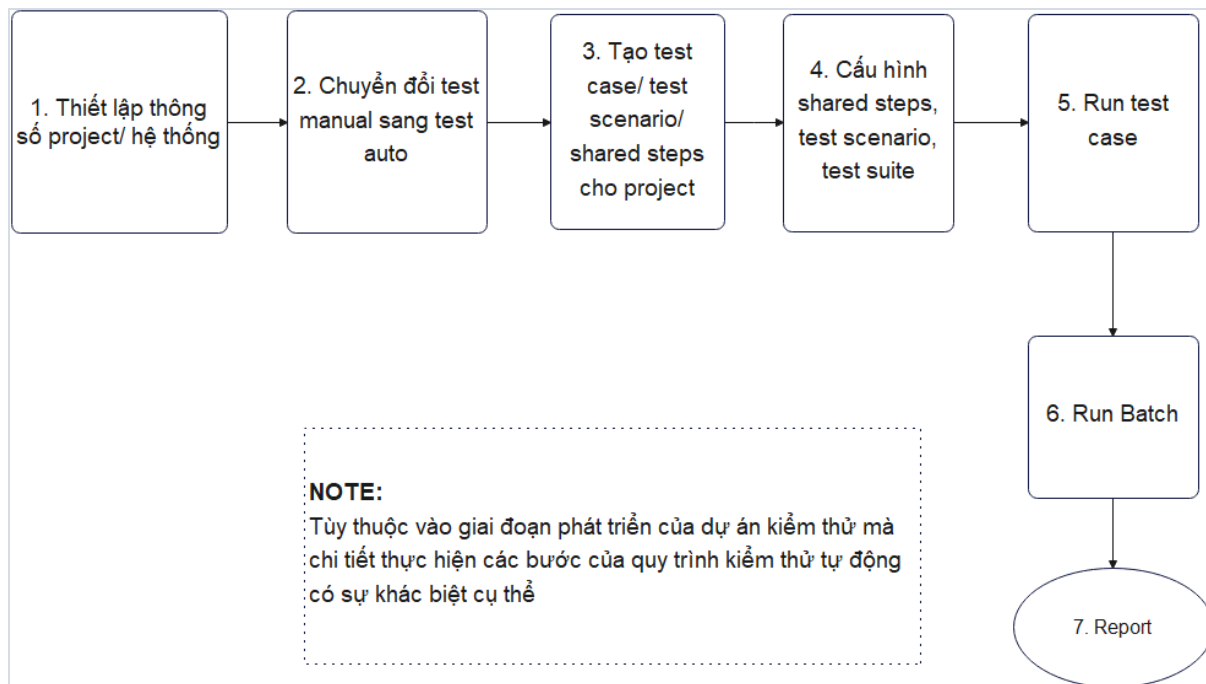
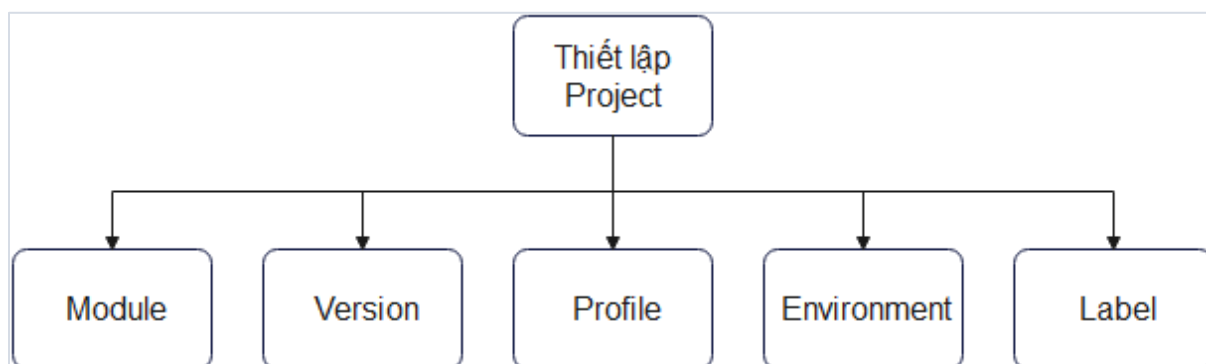


