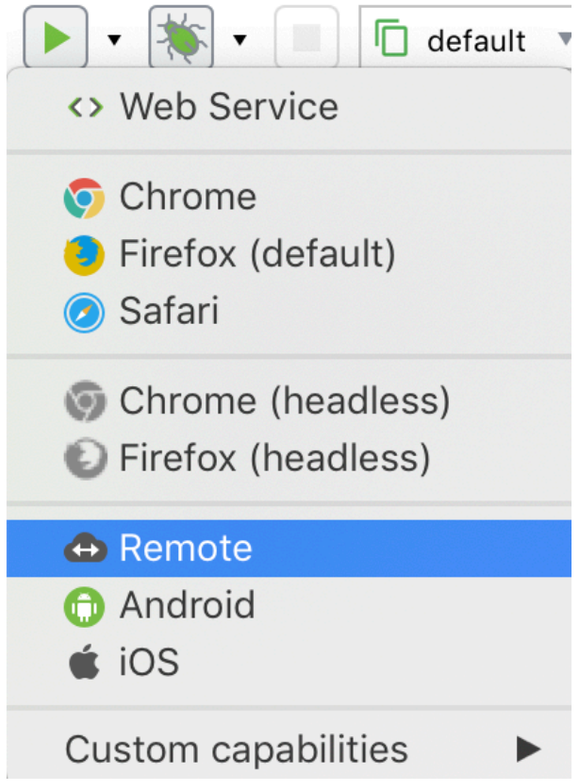
TEST STEP: *navigateToUrl("http://www.demo.amitjakhu.com/login-form/")*

TEST STEP: *setText(findTestObject("Object Repository/Page_Login Form/input_Fill out the form below to login to my super awesome imaginary control panel_username"), user)*

TEST STEP: *setText(findTestObject("Object Repository/Page_Login Form/input_Fill out the form below to login to my super awesome imaginary control panel_password"), password)*

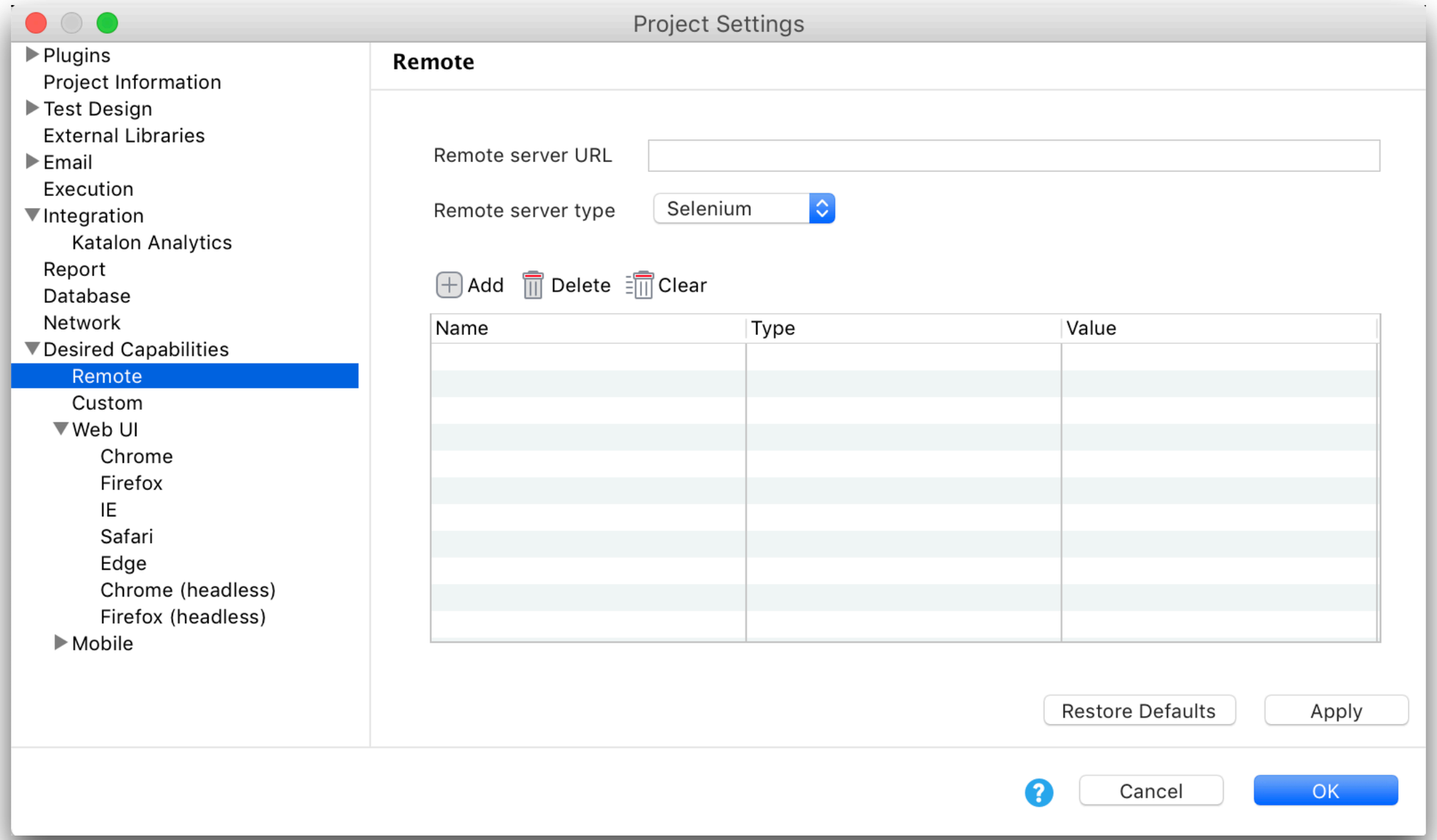
TEST STEP: *click(findTestObject("Object Repository/Page_Login Form/input_Fill out the form below to login to my super awesome imaginary control panel_submit"))*

Katalon with Remote

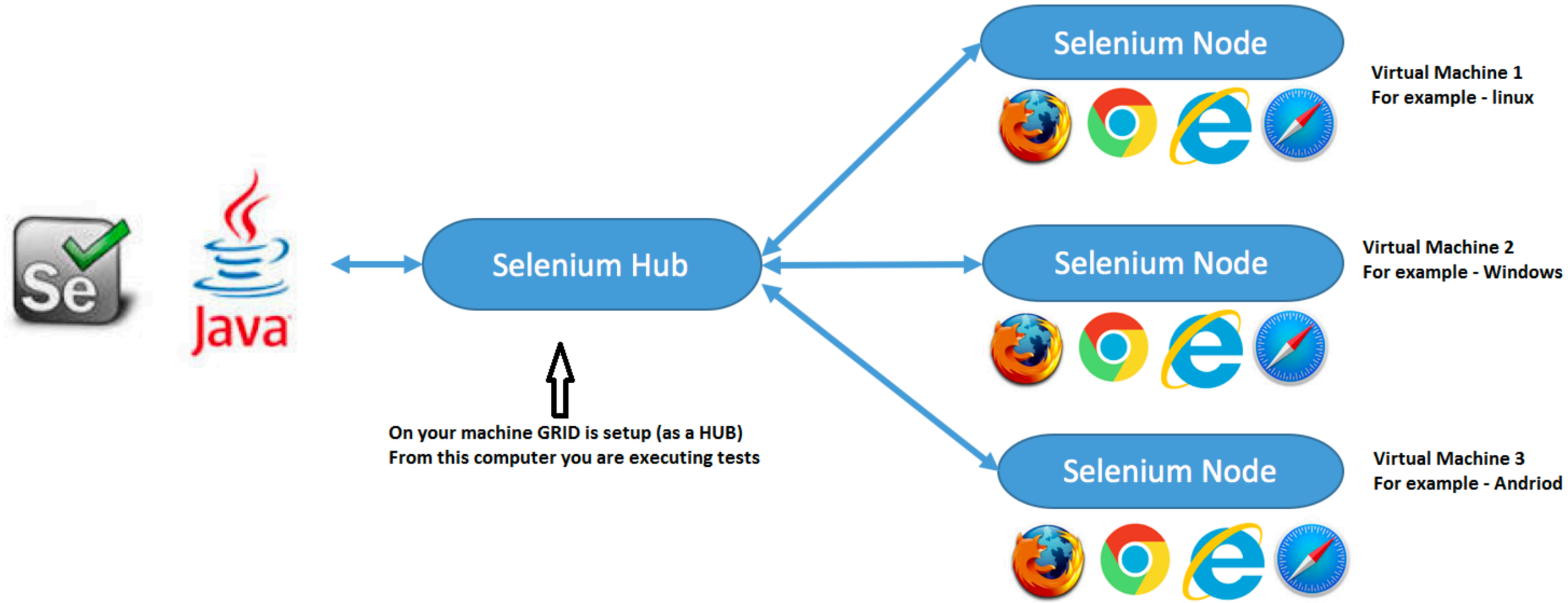


Project setting

Add remote url of **Selenium Grid**



Selenium Grid



https://github.com/SeleniumHQ/selenium/wiki/Grid2

API Testing

Import from other system

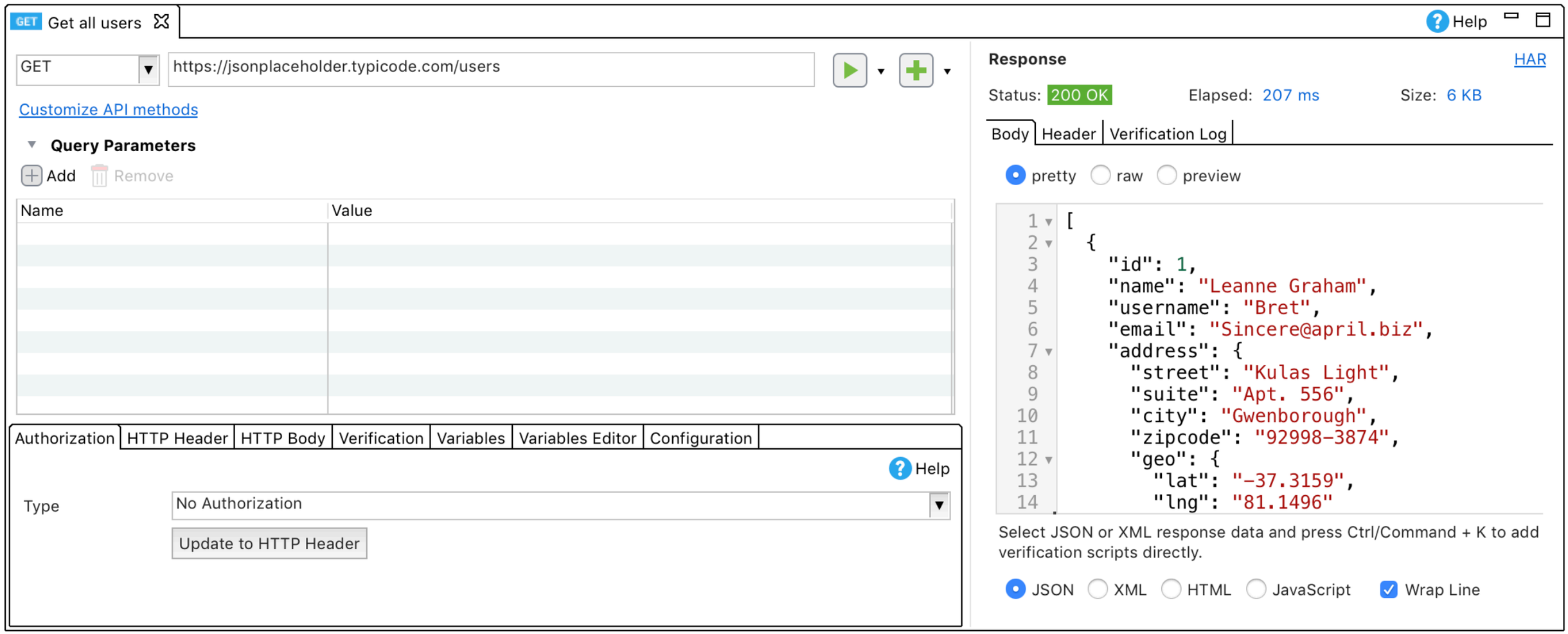
Import from swagger
Import from Postman
Import from WSDL (Web Service)

API Testing

Create REST request
Validate status code
Validate response data

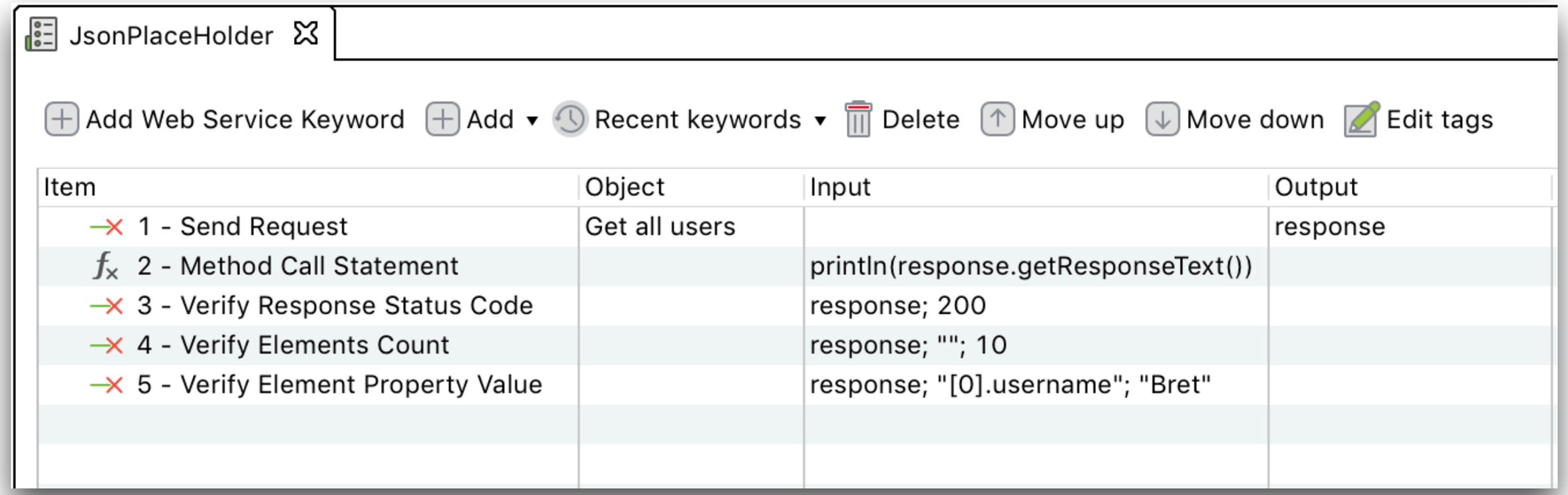
1. Create REST request

Save new request to Object repository



2. Create new test case

Save new request to Object repository
Add Web Service Keyword



3. Validate result

JsonPlaceHolder

Add Web Service Keyword | Add | Recent keywords | Delete | Move up | Move down | Edit tags

Item	Object	Input	Output
1 - Send Request	Get all users		response
2 - Method Call Statement		println(response.getResponseText())	
3 - Verify Response Status Code		response; 200	
4 - Verify Elements Count		response; ""; 10	
5 - Verify Element Property Value		response; "[0].username"; "Bret"	

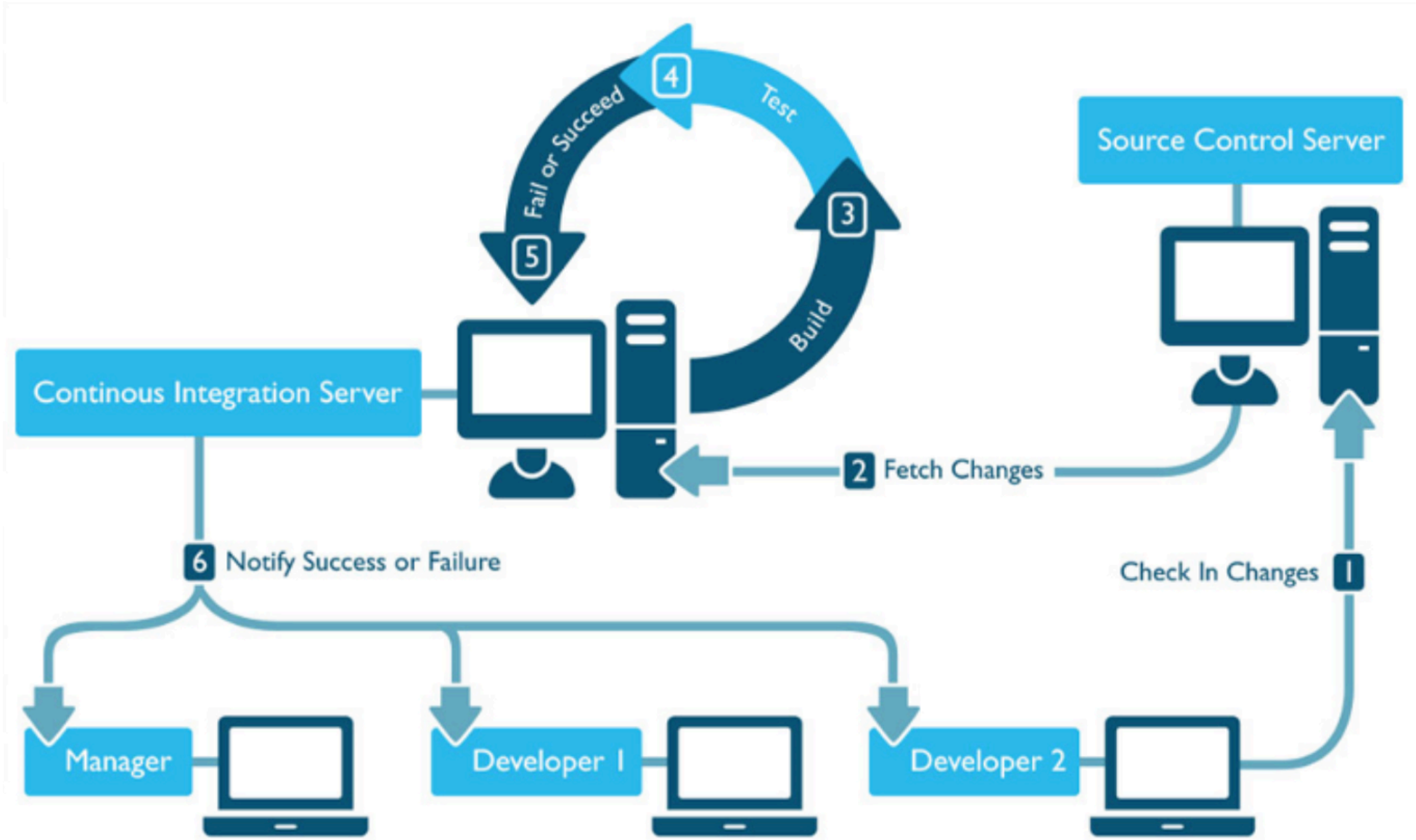
Print data to console

```
import static com.kms.katalon.core.checkpoint.CheckpointFactory.findCheckpoint

response = WS.sendRequest(findTestObject('Get all users'))

println(response.getResponseText())

WS.verifyResponseStatusCode(response, 200)

WS.verifyElementsCount(response, '', 10)

WS.verifyElementPropertyValue(response, '[0].username', 'Bret')
```