Quy trình triển khai một dự án kiểm thử tự động trên LLQ QA Platform được thực hiện như sau:



1.1 Thiết lập project

Một dự án ở giai đoạn khởi đầu, các bước để [thiết lập project kiểm thử](#) được thực hiện bao gồm:



- [Tạo Project](#)
- [Tạo Profile](#)
- [Tạo Environment](#)
- [Tạo Version](#)

- [Tạo Module](#)
- [Tạo Label](#)

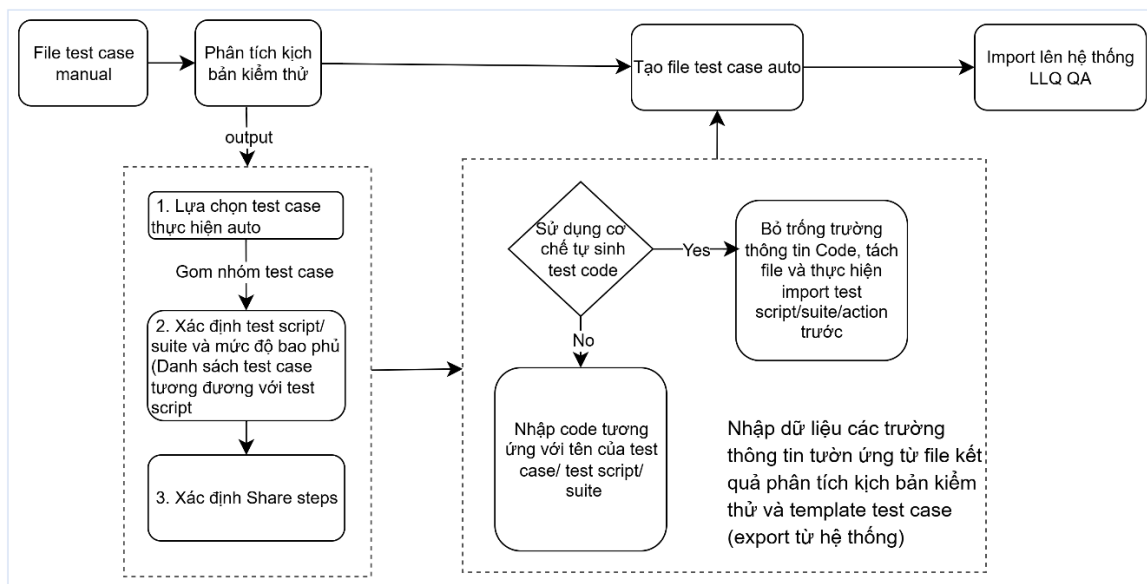
Trong quá trình nâng version cho ứng dụng/ hệ thống kiểm thử, việc cập nhật thiết lập cho project là cần thiết.

Project Owner không cần thực hiện tạo project mới, công việc khi này chỉ cần:

- **Tạo mới version** (bắt buộc), trong đó thông tin của version bao gồm **mã version**, **nền tảng** (nếu ứng dụng/ hệ thống được phát triển trên nhiều nền tảng) và **changelog** của version
- **Module** (nếu ứng dụng phát triển thêm Module mới)
- **Label**: phục vụ gắn nhãn cho những thay đổi của version, nếu có test case không còn phù hợp với version mới, thực hiện tạo Label “Remove on version ...” + code version

1.2 Chuyển đổi kịch bản kiểm thử manual sang auto

Khi bắt đầu vào dự án kiểm thử, việc xây dựng một file test case (tài liệu kiểm thử) là bắt buộc nhằm đảm bảo trong quá trình kiểm thử không bỏ sót trường hợp nào. Tùy thuộc vào giai đoạn phát triển của dự án mà khối lượng test case cần chuyển đổi từ tài liệu file excel sang auto là khác nhau.



1.2.1 Phân tích kịch bản kiểm thử

Trước khi bắt đầu xây dựng một kịch bản kiểm thử tự động, bước quan trọng đầu tiên là phân tích chi tiết các thành phần và bước thực hiện có trong kịch bản kiểm thử của test batch. Từ đó, ta tiến hành bóc tách các bước hoặc nhóm bước có thể dùng làm shared steps trong toàn bộ dự án kiểm thử.

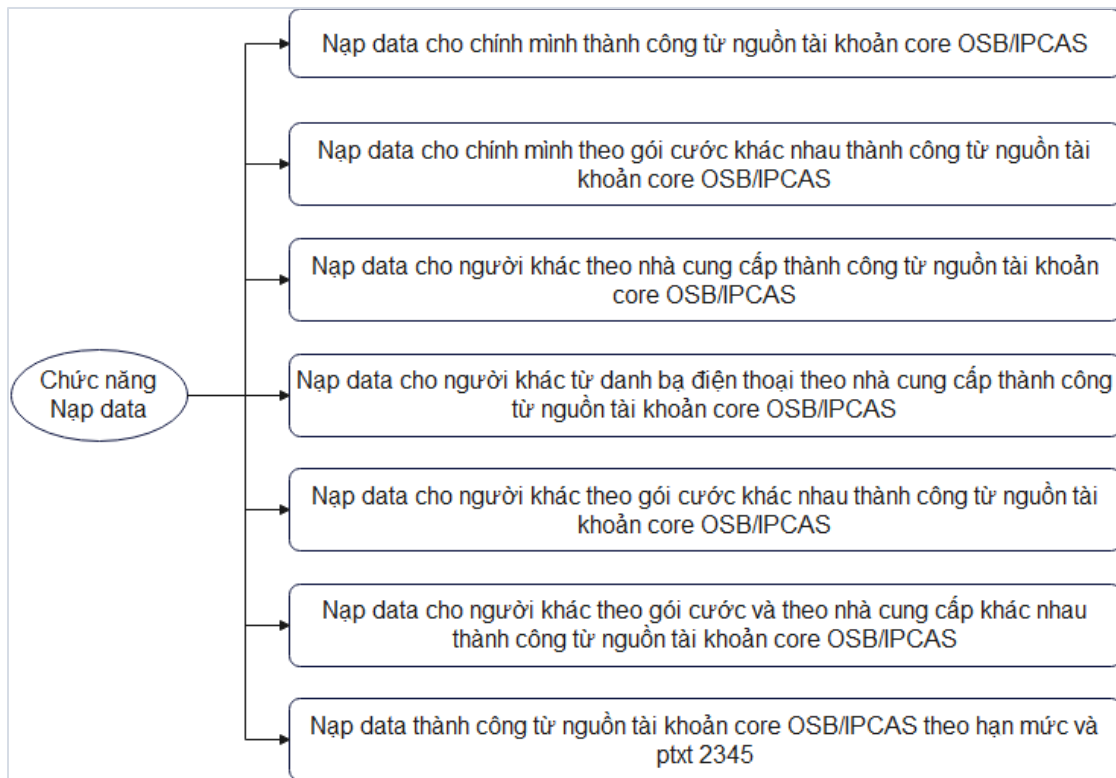
Việc xác định trước các shared steps không chỉ giúp tiết kiệm thời gian và công sức cho nhân viên kiểm thử mà còn giảm thiểu trùng lặp công việc giữa các thành viên trong nhóm. Nhờ vậy, tránh được việc nhiều người thực hiện lại những thao tác giống nhau một cách không cần thiết.

Shared steps là những bước hoặc nhóm bước có thể tái sử dụng trong nhiều hơn một kịch bản kiểm thử.

Ngoài ra, với những luồng của một chức năng hoàn thiện cần thực hiện khá dài, phức tạp, đòi hỏi nhiều rẽ nhánh, mức độ bao phủ cao thì có thể chia nhỏ các steps thành các shared steps phù hợp (thường là thao tác trên một màn hình cụ thể/ bước thực hiện khi có một điều kiện nhất định) giúp script ngắn gọn hơn, dễ theo dõi cũng như thuận tiện hơn khi bảo trì, cập nhật kịch bản kiểm thử khi có thay đổi.

Sau khi phân tích kịch bản kiểm thử, từ các steps thể hiện trên test batch, nhân viên kiểm thử có thể định hình được cần gọi đến shared steps nào, cách xử lý, sử dụng các keyword hợp lý giúp tối ưu kịch bản kiểm thử tự động, tinh giản các bước không cần thiết.

Ví dụ với chức năng Nạp data điện thoại có một số test case như sau:



1.2.1.1 Xác định kịch bản kiểm thử thực hiện tự động

Lựa chọn những test case không bắt buộc xuất hiện yếu tố manual như eKYC, captcha, facepay... để thực hiện kiểm thử tự động.