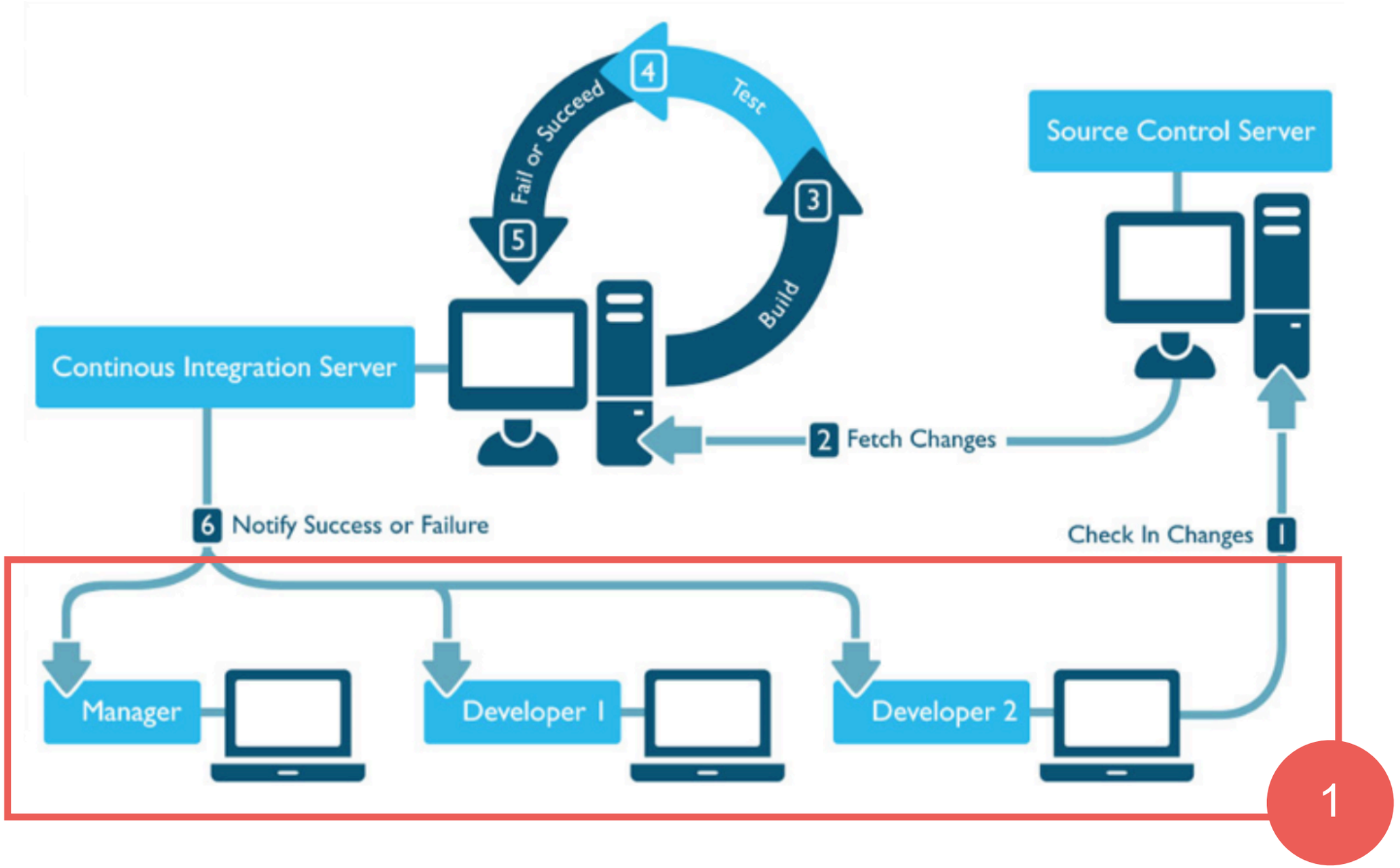
Continuous Testing

https://jenkins.io/

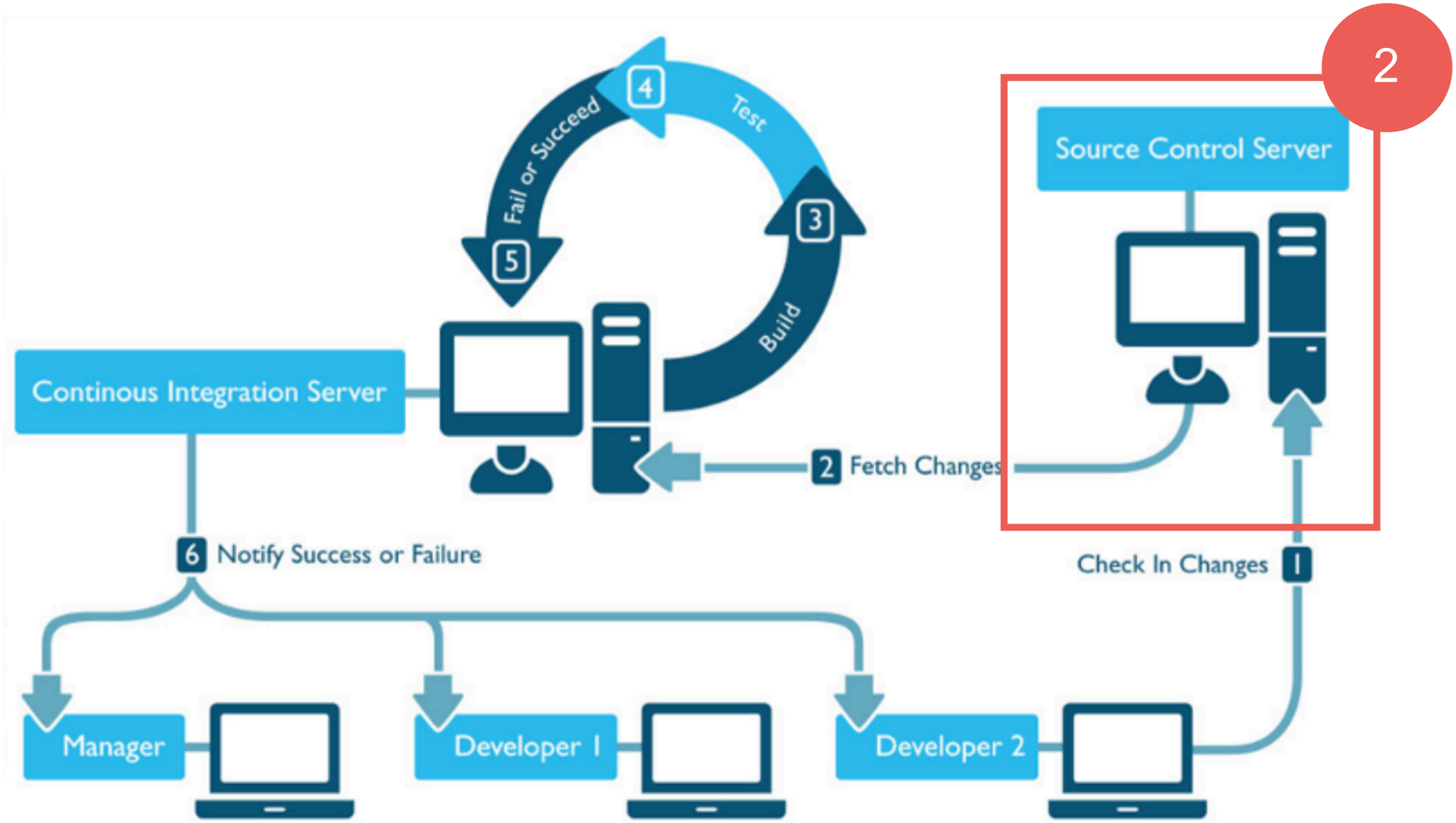
Continuous Integration and Testing



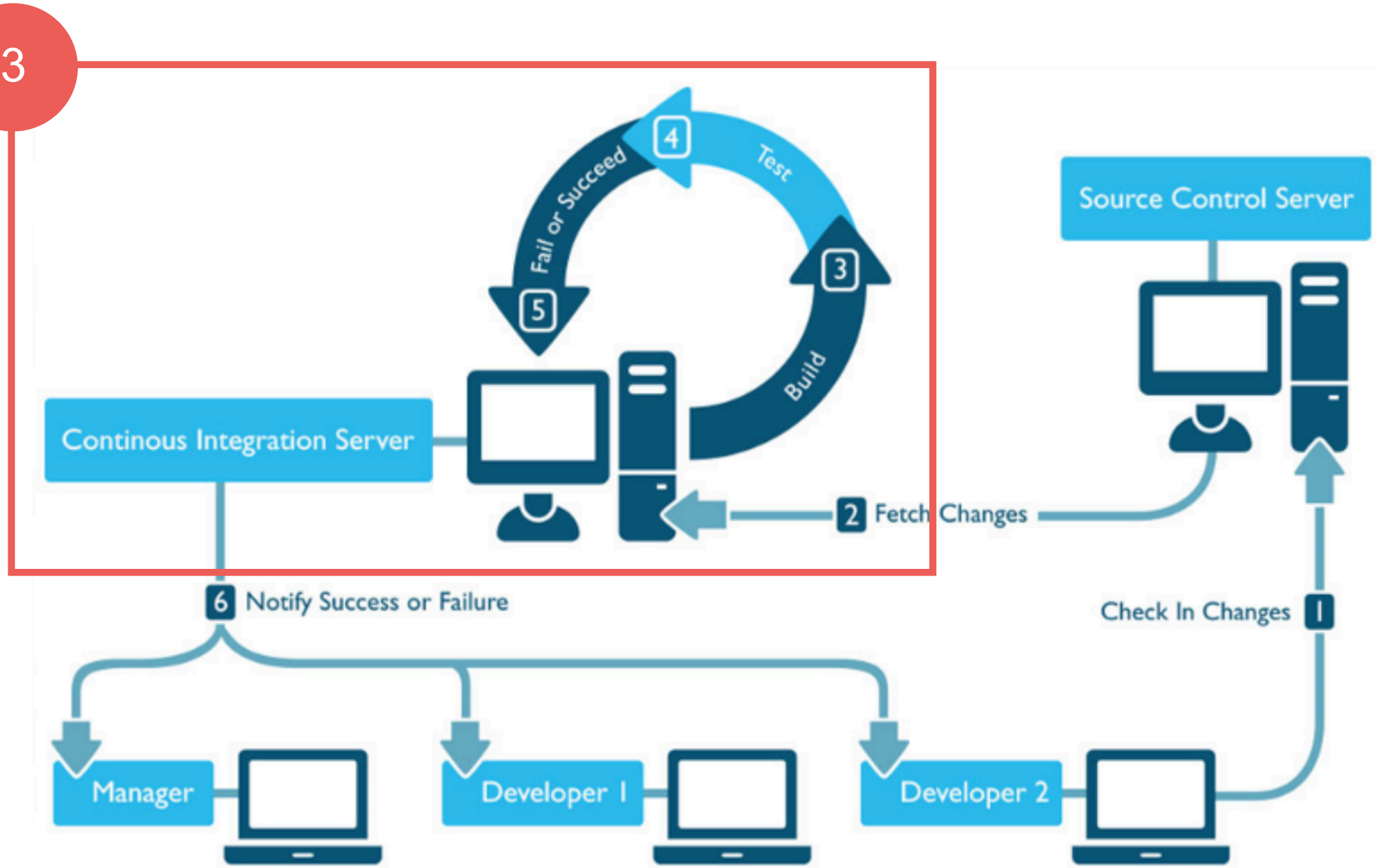
Continuous Integration and Testing



Continuous Integration and Testing



Continuous Integration and Testing



Start Jenkins server

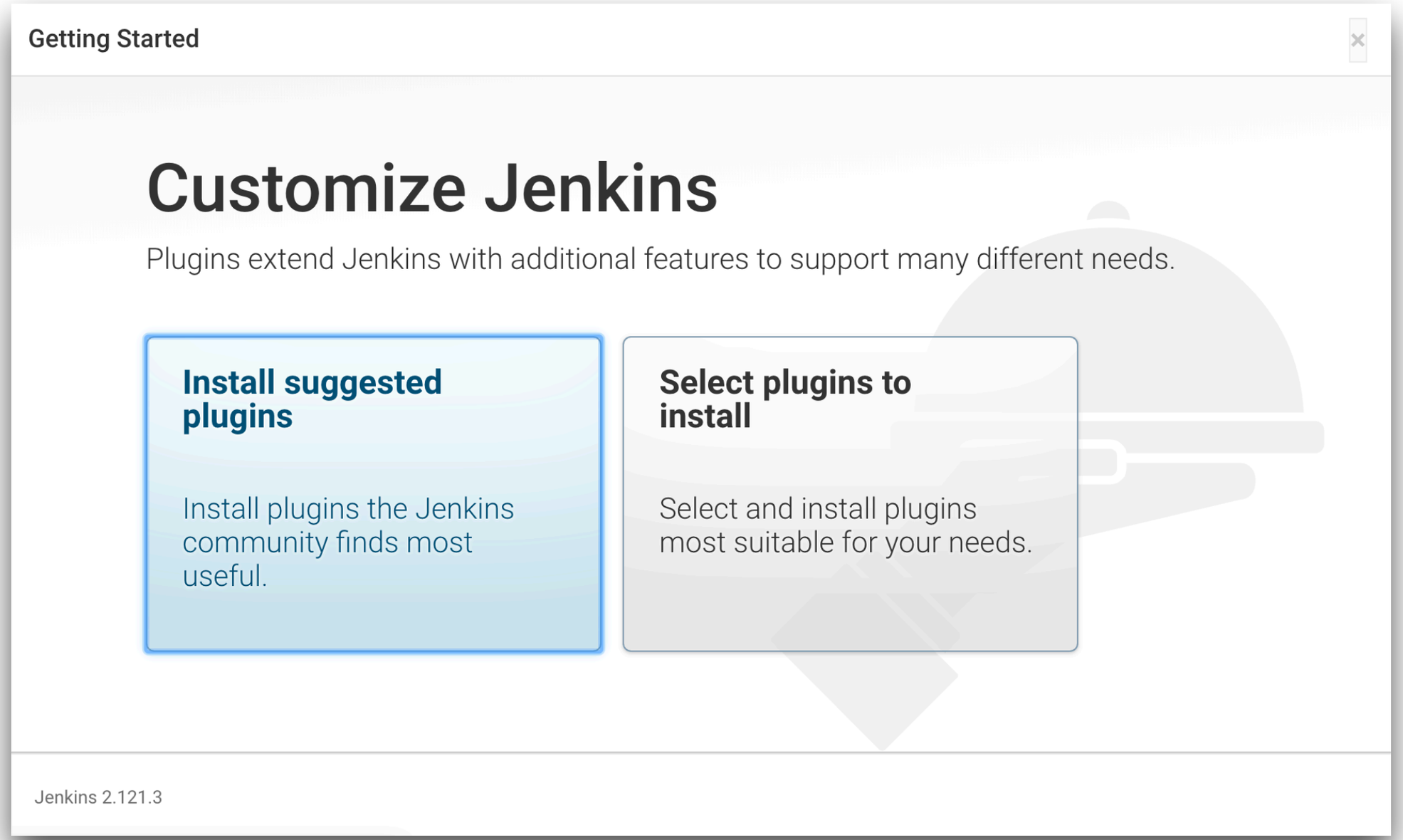
$java -jar jenkins.war

```
Jenkins initial setup is required. An admin user
been created and a password generated.
Please use the following password to proceed to :
llation:

07be76a843f54c3bae1d259ee38ed2d9

This may also be found at: /Users/somkiat/data/s
robot-framework/thaibev-20180806/ci/v2/secrets/i
lAdminPassword
```

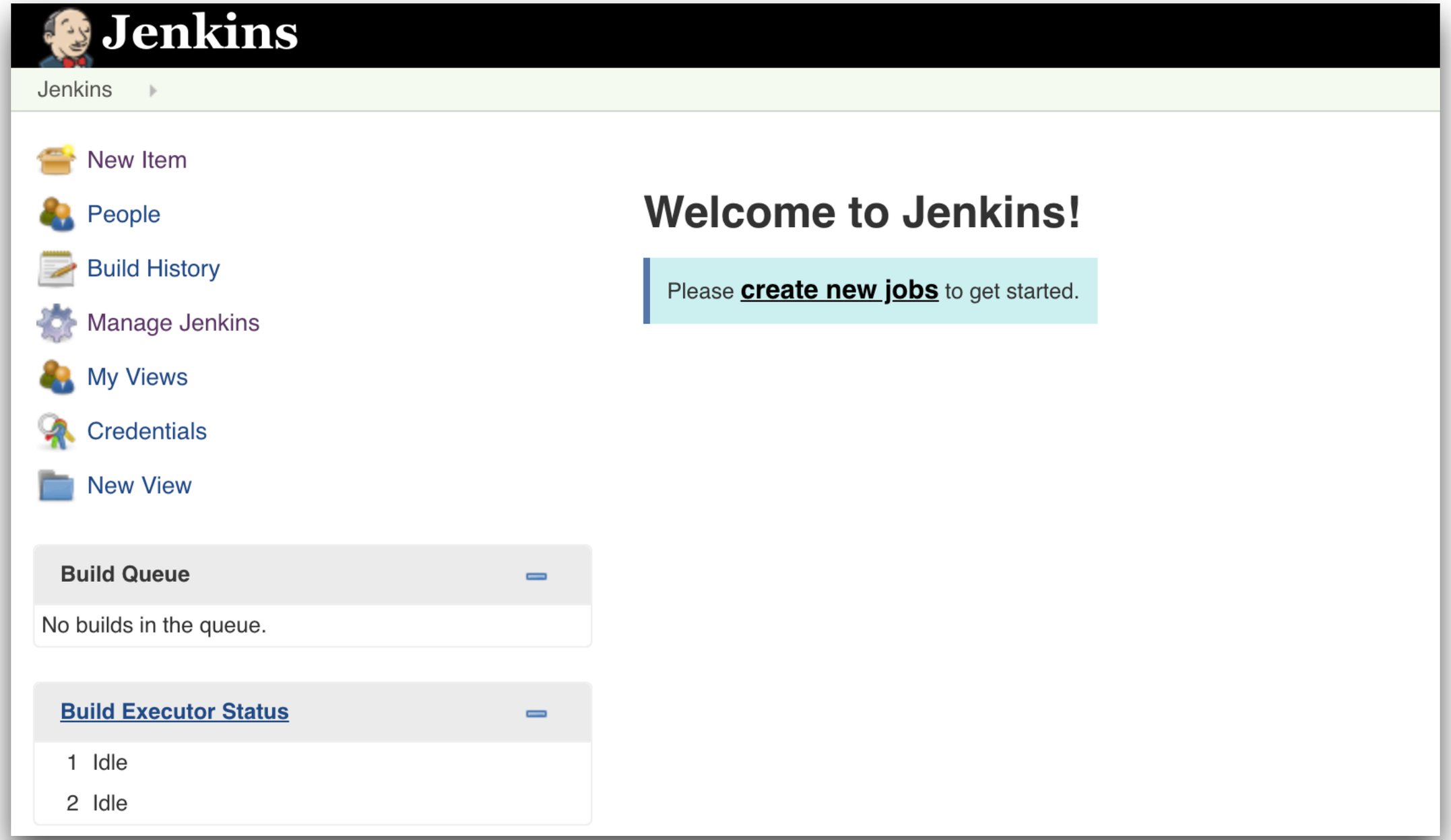
Open in browser

http://localhost:8080

Install suggested plugins



Ready to start !!