Với ví dụ đưa ra, test case “Nạp data thành công từ nguồn tài khoản OSB/IPCAS theo hạn mức và ptxt 2345” sẽ không được thực hiện auto do xuất hiện yếu tố ptxt 2345 (cần thực hiện facepay)

1.2.1.2 Xác định test script/ suite cho dự án

Để xác định script/ suite cần thực hiện, nhân viên kiểm thử cần xác định được các luồng chức năng cần thực hiện trong phạm vi kiểm thử.

Lựa chọn những test case có chung luồng chức năng, các bước thực hiện tương tự, chỉ khác nhau một số thao tác nhỏ trong luồng, có thể xác định bằng data input để lựa chọn rẽ nhánh của test case trong script.

Với ví dụ đưa ra, 6 test case được lựa chọn thực hiện tự động được gom vào 01 script “Nạp data điện thoại” với steps:

1. Mở ứng dụng

2. Đăng nhập vào ứng dụng
3. Click vào chức năng “Nạp tiền điện thoại/ Data”
4. Click tab Data
5. Lựa chọn tài khoản nguồn thực hiện giao dịch
6. Lấy số dư tài khoản nguồn trước giao dịch
7. Chọn nhà mạng
8. Nếu expect lựa chọn từ danh bạ
 - 8.1 Đúng, thực hiện click vào icon danh bạ, sau đó lựa chọn số điện thoại
 - 8.2 Sai, thực hiện nhập số điện thoại vào ô input, bỏ trống số điện thoại để nạp data cho chính mình
9. Click chọn gói cước theo mệnh giá đã nhập
10. Click Tiếp tục
11. Xác nhận thông tin trên màn hình giao dịch, Click Xác nhận
12. Xác thực giao dịch theo các phương thức mật khẩu/ SMS OTP/ PIN
13. Ghi nhận thông tin trên màn hình kết quả giao dịch
14. Lấy số dư sau giao dịch
15. Kiểm tra báo cáo giao dịch

1.2.1.3 Xác định shared steps

Từ các bước thực hiện script, có thể xác định được một số shared steps sẽ xuất hiện trong “Nạp data điện thoại”

- Đăng nhập vào ứng dụng là bước sẽ xuất hiện ở tất cả các script của project
- Lựa chọn tài khoản nguồn thực hiện giao dịch sẽ xuất hiện trong các script về giao dịch tài chính

Lấy số dư tài khoản nguồn xuất hiện 02 lần trong script Nạp data và hoàn toàn có thể xuất hiện ở trong các script của giao dịch tài chính

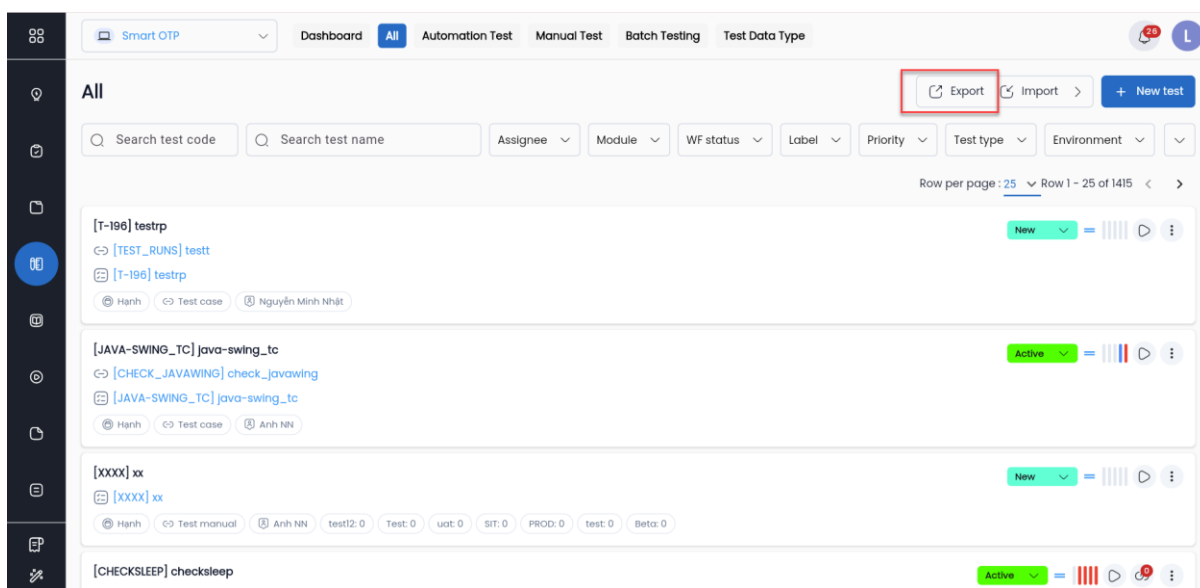
- Tương tự, kiểm tra báo cáo giao dịch và ghi nhận giao dịch trên BO là các bước xuất hiện trong các script của giao dịch tài chính