1. Create new job

Enter an item name

Fill in job name

» *Required field*

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

GitHub Organization
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK

2. Source code management

General | **Source Code Management** | Build Triggers | Build Environment | Build | Post-build Actions

Source Code Management

- (•) None
- ( ) Git
- ( ) Subversion

Build Triggers

- [ ] Trigger builds remotely (e.g., from scripts)
- [ ] Build after other projects are built
- [ ] Build periodically
- [ ] GitHub hook trigger for GITScm polling
- [ ] Poll SCM

3. Build trigger

General | **Source Code Management** | Build Triggers | Build Environment | Build | Post-build Actions

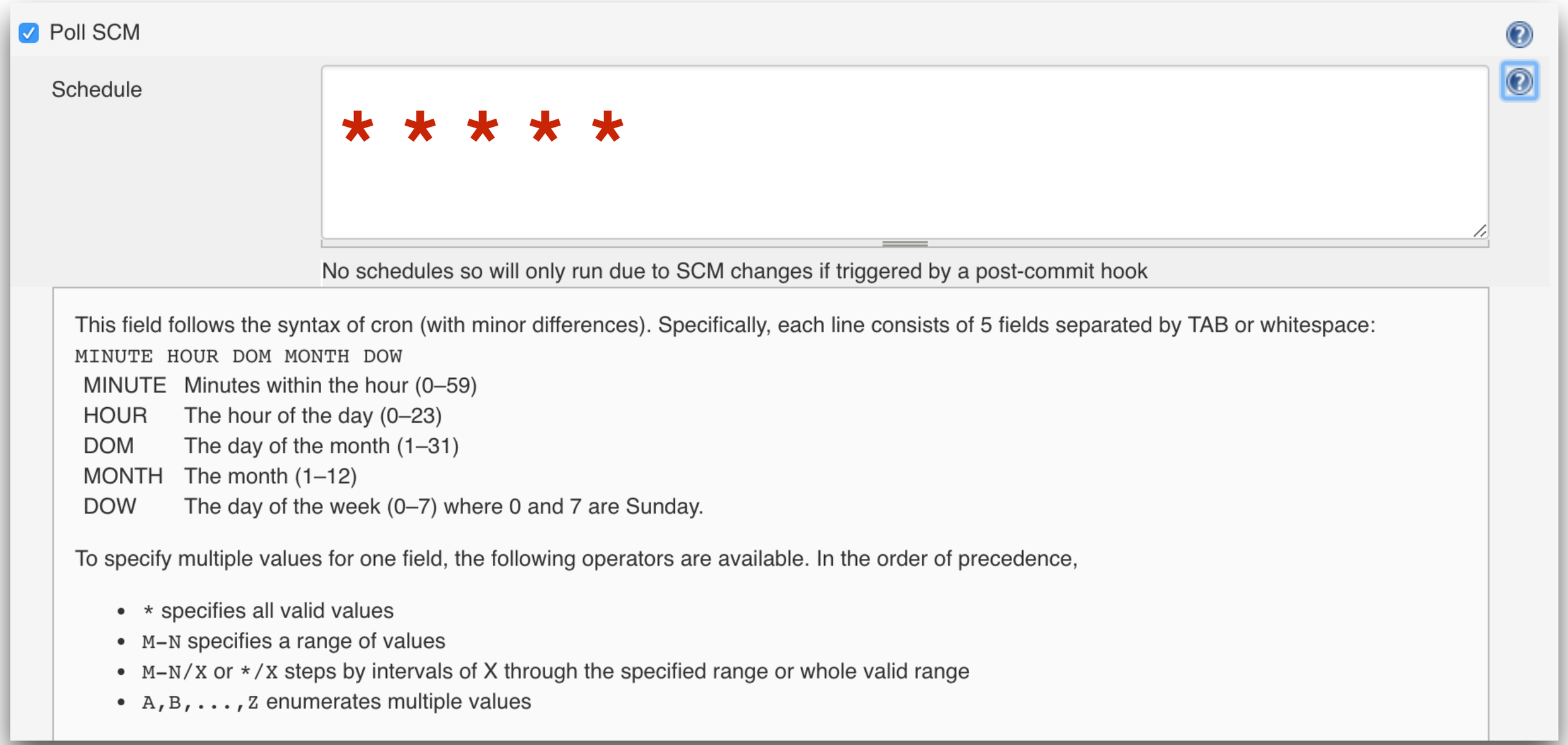
Source Code Management

- None
- Git
- Subversion

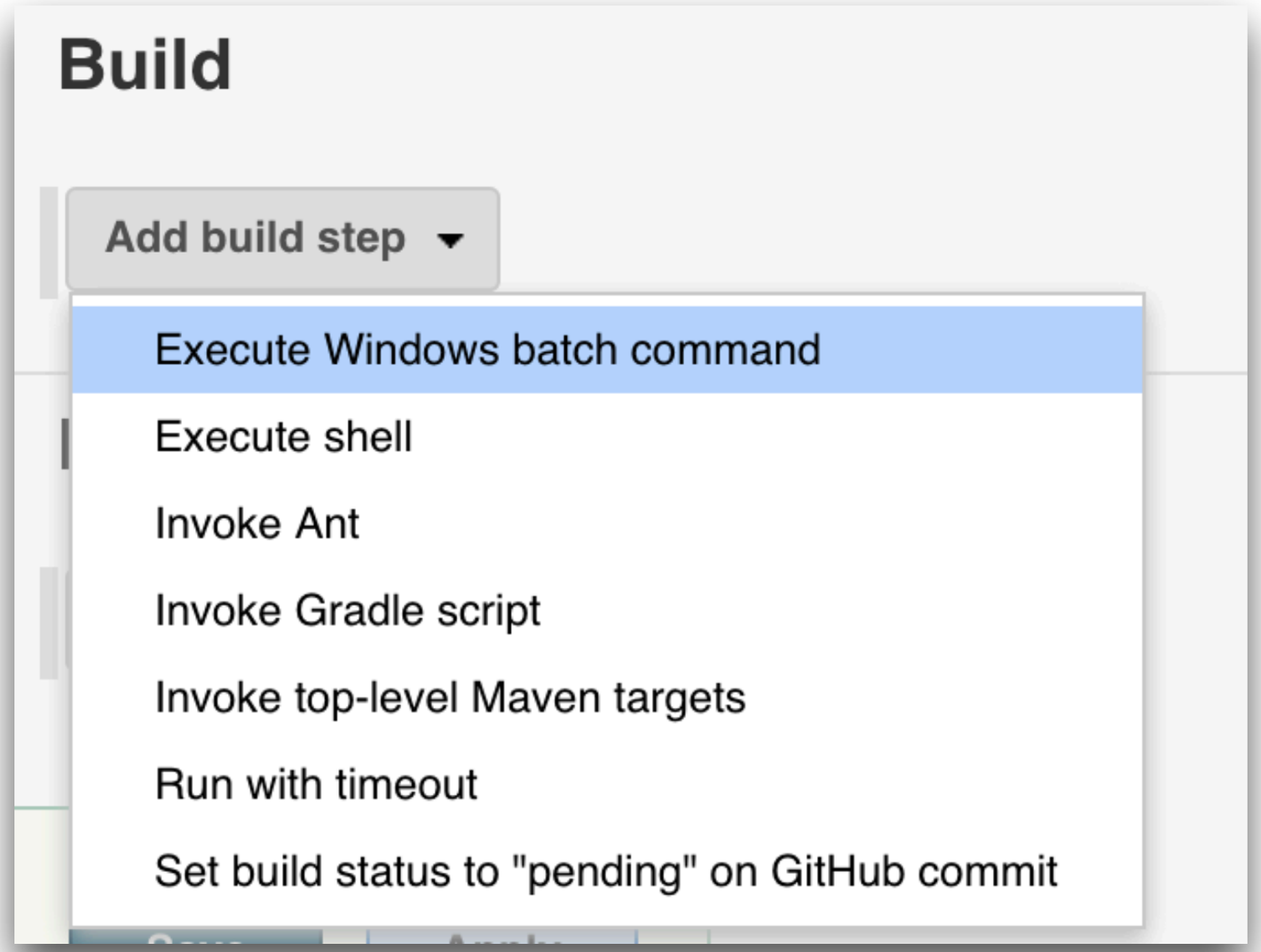
Build Triggers

- Trigger builds remotely (e.g., from scripts)
- Build after other projects are built
- Build periodically
- GitHub hook trigger for GITScm polling
- Poll SCM

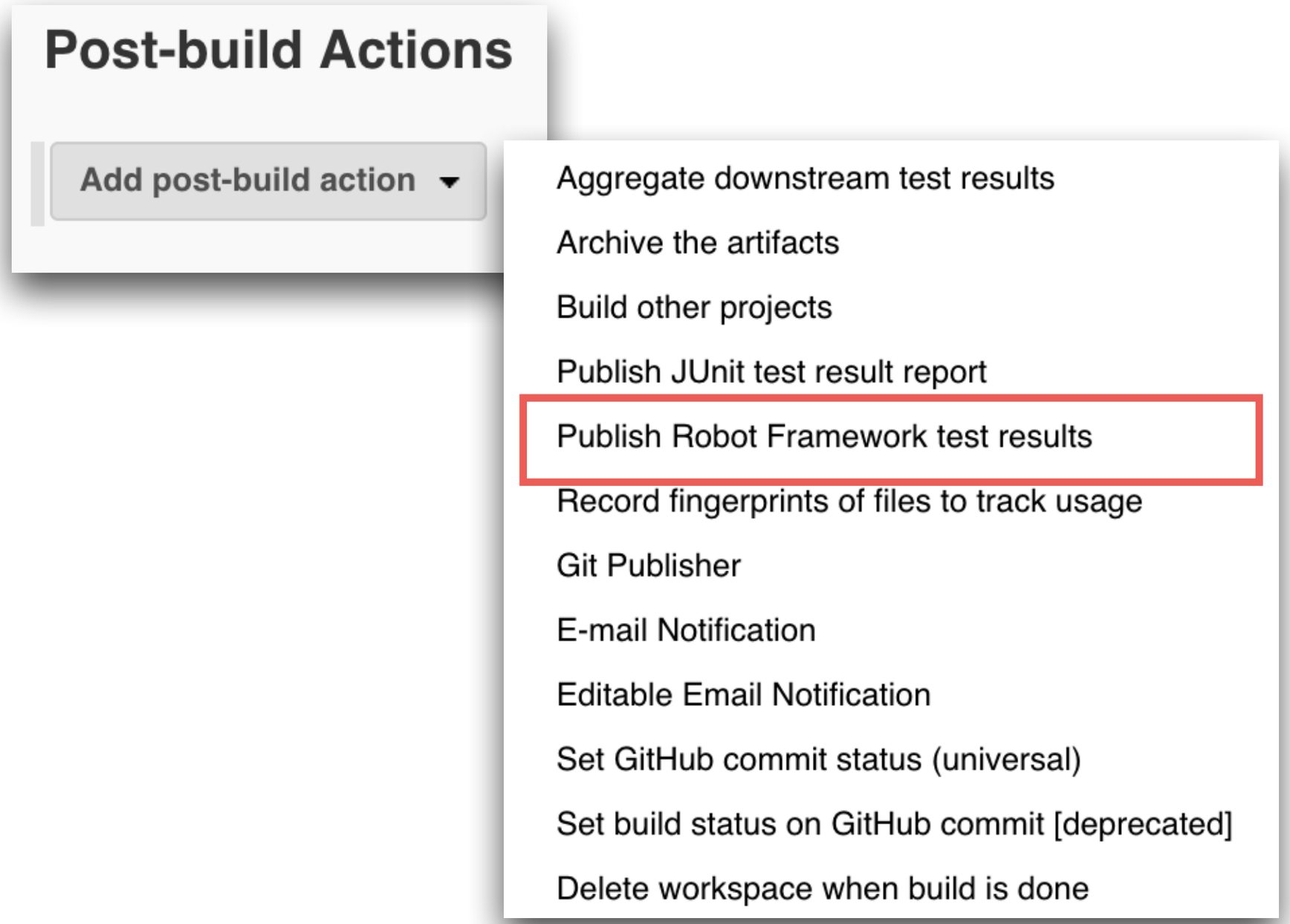
4. Build trigger with poll SCM



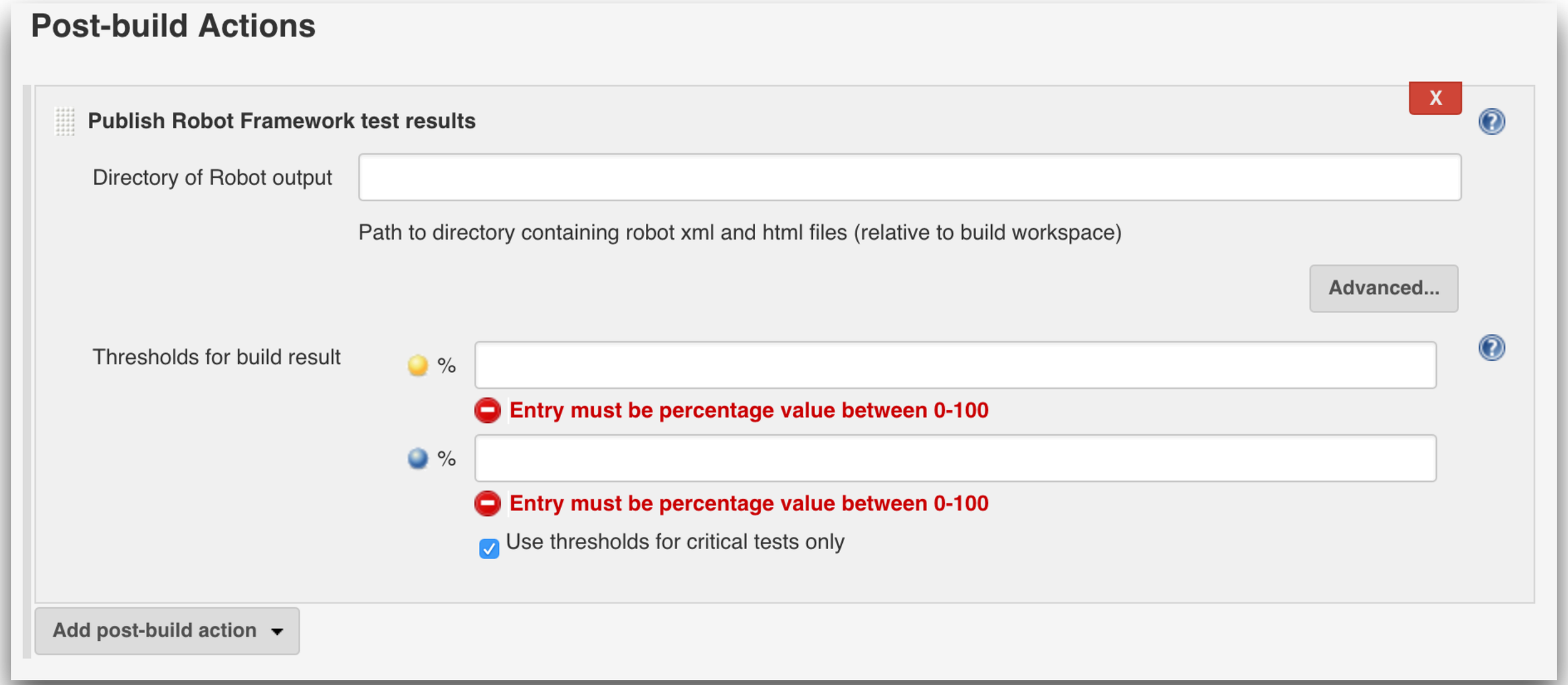
5. Build your job



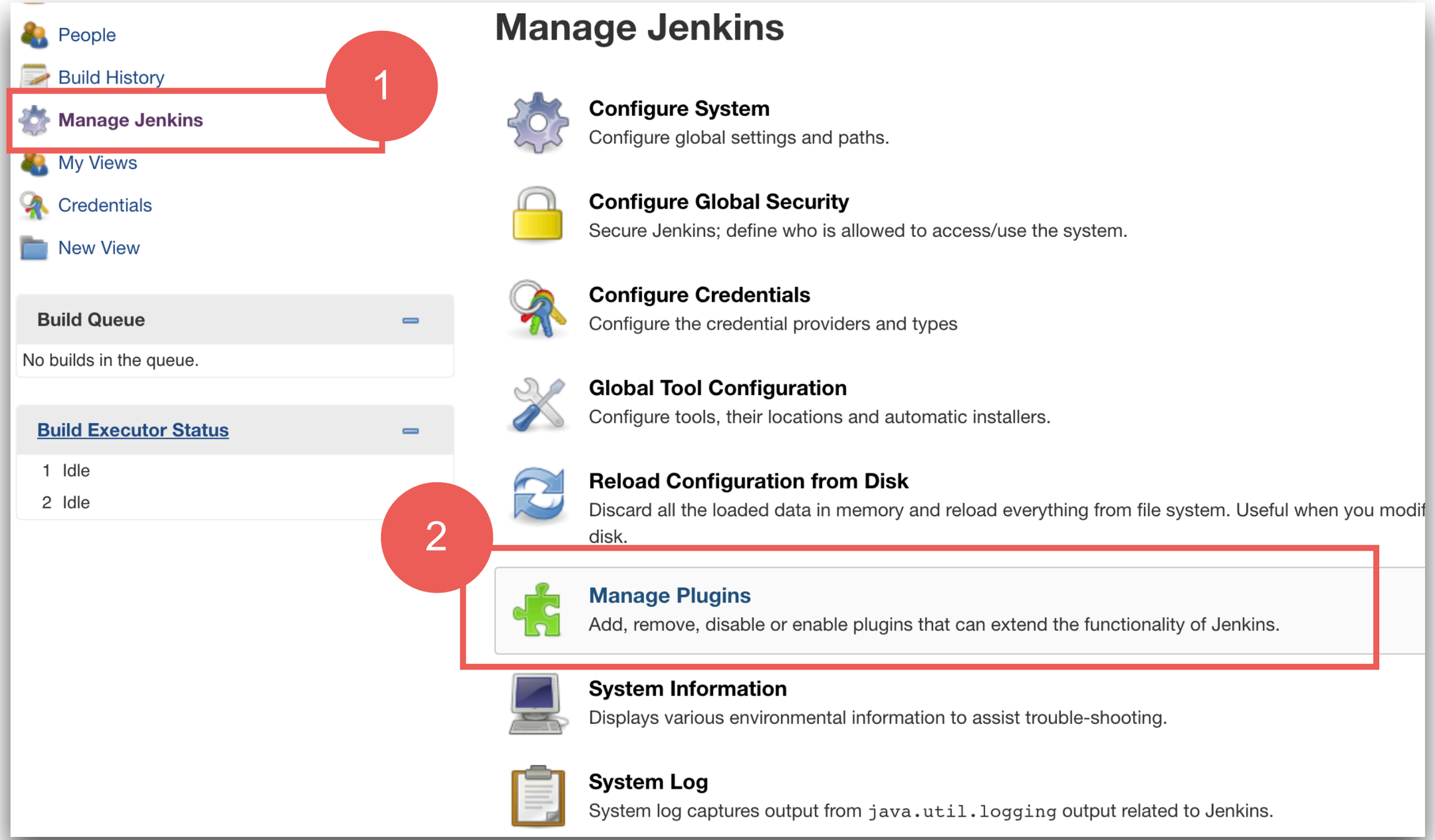
6. Post build actions



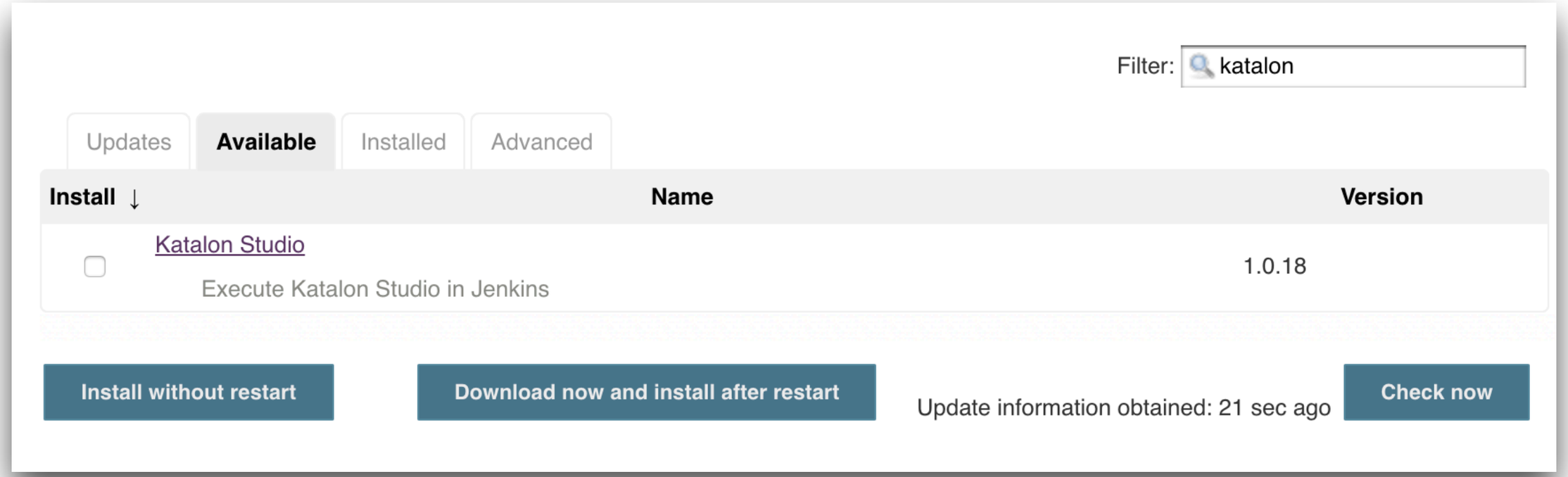
7. Add post build with Robot framework



8. Install Katalon plugin



8. Install Katalon plugin

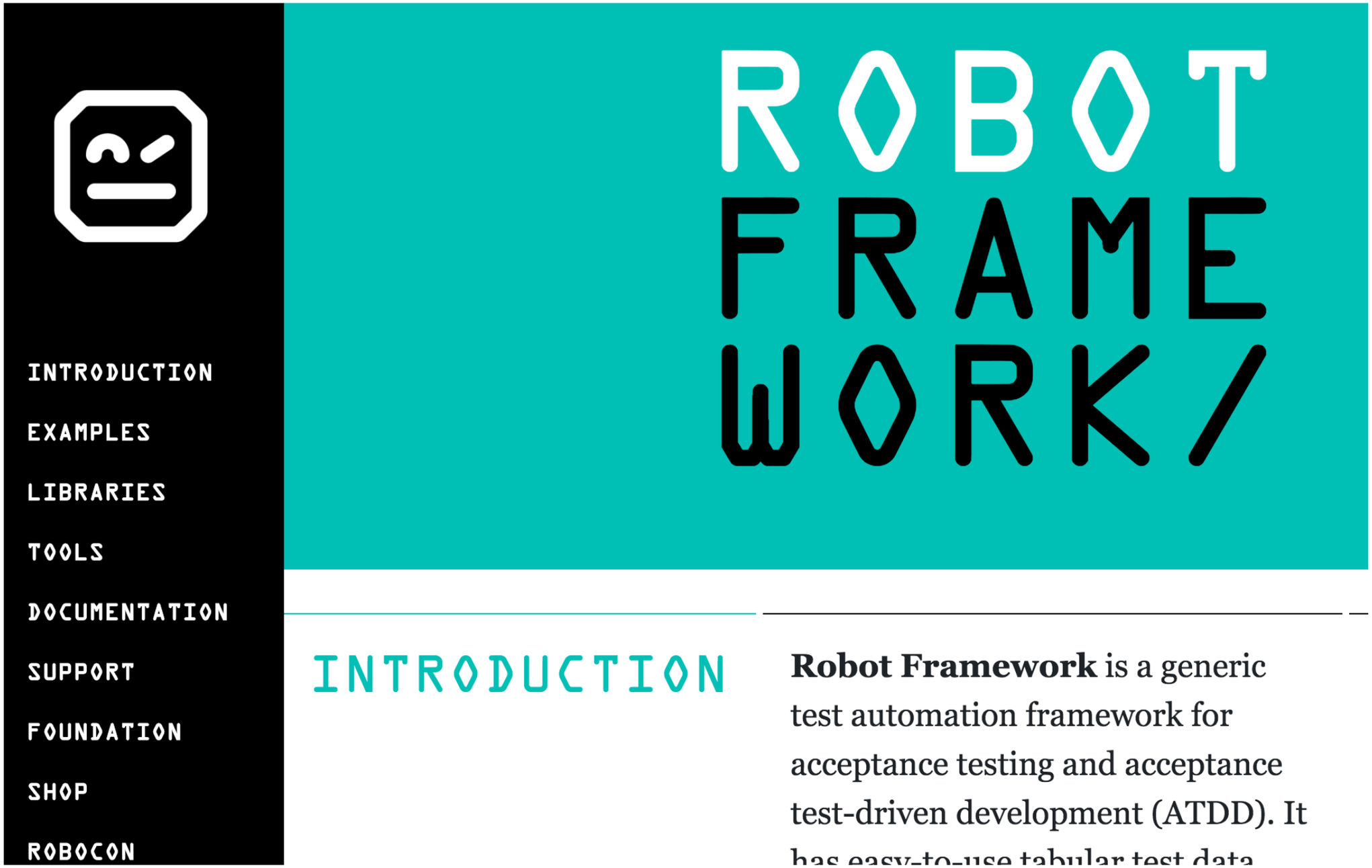


https://plugins.jenkins.io/katalon

Robot Framework

https://robotframework.org/

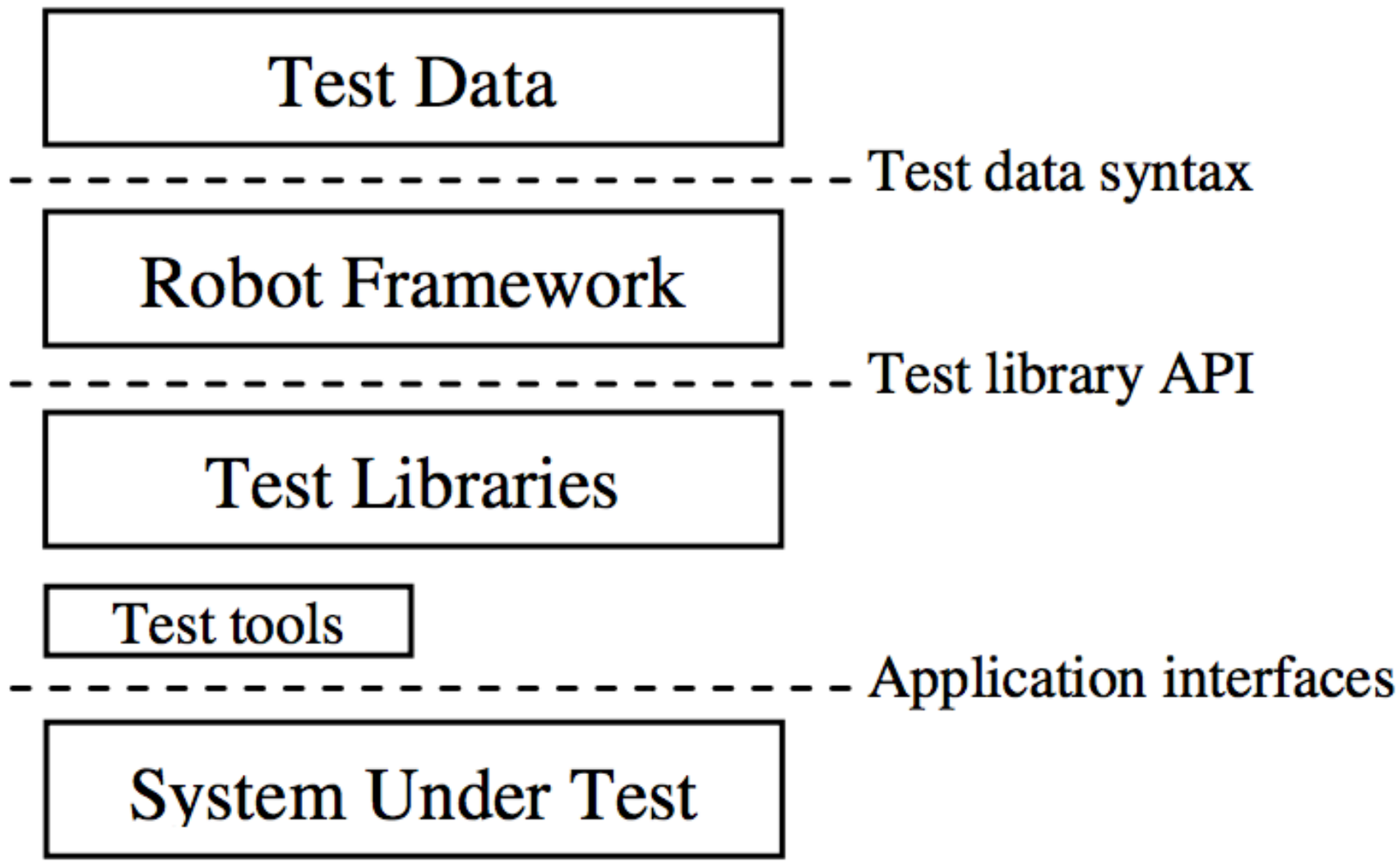
Official website



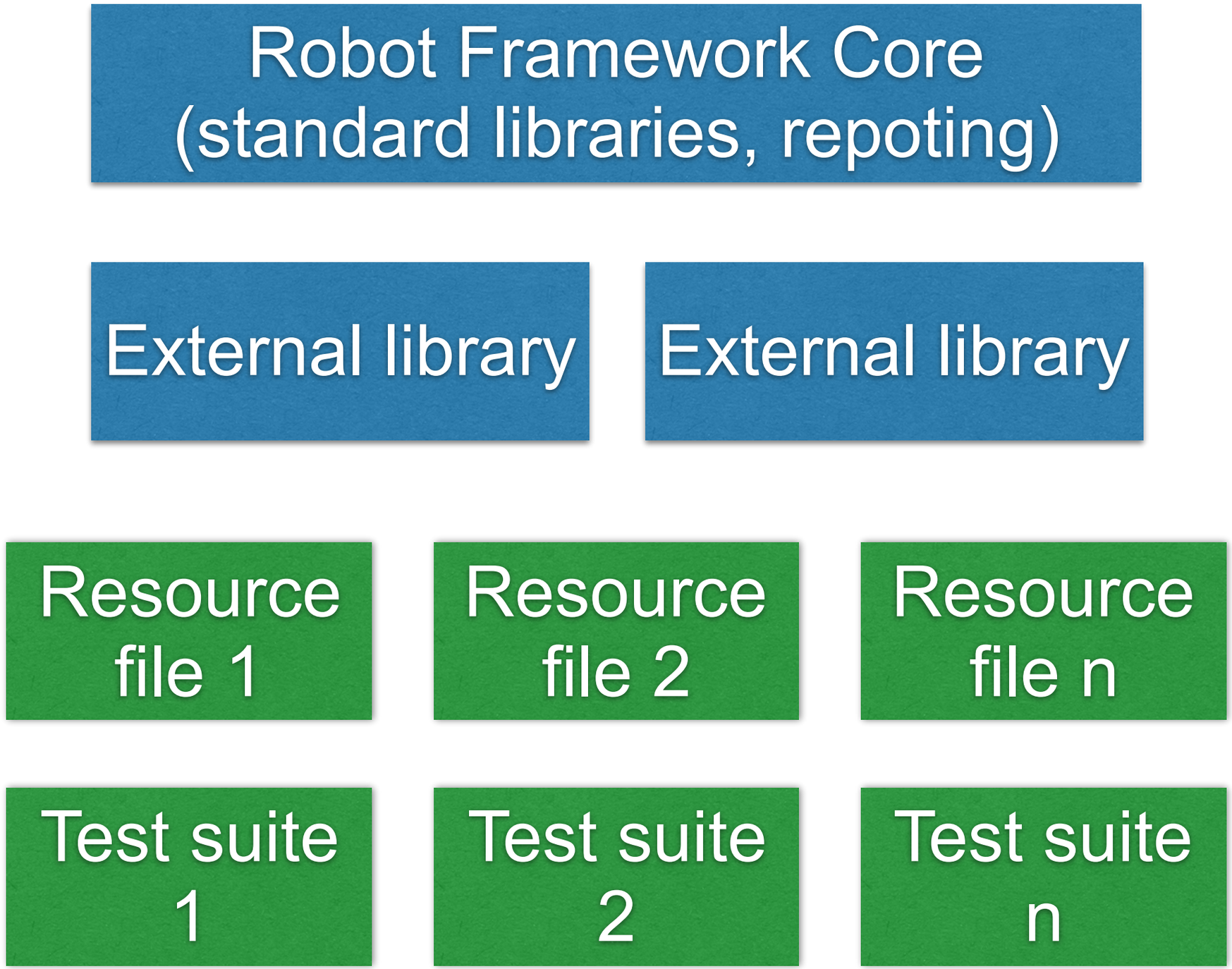
Robot Framework

Test automation framework
Designed for Acceptance testing(ATDD)
Developed with Python
Compatible with Java (Jython)

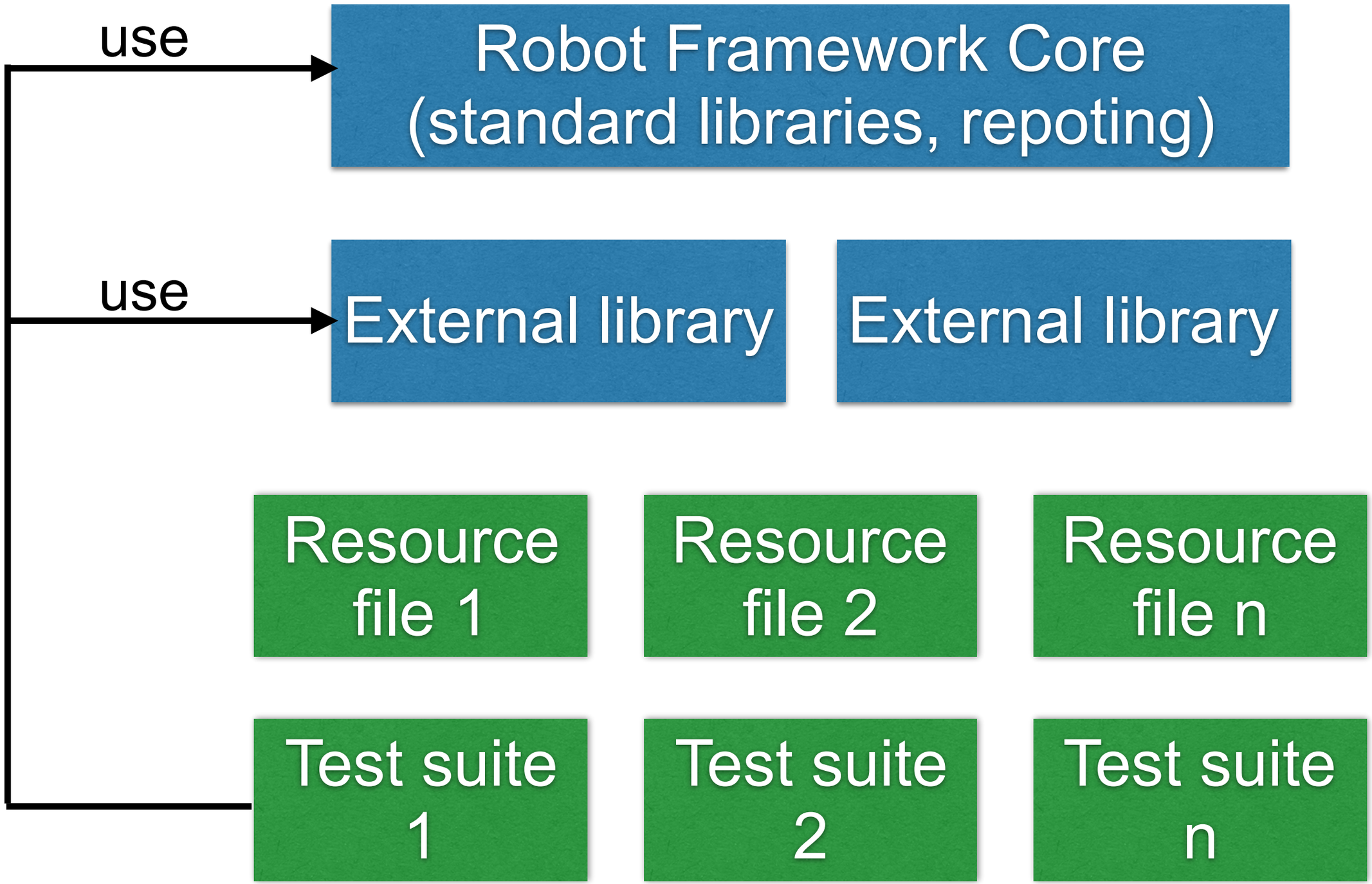
Robot Framework architecture



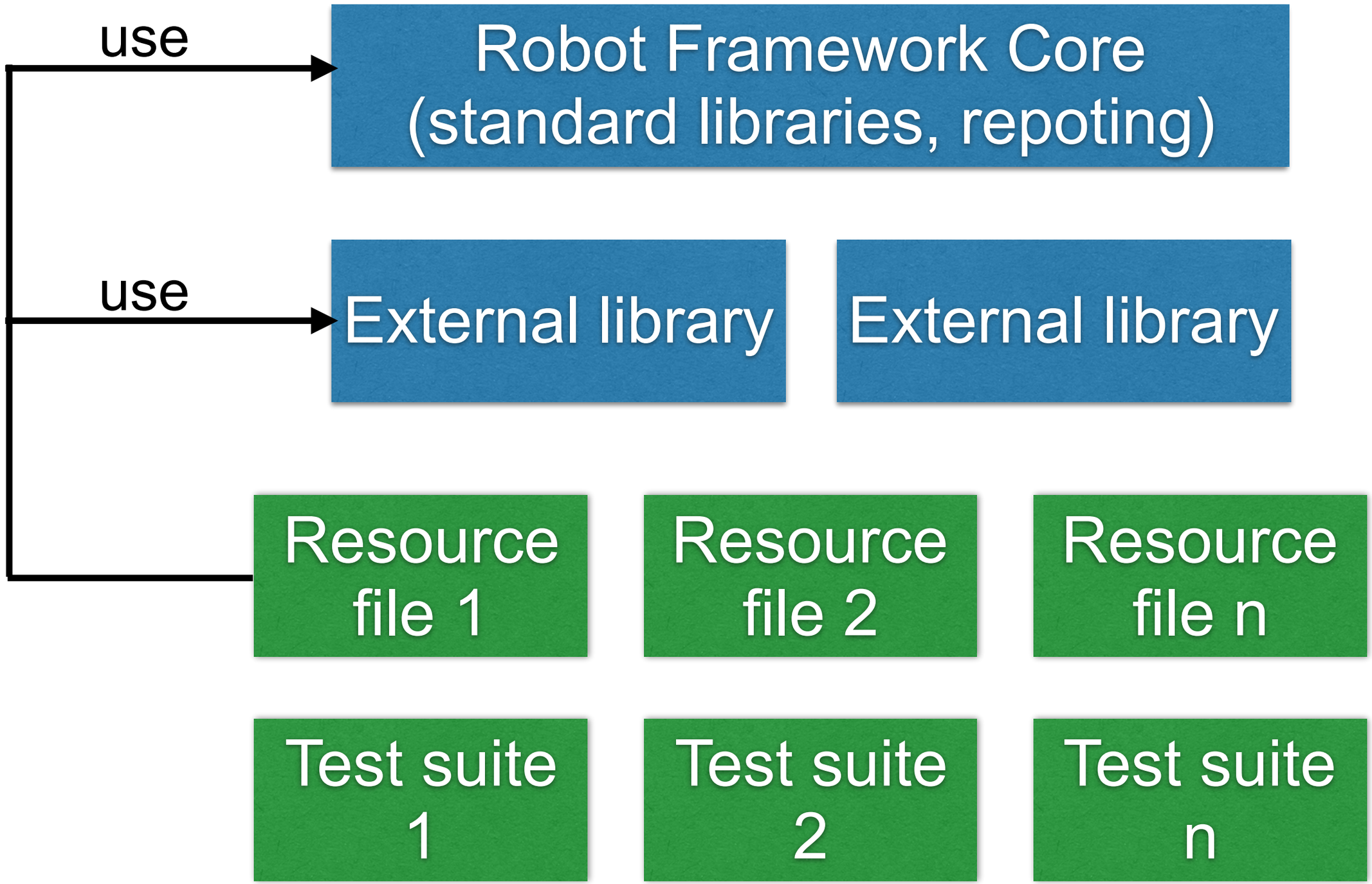
Robot Framework architecture (1)