1.2.2 Tạo file test case auto

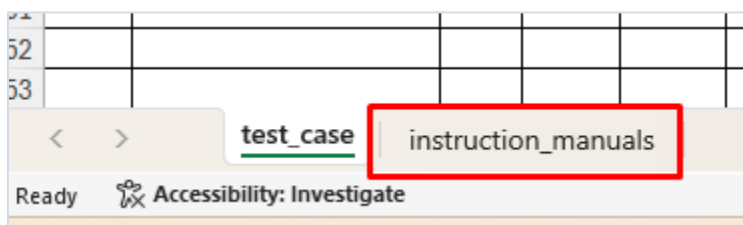
Sau khi phân tích kịch bản kiểm thử, file tài liệu kiểm thử đã có đầy đủ script/ suite và test case cần thực hiện kiểm thử tự động cùng các steps của script/ suite tương ứng.

Để tạo file test case auto, nhân viên kiểm thử cần lấy template từ hệ thống.

Bước 1: Trên màn hình Test Management, ở tab bất kỳ click < > “Export”



Bước 2: Mở file test case vừa tải về, click vào tab instructions_manual để xem hướng dẫn nhập liệu cho file



Bước 3: Nhập dữ liệu vào file test case auto với các trường thông tin tương ứng

#	Field	Required	Description
1	TYPE	Yes	Chọn một loại trường hợp kiểm thử trong danh sách có sẵn, bao gồm: - action (Shared steps) - test_case (Test case automation) - test_scenario (Test scenario) - test_suite (Test suite) - manual_test (Test case manual)
2	APPLICATION_MODULE	Yes	"Nhập mã module của của Ứng dụng cần kiểm thử. Module phải là giá trị đã được khai báo trong tab Module của project" "Phân chia hệ thống thành các module riêng lẻ giúp xác định và sửa lỗi nhanh chóng. Lỗi khi gặp phải thường dễ dàng xác định và có lập trong một module cụ thể mà không gây ảnh hưởng đến phần còn lại của hệ thống" - VD: Tiền gửi Tiền vay Chuyển tiền...
3	CODE	No	Nhập mã trường hợp kiểm thử. Mã phải bắt đầu bằng chữ cái, có thể bao gồm các ký tự chữ số, "_", "-", dấu tối thiểu 2 ký tự và là duy nhất trong toàn bộ project. Nếu không nhập code, hệ thống tự sinh mã test theo cấu trúc: T - số thứ tự VD: T-1
4	NAME	Yes	Nhập tên của trường hợp kiểm thử để mô tả tóm tắt trường hợp cần kiểm thử
5	LABELS	No	Nhập mã của nhãn trường hợp kiểm thử muốn gắn. Người dùng có thể gắn nhãn cho các trường hợp kiểm thử. Trong trường hợp chưa có labels, hệ thống sẽ tự động tạo labels mới. - Một trường hợp kiểm thử có thể gắn nhiều labels và cách nhau bởi dấu "," - VD: gắn nhãn trường hợp kiểm thử là positive hoặc negative hoặc các loại test case mong muốn. Positive, UI, UX
6	TEST_PRIORITY	Yes	Priority là mức độ ưu tiên cần thực hiện test. Mức độ ưu tiên càng cao thì cần thực hiện trước. - Priority trên hệ thống bao gồm: High, Medium, Low
7	PURPOSE	No	Mục đích, kết quả cuối cùng của mã trường hợp kiểm thử hướng đến
8	PRECONDITIONS	No	Nhập điều kiện tiên quyết tương ứng với từng loại trường hợp kiểm thử
9	INPUTS	No	Nhập dữ liệu kiểm thử (test data) được sử dụng trong quá trình thực hiện kiểm thử
10	STEPS	No	Nhập chi tiết các bước thực hiện trong trường hợp kiểm thử.
11	EXPECTED_RESULTS	No	Expected results (Kết quả mong muốn) của trường hợp kiểm thử.
12	DIRECTORY_PATH	No	Nhập tên của thư mục sẽ lưu trữ các loại trường hợp kiểm thử. Trong trường hợp chưa có thư mục, hệ thống sẽ tự động tạo thư mục mới. VD: /login
13	SCENARIO_SUITE_CODE	No	*** Chỉ bắt buộc phải nhập nếu Type = "test_case"(Test case automation). Nhập mã của Test scenario hoặc Test suite mà Type = "test_case"(Test case automation) tham chiếu đến, scenario/ suite phải được tạo ra trước test case . ***Trong trường hợp Type = "manual_test"(Test case manual), có thể nhập hoặc không. VD: T-1
14	SCENARIO_SUITE_NAME	No	Tên của test scenario/test suite mà test case tham chiếu đến
15	APPLICATION_ENVIRONMENT	Yes	Nhập mã môi trường triển khai của Ứng dụng cần kiểm thử. - VD: development(test) sit uat production Environment phải là giá trị đã được khai báo trong Project
16	APPLICATION_VERSION	Yes	Nhập code phiên bản(bản build) hiện tại của Ứng dụng cần kiểm thử. - VD: v1.0, v1.1, ... Version phải là giá trị đã được khai báo trong tab Version của Setting Project
17	WF_STATUS	Yes	Workflow status là các trạng thái phê duyệt của trường hợp kiểm thử - Active: đã phê duyệt - Review: đang xem xét