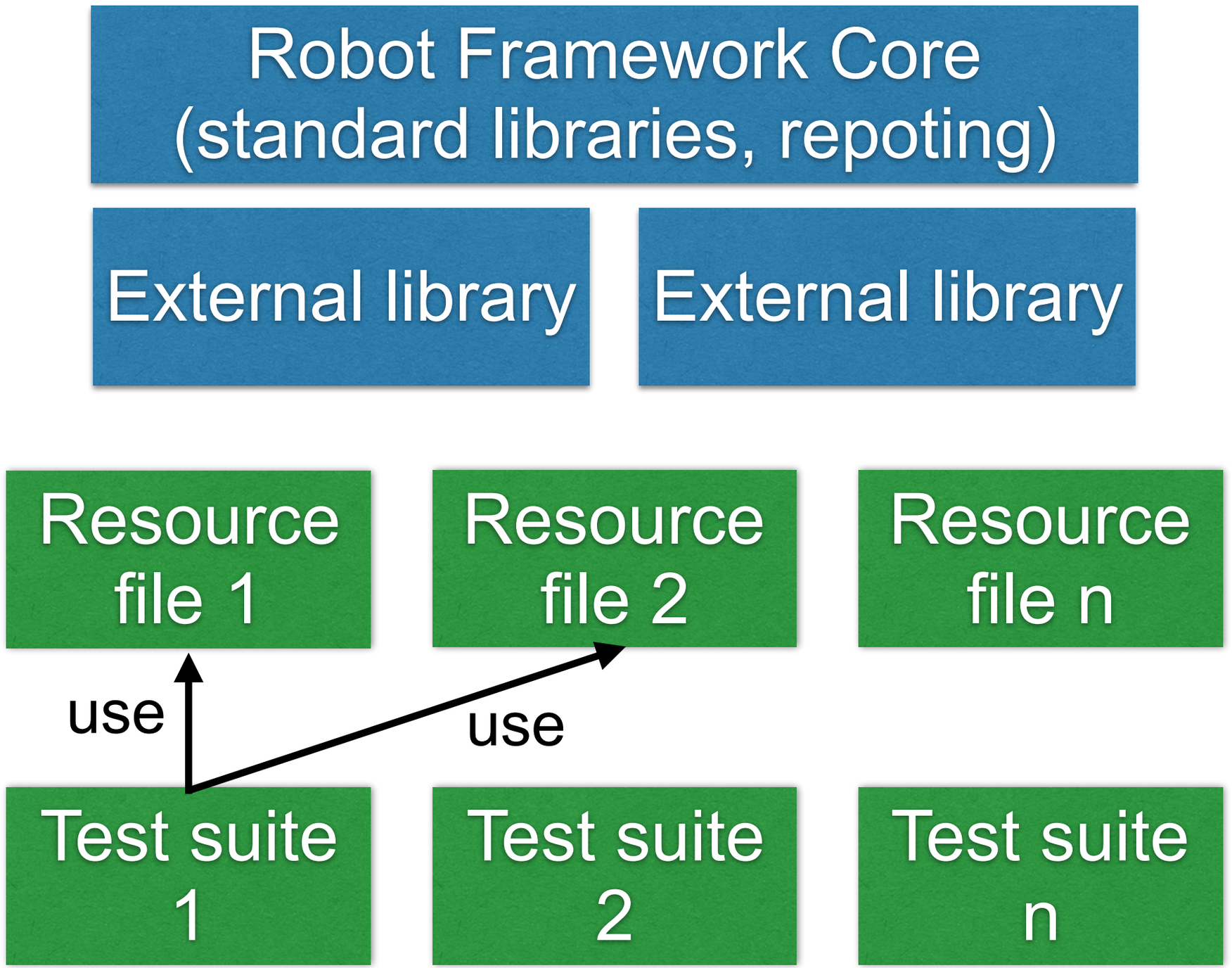
Robot Framework architecture (2)



Robot Framework architecture (3)



Robot Framework architecture (4)



Recommended Libraries

Selenium Library

Web testing library that uses popular Selenium tool internally

HTTP Library (requests)

Library for HTTP level testing using Request internally

REST Instance

Robot Framework test library for HTTP JSON APIs

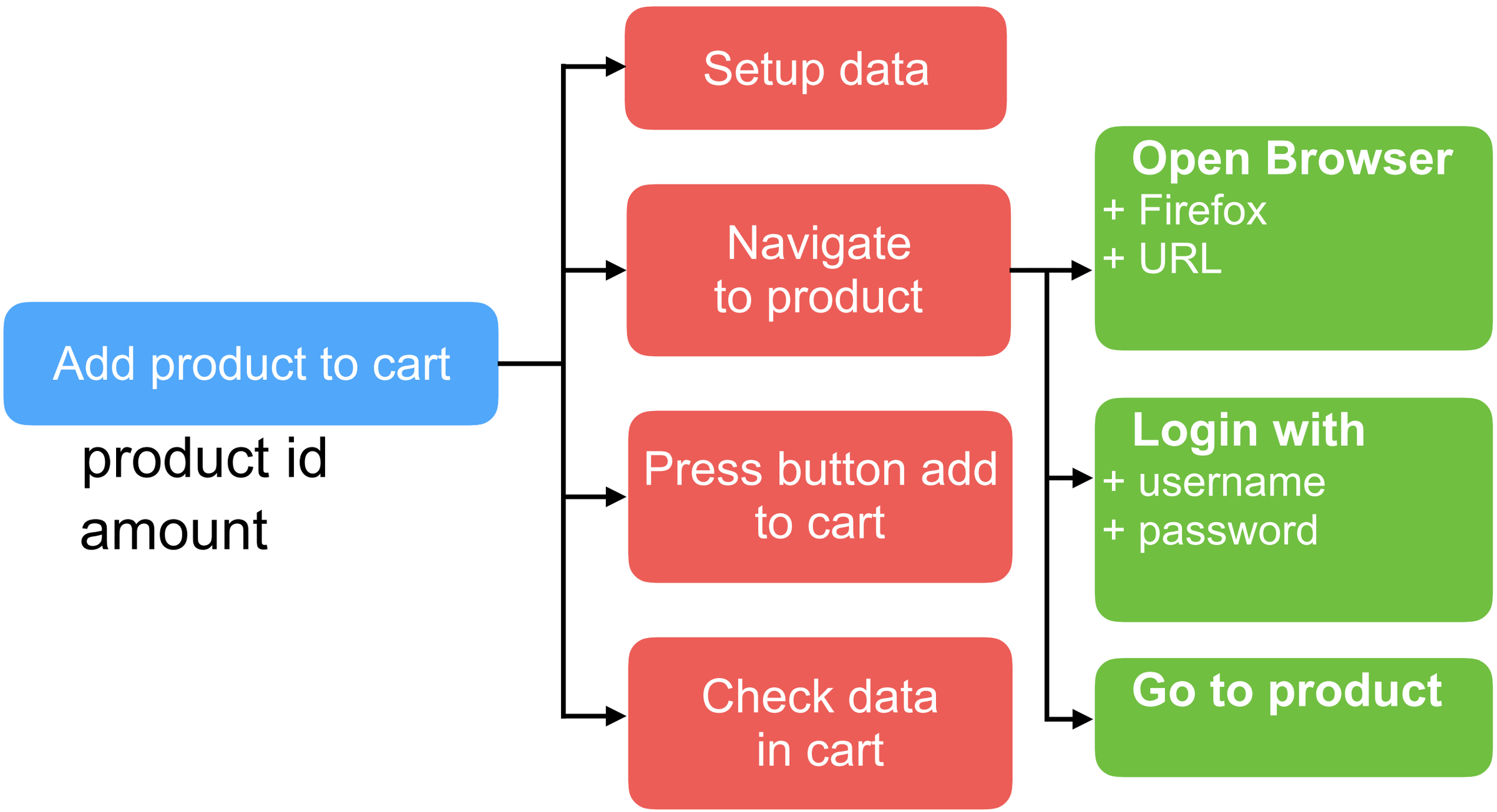
Faker Library

Library for Faker, a fake test data generator

Keyword-driven development

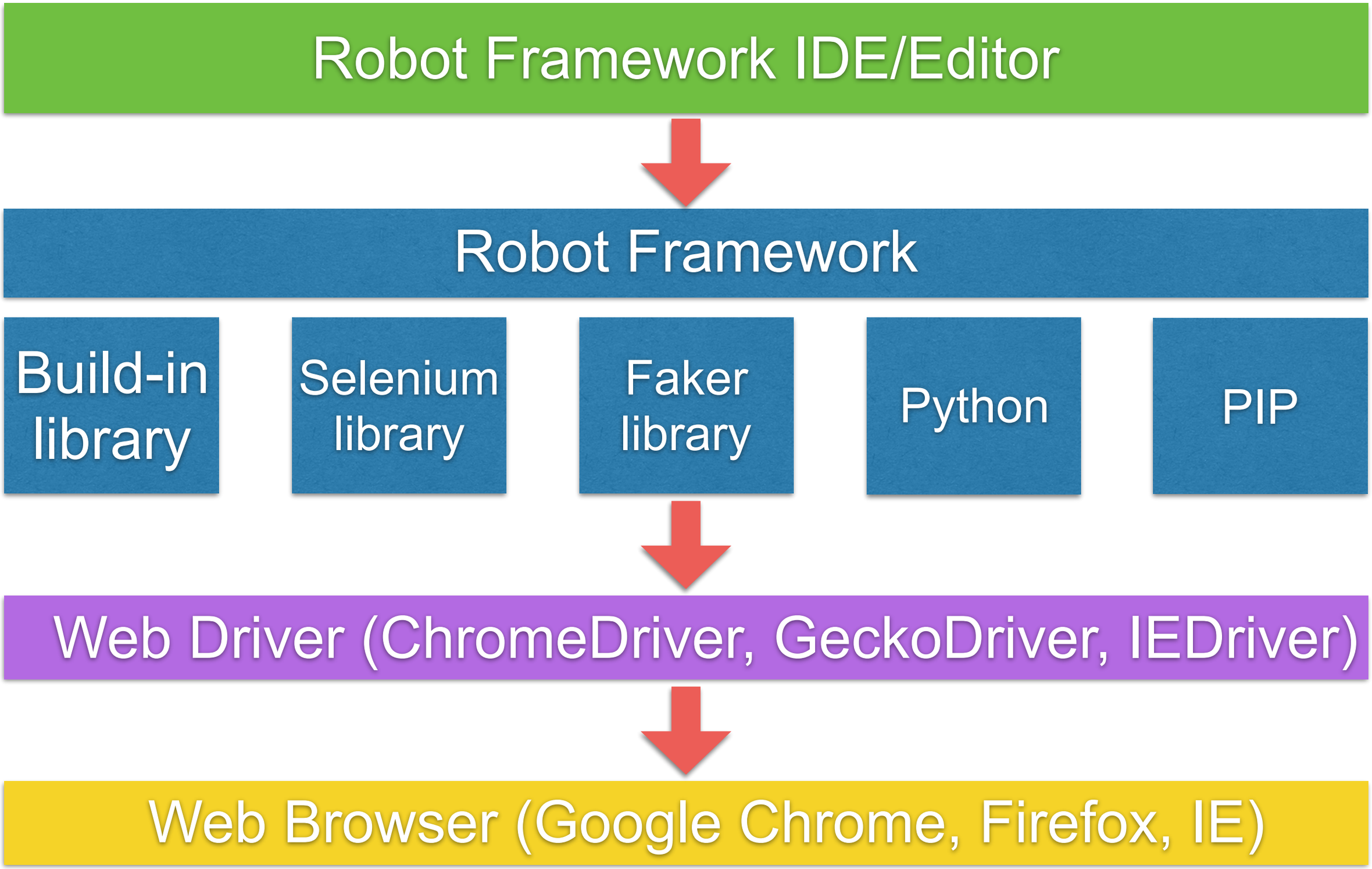
“Add product to cart”

Add product to cart



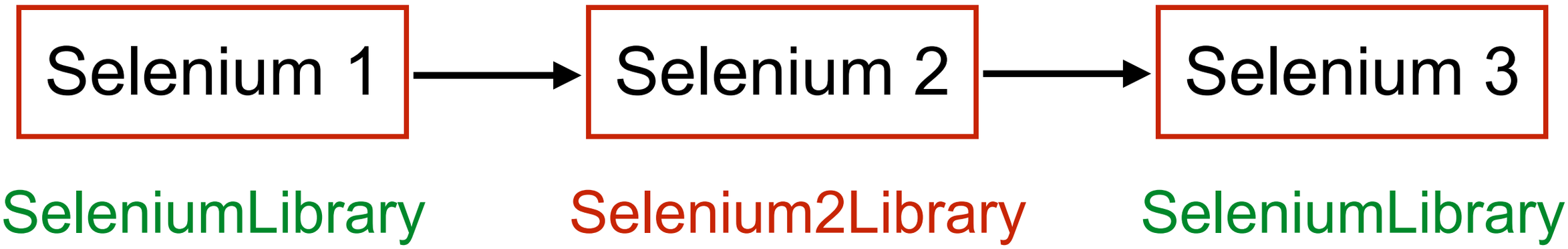
Working with web browser

Working with Web Browser



Install SeleniumLibrary

$pip install robotframework-seleniumlibrary



https://github.com/robotframework/SeleniumLibrary/

SeleniumLibrary keywords

Checkbox Should Be Selected · **C**heckbox Should Not Be Selected · **C**hoose File · **C**lear Element Text · **C**lick Button · **C**lick Element · **C**lick Element At Coordinates · **C**lick Image · **C**lick Link · **C**lose All Browsers · **C**lose Browser · **C**lose Window · **C**over Element · **C**reate Webdriver · **C**urrent Frame Should Contain · **C**urrent Frame Should Not Contain · **D**elete All Cookies · **D**elete Cookie · **D**ouble Click Element · **D**rag And Drop · **D**rag And Drop By Offset · **E**lement Attribute Value Should Be · **E**lement Should Be Disabled · **E**lement Should Be Enabled · **E**lement Should Be Focused · **E**lement Should Be Visible · **E**lement Should Contain · **E**lement Should Not Be Visible · **E**lement Should Not Contain · **E**lement Text Should Be · **E**lement Text Should Not Be · **E**xecute Async Javascript · **E**xecute Javascript · **F**rame Should Contain · **G**et All Links · **G**et Browser Aliases · **G**et Browser Ids · **G**et Cookie · **G**et Cookies · **G**et Element Attribute · **G**et Element Count · **G**et Element Size · **G**et Horizontal Position · **G**et List Items · **G**et Location · **G**et Locations · **G**et Selected List Label · **G**et Selected List Labels · **G**et Selected List Value · **G**et Selected List Values · **G**et Selenium Implicit Wait · **G**et Selenium Speed · **G**et Selenium Timeout · **G**et Session Id · **G**et Source · **G**et Table Cell · **G**et Text · **G**et Title · **G**et Value · **G**et Vertical Position · **G**et WebElement · **G**et WebElements · **G**et Window Handles · **G**et Window Identifiers · **G**et Window Names · **G**et Window Position · **G**et Window Size · **G**et Window Titles · **G**o Back · **G**o To · **H**andle Alert · **I**nput Password · **I**nput Text · **I**nput Text Into Alert · **L**ist Selection Should Be · **L**ist Should Have No Selections · **L**ocation Should Be · **L**ocation Should Contain · **L**ocator Should Match X Times · **L**og Location · **L**og Source · **L**og Title · **M**aximize Browser Window · **M**ouse Down · **M**ouse Down On Image · **M**ouse Down On Link · **M**ouse Out · **M**ouse Over · **M**ouse Up · **O**pen Browser · **O**pen Context Menu · **P**age Should Contain · **P**age Should Contain Button · **P**age Should Contain Checkbox · **P**age Should Contain Element · **P**age Should Contain Image · **P**age Should Contain Link · **P**age Should Contain List · **P**age Should Contain Radio Button · **P**age Should Contain Textfield · **P**age Should Not Contain · **P**age Should Not Contain Button · **P**age Should Not Contain Checkbox · **P**age Should Not Contain Element · **P**age Should Not Contain Image · **P**age Should Not Contain Link · **P**age Should Not Contain List · **P**age Should Not Contain Radio Button · **P**age Should Not Contain Textfield · **P**ress Key · **P**ress Keys · **R**adio Button Should Be Set To · **R**adio Button Should Not Be Selected · **R**egister Keyword To Run On Failure · **R**eload Page · **R**emove Location Strategy · **S**croll Element Into View · **S**elect All From List · **S**elect Checkbox · **S**elect Frame · **S**elect From List By Index · **S**elect From List By Label · **S**elect From List By Value · **S**elect Radio Button · **S**elect Window · **S**et Browser Implicit Wait · **S**et Focus To Element · **S**et Screenshot Directory · **S**et Selenium Implicit Wait · **S**et Selenium Speed · **S**et Selenium Timeout · **S**et Window Position · **S**et Window Size · **S**imulate Event · **S**ubmit Form · **S**witch Browser · **S**witch Window · **T**able Cell Should Contain · **T**able Column Should Contain · **T**able Footer Should Contain · **T**able Header Should Contain · **T**able Row Should Contain · **T**able Should Contain · **T**extarea Should Contain · **T**extarea Value Should Be · **T**extfield Should Contain · **T**extfield Value Should Be · **T**itle Should Be · **U**nselect All From List · **U**nselect Checkbox · **U**nselect Frame · **U**nselect From List By Index · **U**nselect From List By Label · **U**nselect From List By Value · **W**ait For Condition · **W**ait Until Element Contains · **W**ait Until Element Does Not Contain · **W**ait Until Element Is Enabled · **W**ait Until Element Is Not Visible · **W**ait Until Element Is Visible · **W**ait Until Location Contains · **W**ait Until Location Is · **W**ait Until Page Contains · **W**ait Until Page Contains Element · **W**ait Until Page Does Not Contain · **W**ait Until Page Does Not Contain Element

https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html

Open Browser

Browser	Name(s)
Firefox	firefox, ff
Google Chrome	googlechrome, chrome, gc
Headless Firefox	headlessfirefox
Headless Chrome	headlesschrome
Internet Explorer	internetexplorer, ie
Edge	edge
Safari	safari
Opera	opera
Android	android
Iphone	iphone
PhantomJS	phantomjs
HTMLUnit	htmlunit
HTMLUnit with Javascript	htmlunitwithjs

https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html

Locator strategies

Strategy	Match based on	Example
id	Element `id`.	`id:example`
name	`name` attribute.	`name:example`
identifier	Either `id` or `name`.	`identifier:example`
class	Element `class`.	`class:example`
tag	Tag name.	`tag:div`
xpath	XPath expression.	`xpath://div[@id="example"]`
css	CSS selector.	`css:div#example`
dom	DOM expression.	`dom:document.images[5]`
link	Exact text a link has.	`link:The example`
partial link	Partial link text.	`partial link:he ex`
sizzle	Sizzle selector deprecated.	`sizzle:div.example`
jquery	jQuery expression.	`jquery:div.example`
default	Keyword specific default behavior.	`default:example`

https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html

Verify elements in browser

Using keyword that have prefix **“Wait Until”**

Wait Until Element Contains	*locator*, *text*, *timeout=None*, *error=None*	Waits until the element `locator` contains `text`. Fails if `timeout` expires before the text appears. See the *Timeouts* sect section for details about the locator syntax. `error` can be used to override the default error message.
Wait Until Element Does Not Contain	*locator*, *text*, *timeout=None*, *error=None*	Waits until the element `locator` does not contain `text`. Fails if `timeout` expires before the text disappears. See the *Timeouts* s section for details about the locator syntax. `error` can be used to override the default error message.
Wait Until Element Is Enabled	*locator*, *timeout=None*, *error=None*	Waits until the element `locator` is enabled. Element is considered enabled if it is not disabled nor read-only. Fails if `timeout` expires before the element is enabled. See the *Timeou elements* section for details about the locator syntax. `error` can be used to override the default error message. Considering read-only elements to be disabled is a new feature in Seleniu
Wait Until Element Is Not Visible	*locator*, *timeout=None*, *error=None*	Waits until the element `locator` is not visible. Fails if `timeout` expires before the element is not visible. See the *Time elements* section for details about the locator syntax.

https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html

First test case

Create file => **google_success.robot**

```
*** Settings ***

*** Variables ***

*** Test Cases ***

*** Keywords ***
```

Run test and see result

$robot google_success.robot

Working with Selenium

```
*** Settings ***
Library     SeleniumLibrary

*** Variables ***
${URL}   http://www.google.com

*** Test Cases ***
ค้นหาคำว่าเหล็กไหล
    [Tags]  done
    เข้าไปยังหน้าค้นหาของ google
    ค้นหาคำว่า "เหล็กไหล"
    จะต้องเจอ "เหล็กไหล" นะ

*** Keywords ***
ค้นหาคำว่า "${keyword}"
    Input Text      name:q   ${keyword}
    Press Keys      name:q   RETURN
```

Setup web browser driver

Web Browser	Driver
Google Chrome	ChromeDriver
FireFox	GeckoDriver
Internet Explorer	InternetExplorerDriver
Microsoft Edge	EdgeDriver

eg. ChromeDriver

$set WEB_DRIVER=<path of driver>
$set PATH=.;%WEB_DRIVER%;%PATH%

https://chromedriver.chromium.org/downloads

Reporting

Open file report.html

Comments Api Report

Generated
20190925 16:13:40 UTC+07:00
13 days 20 hours ago

Summary Information

Status:	All tests passed
Start Time:	20190925 16:13:40.553
End Time:	20190925 16:13:40.961
Elapsed Time:	00:00:00.408
Log File:	log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	1	1	0	00:00:00	
All Tests	1				

Statistics by Tag	Total
No Tags	

Statistics by Suite	Total
Comments Api	1

Test Details

Totals | Tags | Suites | Search

Type: Critical Tests / All Tests

Comments Api Log

Generated
20190925 16:13:40 UTC+07:00
13 days 20 hours ago

Test Statistics

Total Statistics	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	1	1	0	00:00:00	
All Tests	1	1	0	00:00:00	

Statistics by Tag	Total	Pass	Fail	Elapsed	Pass / Fail
No Tags					

Statistics by Suite	Total	Pass	Fail	Elapsed	Pass / Fail
Comments Api	1	1	0	00:00:00	

Test Execution Log

SUITE **Comments Api**
Full Name: Comments Api
Source: /Users/somkiat/data/slide/robotframework/aycap-20190924/comments_api.robot
Start / End / Elapsed: 20190925 16:13:40.553 / 20190925 16:13:40.961 / 00:00:00.408
Status: 1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed

TEST **Flow 1**
Full Name: Comments Api.Flow 1
Start / End / Elapsed: 20190925 16:13:40.750 / 20190925 16:13:40.960 / 00:00:00.210
Status: PASS (critical)

KEYWORD ${email} = **Get Comments**
KEYWORD BuiltIn. **Log To Console** ${email}

Variables

Define global variables in test suites

```
*** Variables ***
${URL}  http://www.google.com
```

Variables

Set value of variables from command line

```
*** Variables ***
${URL}  http://www.google.com
```

$robot **-v URL:<value>** google_success.robot

Build-ins variables

Usage	Description
${SPACE}	\ \
“${SPACE}”	“ ”
${SPACE * 5}	\ \ \ \ \
“${SPACE * 5}”	“ \ \ \ \ ”
${EMPTY}	\

Tagging test cases

Grouping your test cases

```
*** Test Cases ***
ค้นหาคำว่า robot
    [Tags]  done
    ค้นหาคำว่า    robot
    ต้องเจอคำว่า robot

ค้นหาคำว่า thai
    [Tags]  testing
    ค้นหาคำว่า   thai
    ต้องเจอคำว่า robot
```

Run tests with specified tags

$robot **-i testing** google_success.robot
$robot **-e done** google_success.robot

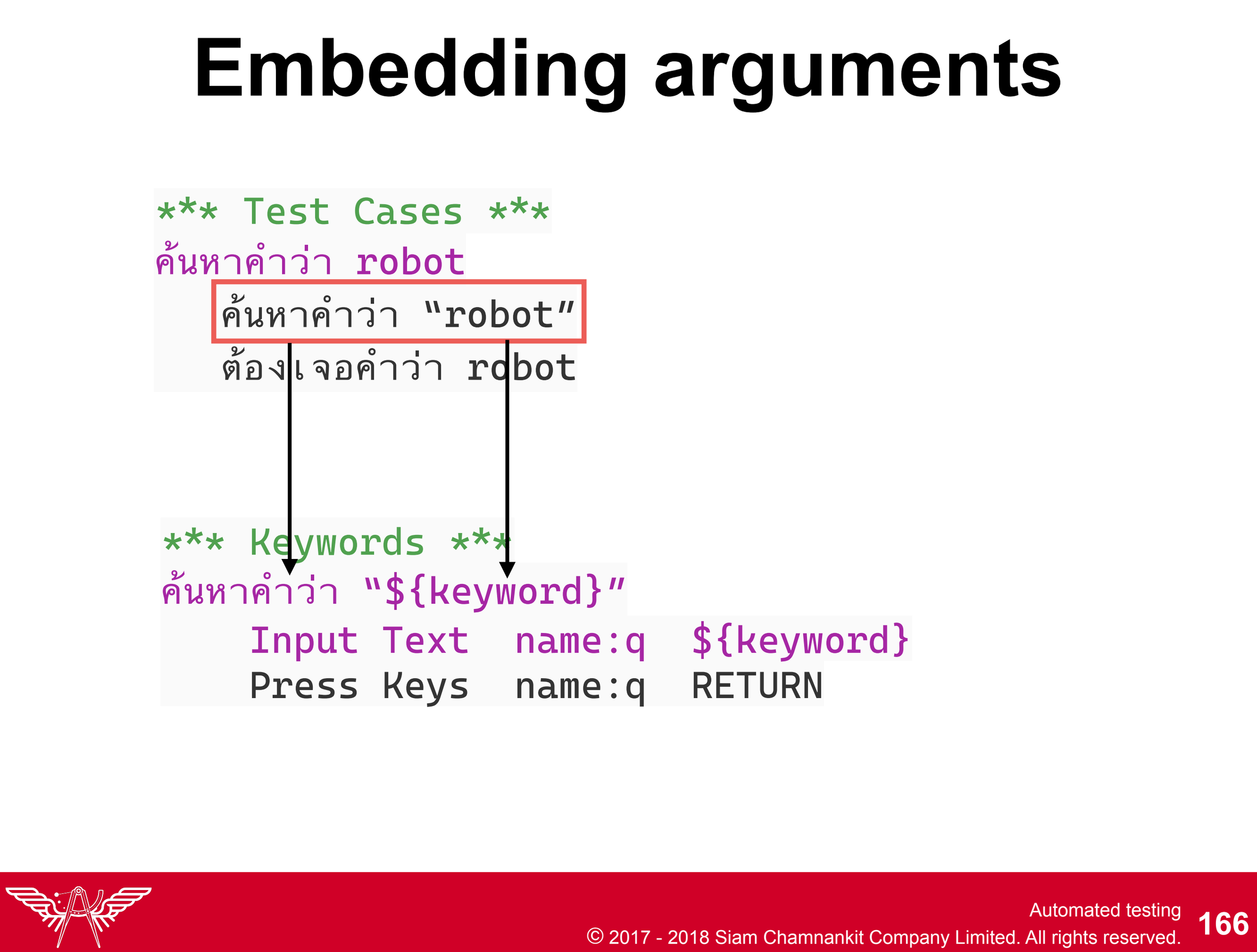
-i = Include, -e = Exclude

User keyword arguments

```
*** Test Cases ***
ค้นหาคำว่า robot
    ค้นหาคำว่า    robot
    ต้องเจอคำว่า robot

*** Keywords ***
ค้นหาคำว่า
    [Arguments]  ${keyword}
    Input Text   name:q   ${keyword}
    Press Keys   name:q   RETURN
```

Embedding arguments

```
*** Test Cases ***
ค้นหาคำว่า robot
    ค้นหาคำว่า "robot"
    ต้องเจอคำว่า robot

*** Keywords ***
ค้นหาคำว่า "${keyword}"
    Input Text  name:q  ${keyword}
    Press Keys  name:q  RETURN
```



Improve your test with

Test Life Cycle

Test Life Cycle

Suite

Suite Setup

Test Setup
Test case 1
Test Teardown

Test Setup
Test case 2
Test Teardown

Test Setup
Test case 3
Test Teardown

Suite Teardown

Test Life Cycle

```
*** Settings ***
Suite Setup       เข้า web google
Test Setup        เพิ่มข้อมูลการทดสอบ
Test Teardown     กลับไปยังหน้าแรก
Suite Teardown    Close Browser
```

Improve your test with **Test Template**

Test template

One template for all test cases

```
*** Settings ***
Test Template    Search ${keyword} should be found ${expected result}

*** Test Cases ***
Case 01          robot        robot
Case 02          thai         thai
```

```
*** Keywords ***
Search ${keyword} should be found ${expected result}
    ค้นหาคำว่า    ${keyword}
    ต้องเจอคำว่า robot
```

Test template

Template per test case

```
*** Test Cases ***
Search from google
    [Template]  Flow success to search from google
      robot       robot
      thai        thai
```

```
*** Keywords ***
Flow success to search from google
    [Arguments]  ${keyword}   ${expected result}
    Search ${keyword} shoud be found ${expected result}
```

Reuse keywords/variables with **Resource**

Using resource

```
*** Settings ***
Resource          resources/sample.robot
Suite Setup       sample.เข้า web google
Suite Teardown    Close Browser
```

1. Create folder = resourses
2. Create file = resourses/sample.robot

```
*** Settings ***
Library           SeleniumLibrary
Library           String

*** Variables ***
${URL}  http://www.google.com

*** Keywords ***
กลับไปยังหน้าแรก
    Go To  ${URL}
```

Page Object Pattern

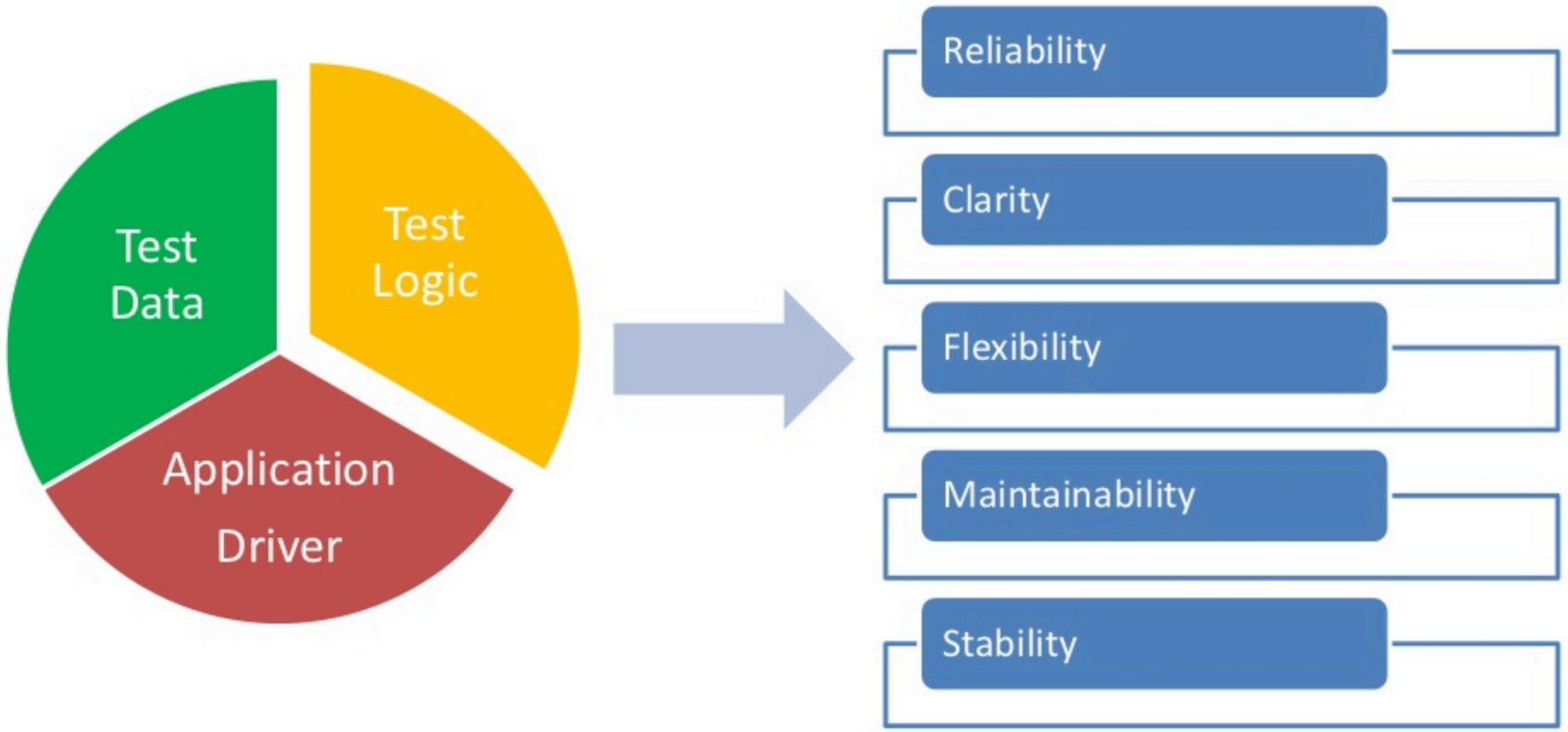
Problems

Fragile tests
Bad maintainability
Hard to develop new tests
Hard to understand
Test code duplication
High cost to maintenance tests

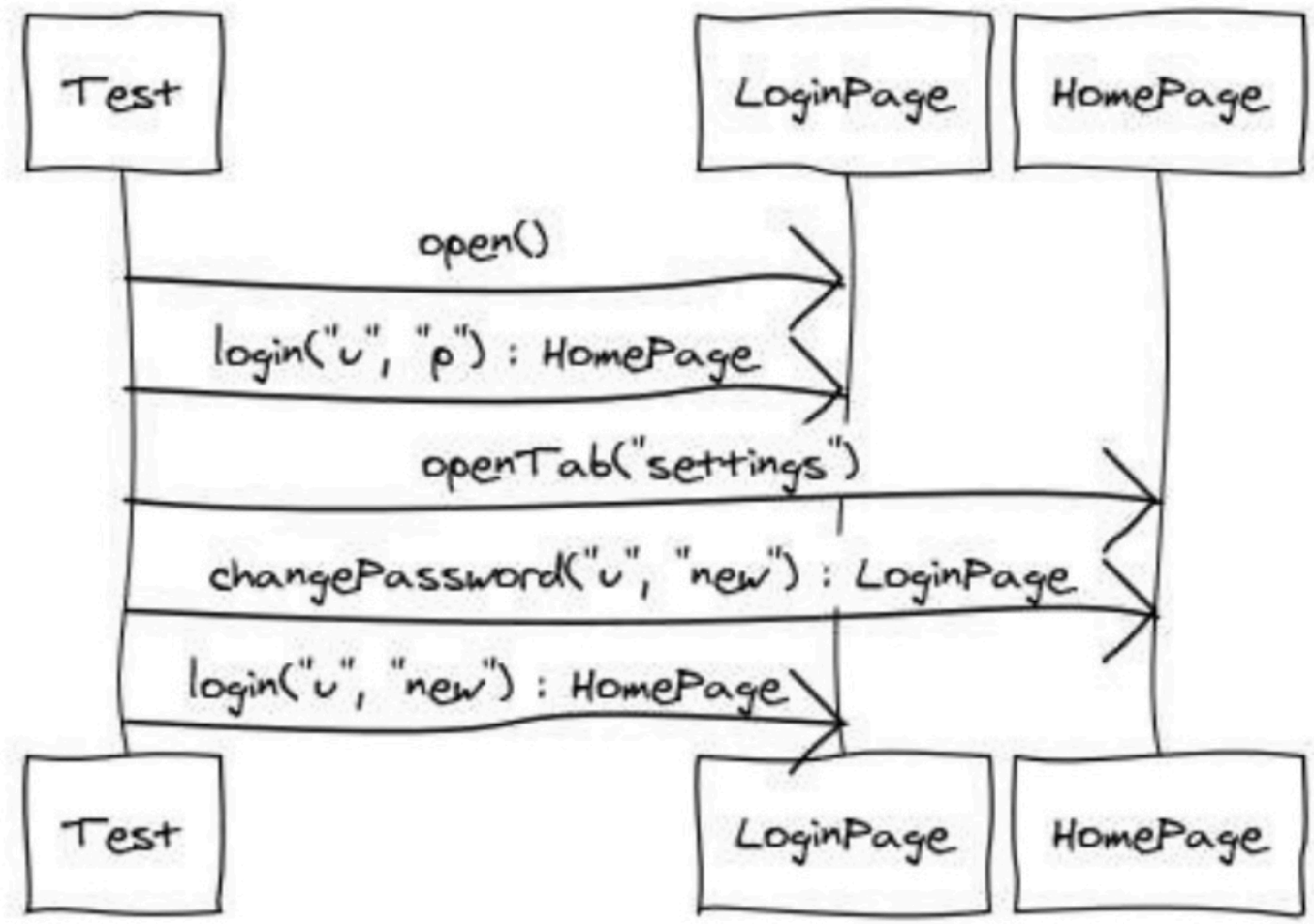
Primary goal ?

To enable reliable stable tests

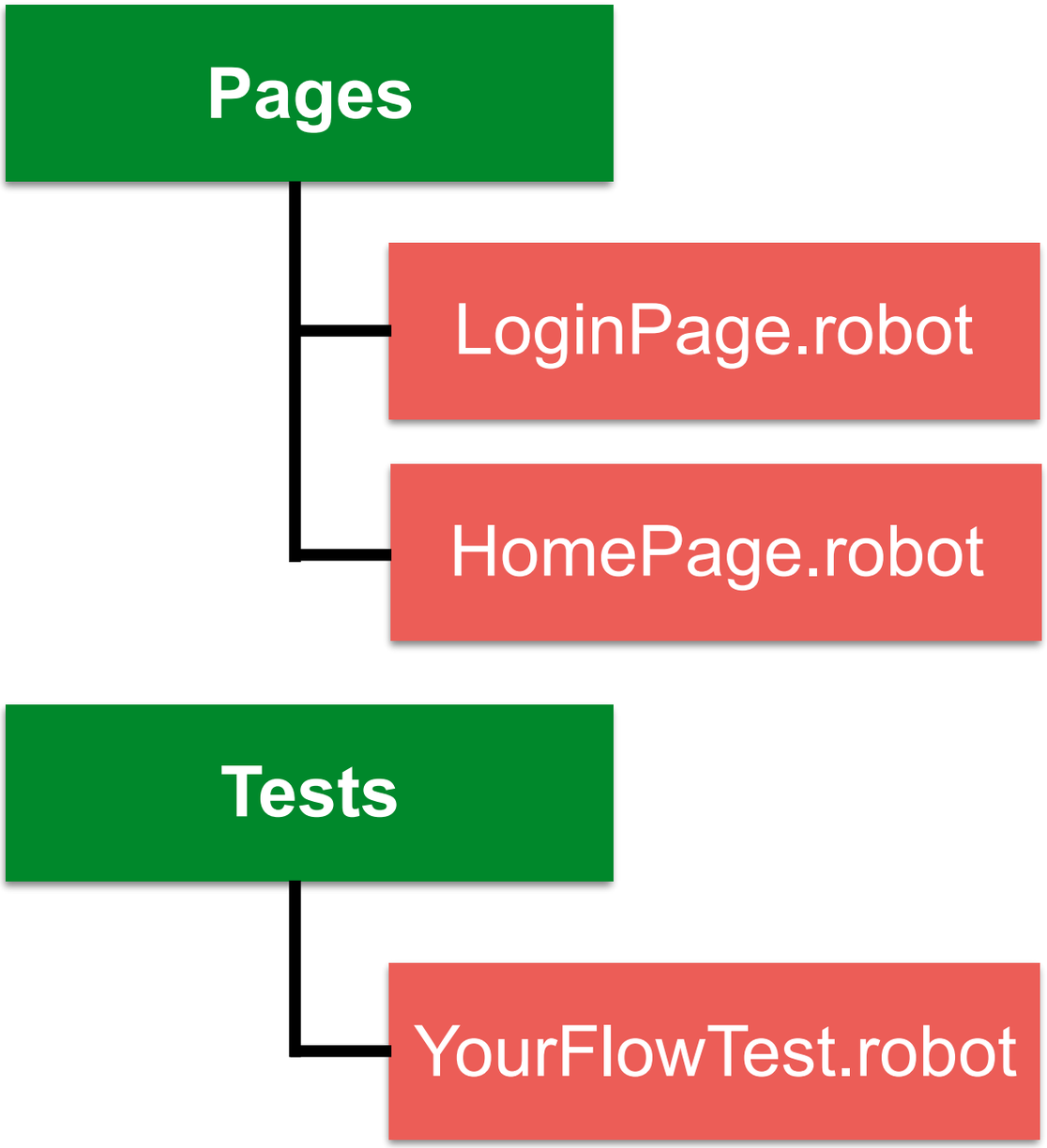
We need to separate all parts as much as possible



Test architecture with Page object



Test architecture with Page object



Tips and Tricks

Basic page with browser details
Expose only what is allowed to do on page
Don’t use browser details in tests
Use site map to prepare Page object

Try to improve …

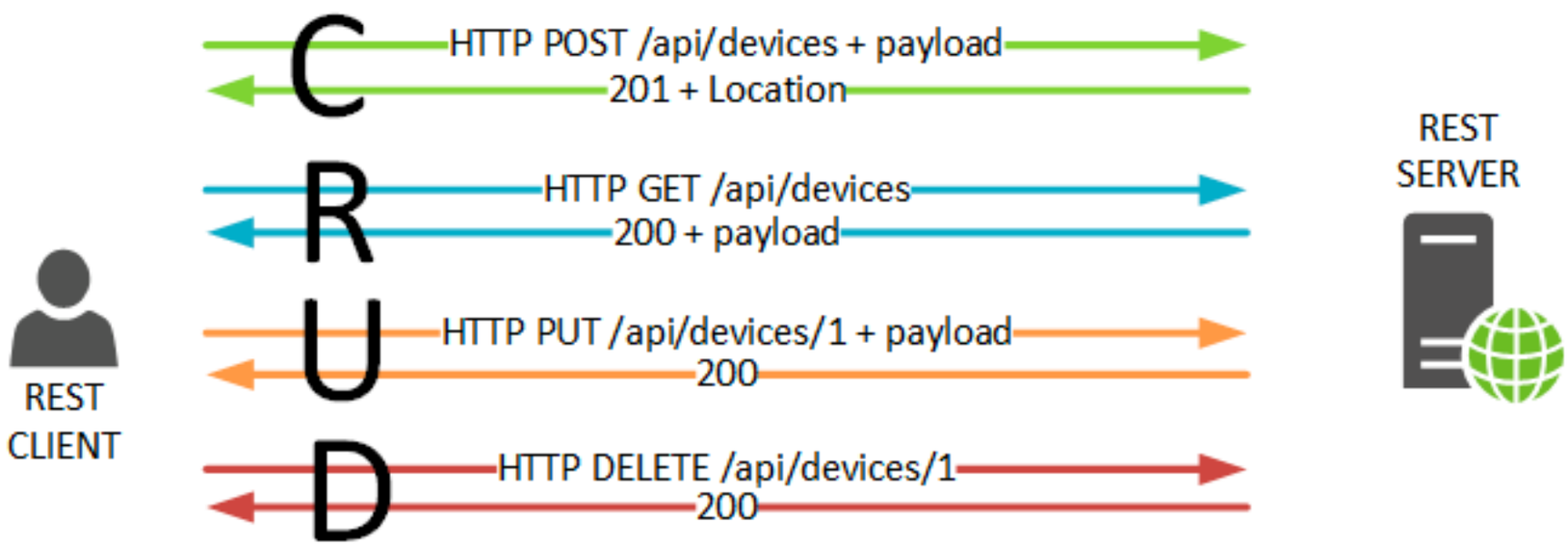
Reliable tests
Reusable test code
Separation of concepts
Expressive UI structure
Short and clear tests
Tests look more like acceptance tests
Tests are understand by non-technical people

API Testing

Install RequestsLibrary

$pip install requests

$pip install robotframework-requests



https://github.com/bulkan/robotframework-requests/

RequestsLibrary

Introduction

`RequestsLibrary` is a Robot Framework test library that uses the Requests HTTP client.

Here is an example testcase

*** Settings ***				
Library	Collections			
Library	RequestsLibrary			
*** Test Cases ***				
Get Requests				
	Create Session	github	http://api.github.com	
	Create Session	google	http://www.google.com	
	${resp}=	Get Request	google	/
	Should Be Equal As Strings	${resp.status_code}	200	
	${resp}=	Get Request	github	/users/bulkan
	Should Be Equal As Strings	${resp.status_code}	200	
	Dictionary Should Contain Value	${resp.json()}	Bulkan Savun Evcimen	

Shortcuts

Create Digest Session · **C**reate Ntlm Session · **C**reate Session · **D**elete · **D**elete All Sessions · **D**elete Request · **G**et · **G**et Request · **H**ead · **H**ead Request · **O**ptions · **O**ptions Request · **P**atch · **P**atch Request · **P**ost · **P**ost Request · **P**ut · **P**ut Request · **T**o Json

https://bulkan.github.io/robotframework-requests/doc/RequestsLibrary.html

RequestsLibrary keywords

Shortcuts

Create Client Cert Session · **C**reate Custom Session · **C**reate Digest Session · **C**reate Ntlm Session · **C**reate Session · **D**elete All Sessions · **D**elete Request · **G**et Request · **H**ead Request · **O**ptions Request · **P**atch Request · **P**ost Request · **P**ut Request · **S**ession Exists · **T**o Json · **U**pdate Session

https://bulkan.github.io/robotframework-requests/doc/RequestsLibrary.html

Create tests

Connect to service and check status code

```
*** Settings ***
Library  RequestsLibrary

*** Test Case ***
Success with /users
    Create Session   jph  https://jsonplaceholder.typicode.com
    ${response}=  Get Request  jph  /users
    Should Be Equal   ${response.status_code}  ${200}
```

Create tests

Check size of result from service

```
*** Test Case ***
Success with /users
    Create Session   jph  https://jsonplaceholder.typicode.com
    ${response}=  Get Request  jph  /users
    Should Be Equal   ${response.status_code}  ${200}

    ${json}=  Set Variable  ${response.json()}
    Length Should Be   ${json}  10
```

Create tests

Check first user data from result

```
*** Test Case ***
Success with /users
    Create Session   jph  https://jsonplaceholder.typicode.com
    ${response}=  Get Request  jph  /users
    Should Be Equal   ${response.status_code}  ${200}
    ${json}=  Set Variable  ${response.json()}
    Length Should Be   ${json}  10

    Should Be Equal   ${json[0]["name"]}  Leanne Graham
    Should Be Equal   ${json[0]["username"]}  Bret
```

Working with HTTP POST

```
Add product id=1000 to empty basket
  Create Session    baskets    http://localhost:8882
  &{headers}=  Create Dictionary  Content-Type=application/json
  &{data}=  Create Dictionary
  ...  product_id=${1000}
  ...  product_name=Adidas
  ...  product_price=${1500}
  ...  product_image=http://xxx.jpg
  ...  quantity=${1}
  ${response}=  Post Request  baskets  /baskets
  ...  data=${data}  headers=${headers}
  Should Be Equal As Strings  ${response.status_code}  200
```

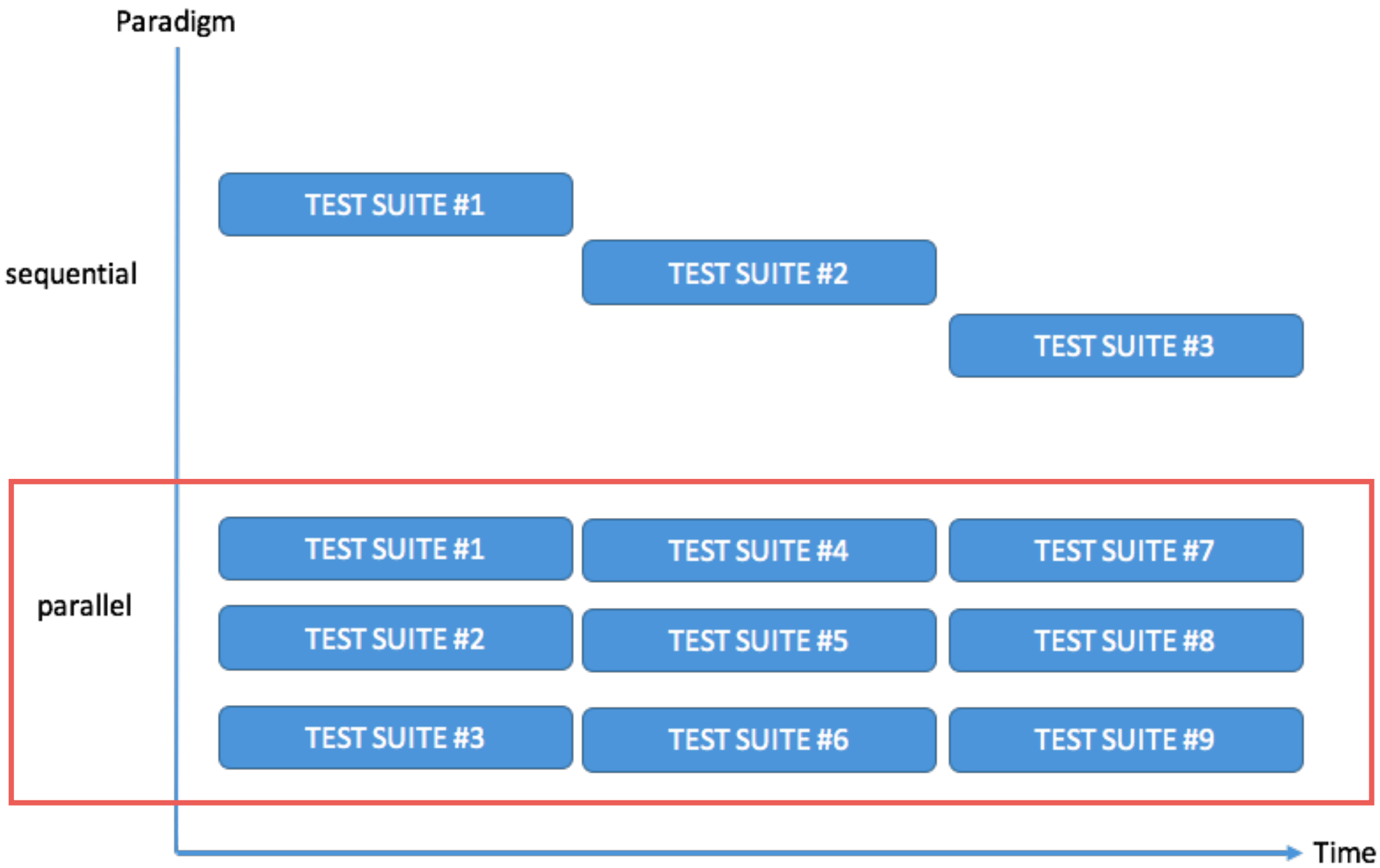
Parallel testing with pabot

Problems

More test cases more problems

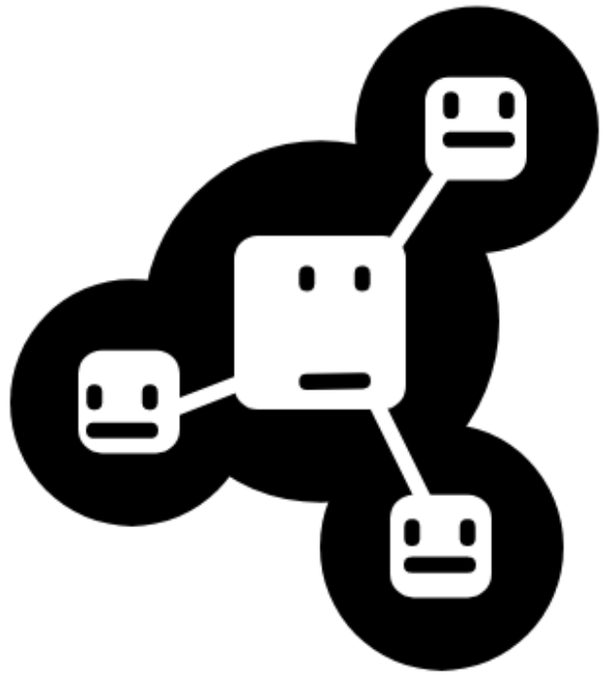
More test cases more testing time

Parallel testing



Install Pabot

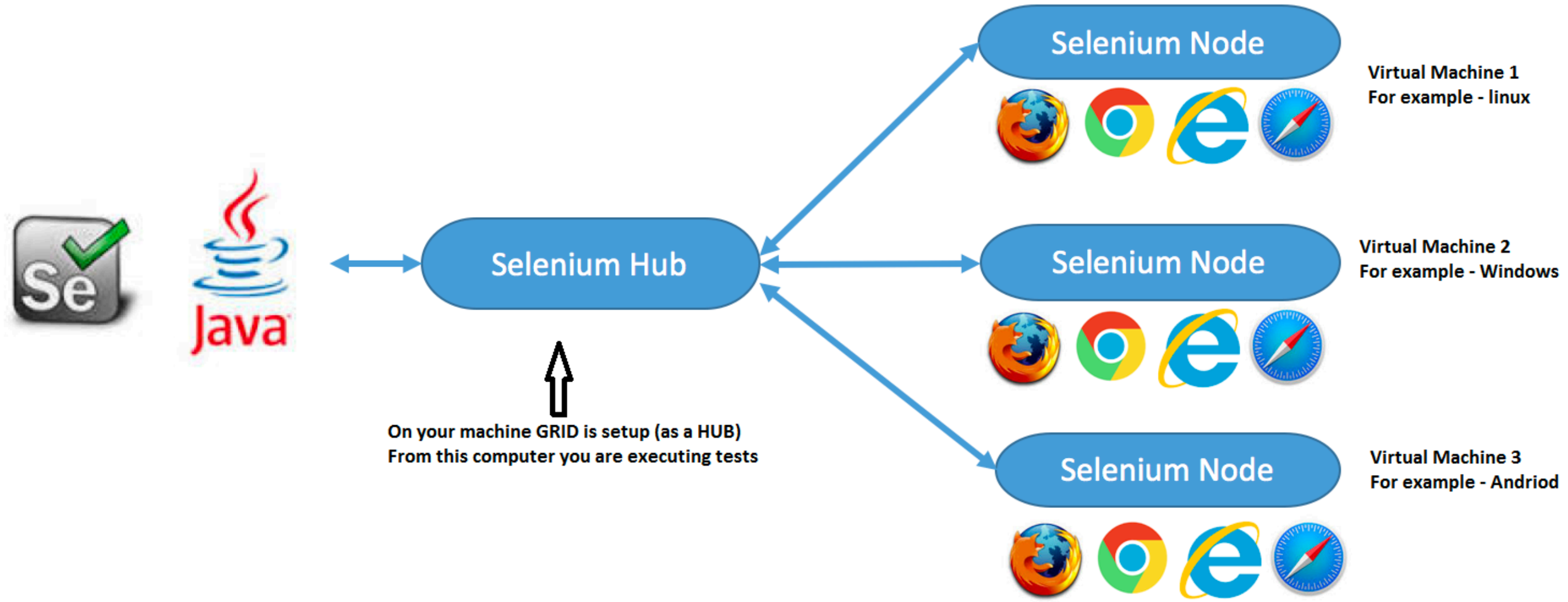
$pip install robotframework-pabot
$pabot



https://pabot.org/

Scaling UI testing with **Selenium Grid**

Selenium Grid



https://github.com/SeleniumHQ/selenium/wiki/Grid2

Using from Robot Framework

```
Open Browser   http://localhost:8080/demo/
...  browser=chrome
...  remote_url=http://localhost:4444/wd/hub
...  desired_capabilities=browserName:chrome
```

Continuous Testing

https://jenkins.io/

1. Create new job

Enter an item name

Fill in job name

» *Required field*

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

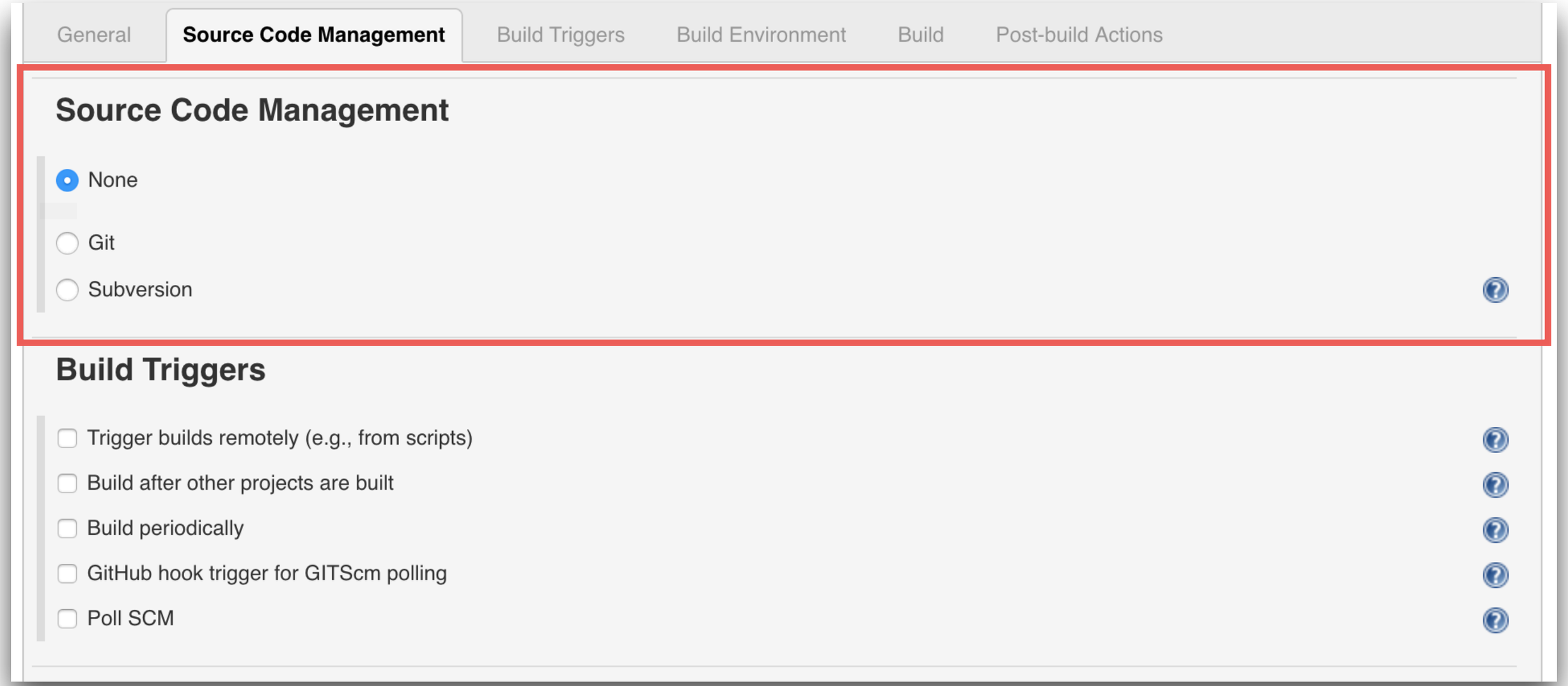
Folder
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

GitHub Organization
Scans a GitHub organization (or user account) for all repositories matching some defined markers.

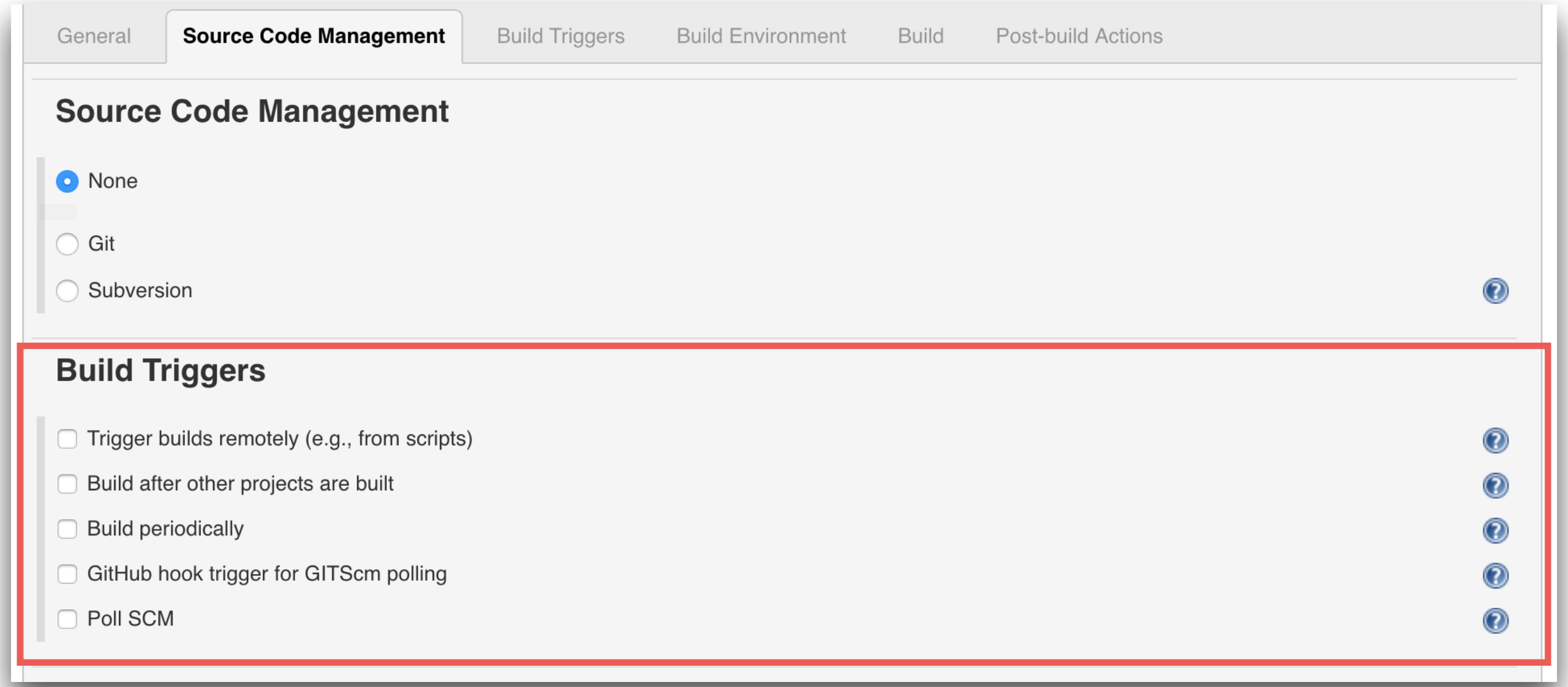
Multibranch Pipeline
Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK

2. Source code management



3. Build trigger



4. Build trigger with poll SCM

Poll SCM

Schedule

* * * * *

No schedules so will only run due to SCM changes if triggered by a post-commit hook

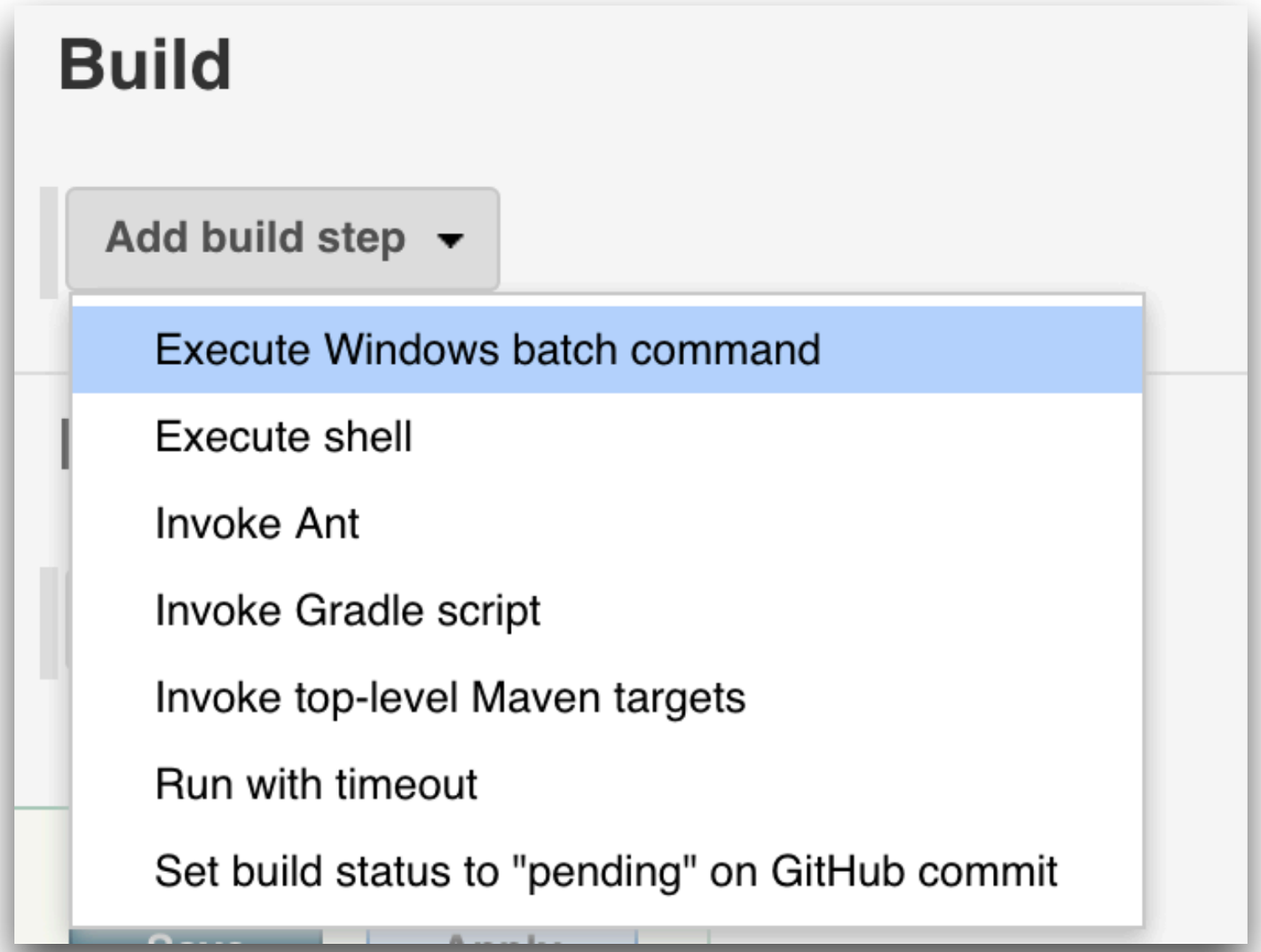
This field follows the syntax of cron (with minor differences). Specifically, each line consists of 5 fields separated by TAB or whitespace:
`MINUTE HOUR DOM MONTH DOW`

MINUTE	Minutes within the hour (0–59)
HOUR	The hour of the day (0–23)
DOM	The day of the month (1–31)
MONTH	The month (1–12)
DOW	The day of the week (0–7) where 0 and 7 are Sunday.

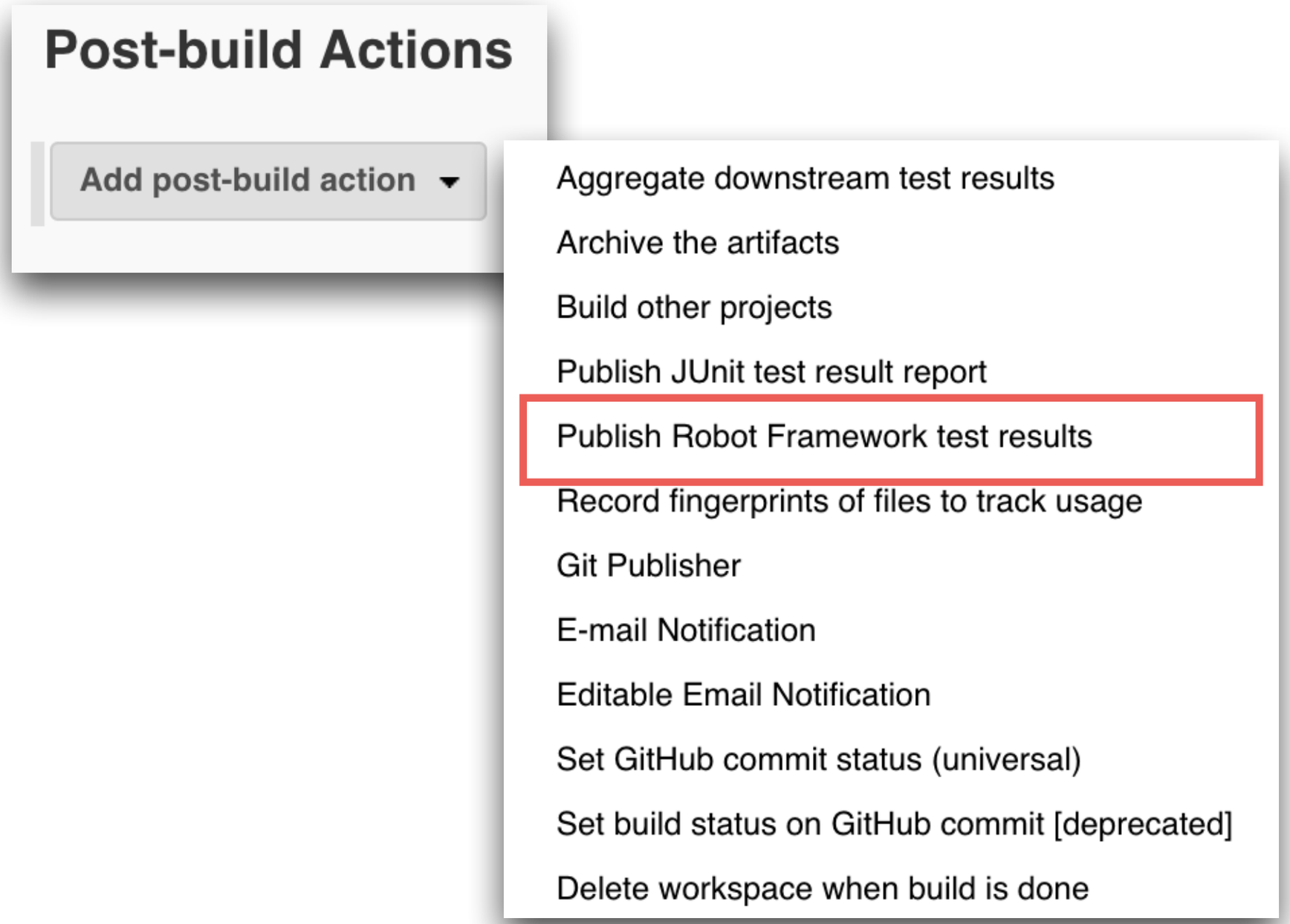
To specify multiple values for one field, the following operators are available. In the order of precedence,

- `*` specifies all valid values
- `M-N` specifies a range of values
- `M-N/X` or `*/X` steps by intervals of X through the specified range or whole valid range
- `A,B,...,Z` enumerates multiple values

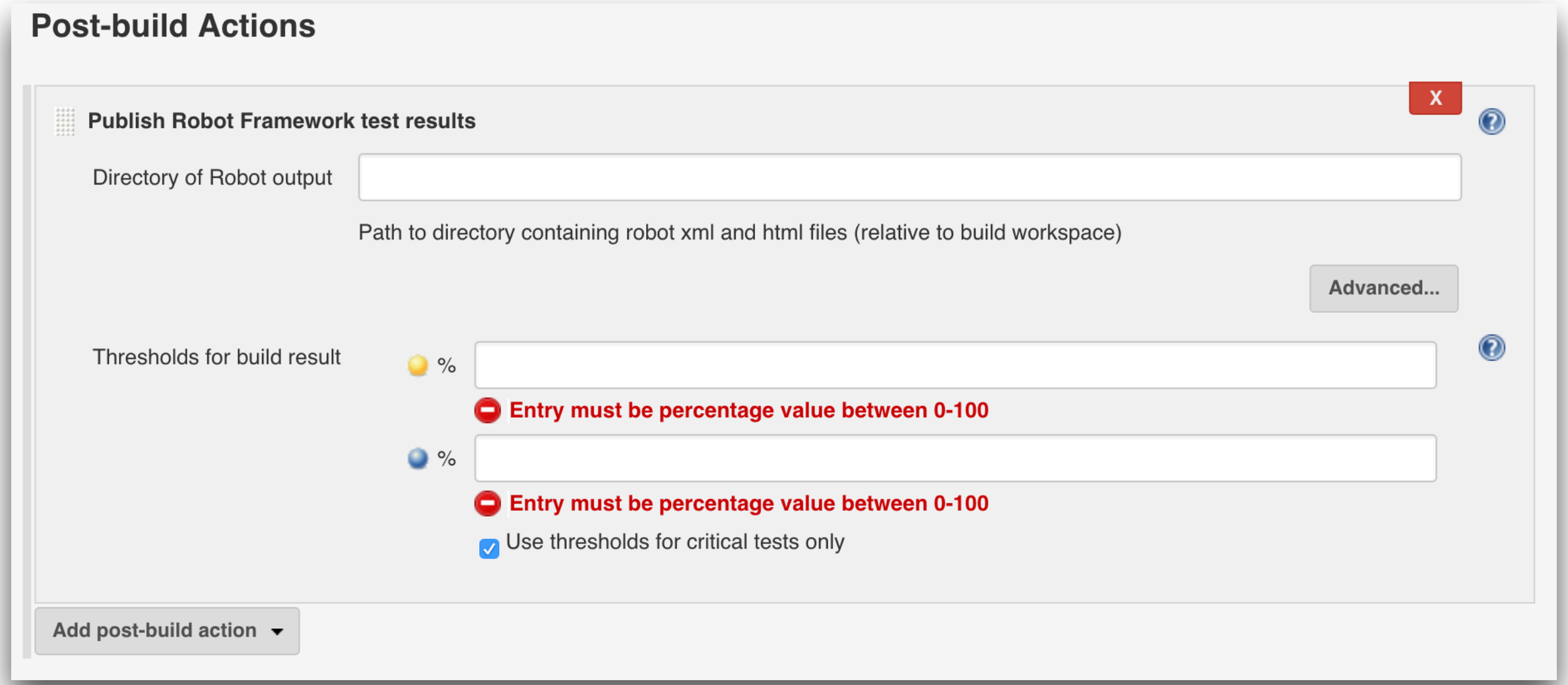
5. Build your job



6. Post build actions



7. Add post build with Robot framework



8. Install Robot framework plugin

People
Build History
Manage Jenkins
My Views
Credentials
New View

1

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

2

Manage Jenkins

Configure System
Configure global settings and paths.

Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Configure Credentials
Configure the credential providers and types

Global Tool Configuration
Configure tools, their locations and automatic installers.

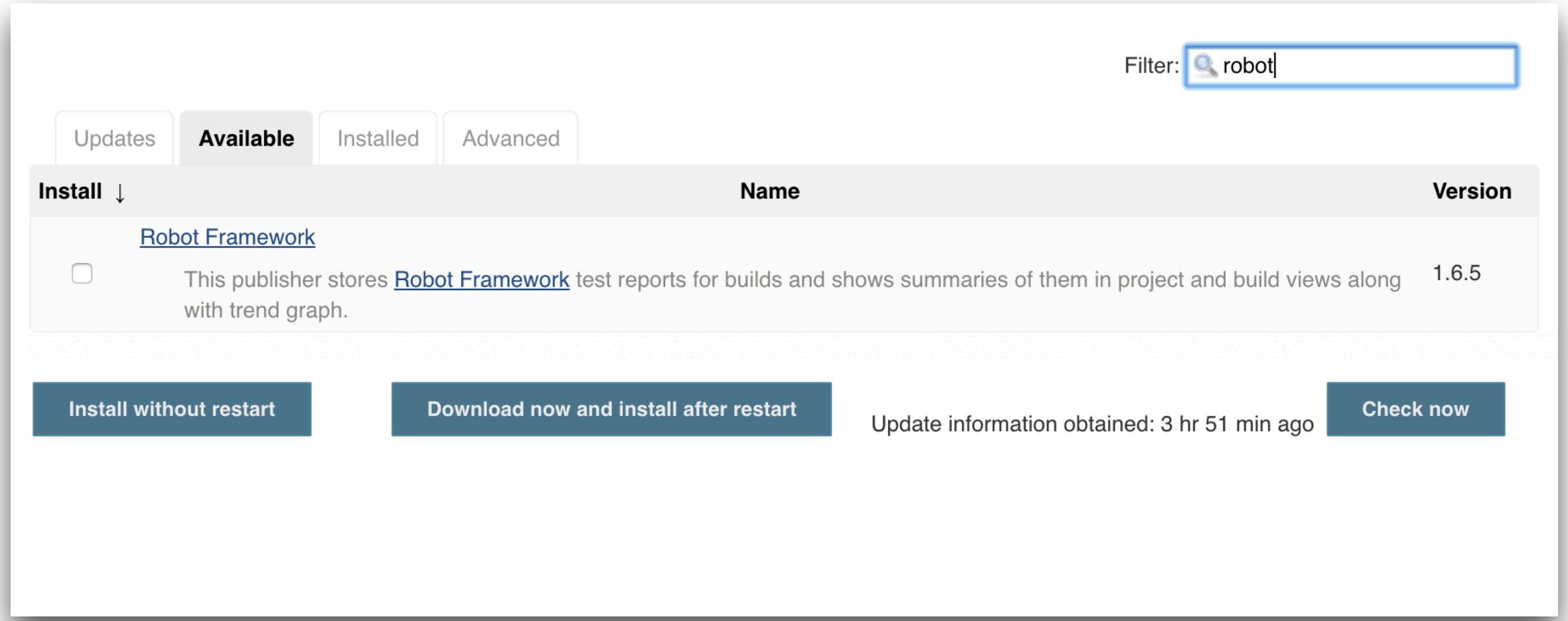
Reload Configuration from Disk
Discard all the loaded data in memory and reload everything from file system. Useful when you modif
disk.

Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

System Information
Displays various environmental information to assist trouble-shooting.

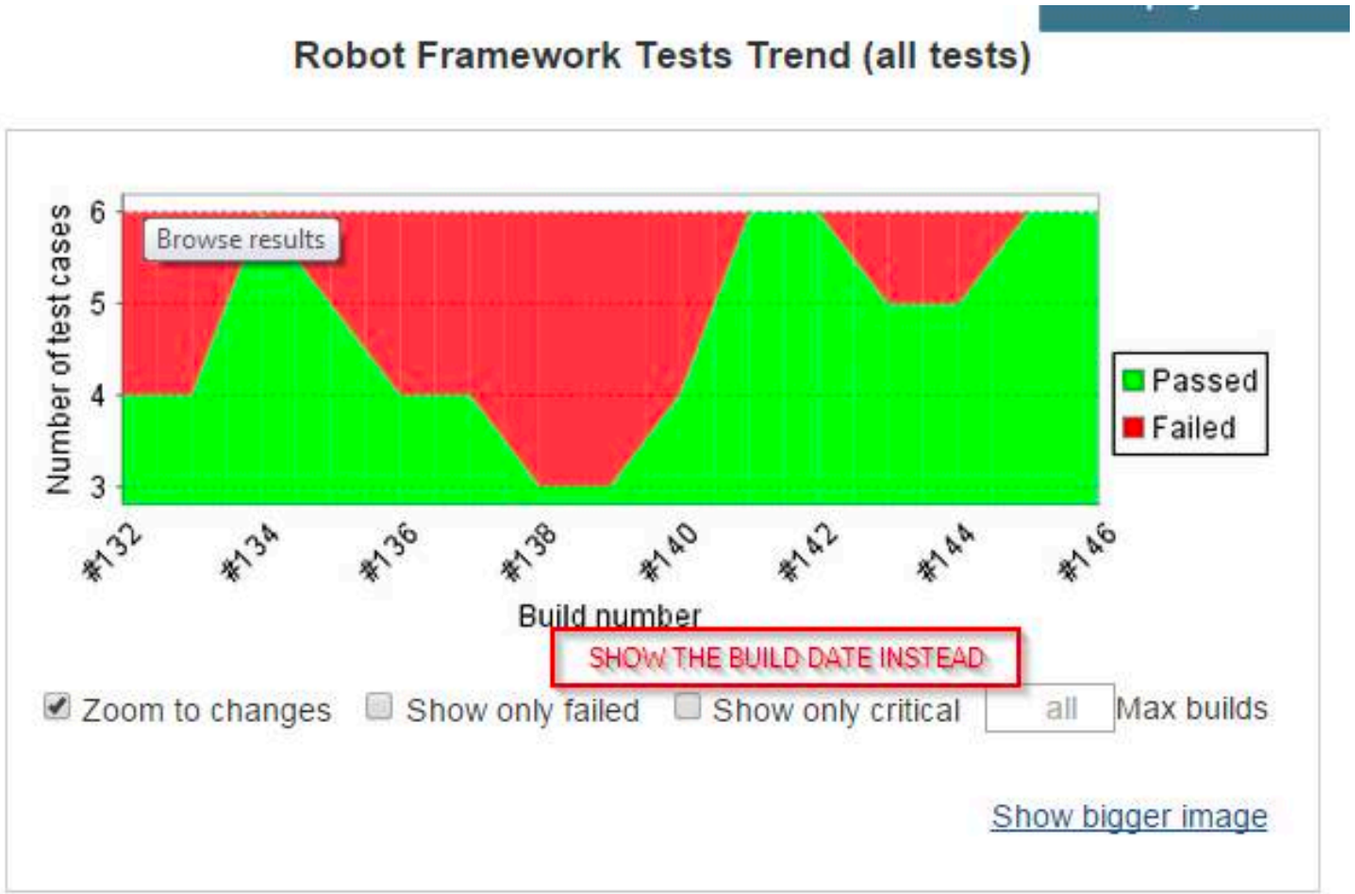
System Log
System log captures output from `java.util.logging` output related to Jenkins.

8. Install Robot framework plugin



https://plugins.jenkins.io/robot

Robot framework report



https://plugins.jenkins.io/robot