Bước 4: Save file test case auto

Tại bước 3, ngoài các thông tin cơ bản tương của type test_case được nhập tương tự type test_script, test_suite thì test case còn có trường thông tin SCRIPT_SUITE_CODE và SCRIPT_SUITE_NAME, việc nhập liệu đầy đủ và chính xác ở 2 trường thông tin này giúp script và test case được mapping với nhau, đảm bảo không bị bỏ sót số lượng test case của script cũng như xác định được có còn thiếu script/suite nào hay không.

Để thực hiện mapping test script/suite code với test case thì script/suite cần phải có code

- **Trường hợp 1:** script/ suite sử dụng code nhập tay trên file excel

Trực tiếp nhập script/ suite code tương ứng với tên test script/suite của test case (kết quả của phân tích kịch bản kiểm thử)

- **Trường hợp 2:** script/ suite sử dụng code sinh tự động từ hệ thống

1. Thực hiện import file test auto đã tạo cho type script/suite
2. Export danh sách test script/suite trên tab tương ứng của chức năng Test Management, move cột CODE từ cột C sang cột D
3. Mở file test case auto của type test_case (file 1) và file test case vừa tải về (file 2)

A	B	C	D	E	F	G	H
TYPE*	APPLICATION_MODULE*	CODE	NAME*	SCENARIO_SUITE_CODE	SCENARIO_SUITE_NAME	APPLICATION_ENVIRONMENT	APR
test_case	BO		test	get testing		JAT	And
test_case	Module đăng nhập		Android - Upapp - Cài đặt		Android - Install app - Scenario	JAT	And
test_case	Module đăng nhập		iOS - Upapp - Cài đặt		iOS - Install App (Test scenario)	JAT	And
test_case	Module tiết kiệm		iOS - M0 - Tài toán tài khoản tiền gửi trực tuyến từ tài khoản nguồn mặc định		iOS - M0 - Tài toán tài khoản tiền gửi trực tuyến	PRO	V5.0
test_case	Module tiết kiệm		Android - M0 - Tài toán tài khoản tiền gửi trực tuyến từ tài khoản nguồn mặc định		Android - M0 - Tài toán tài khoản tiền gửi trực tuyến	PRO	V5.0
test_case	Module thanh toán hoá đơn		Android - Thanh toán hoá đơn bảo hiểm thành công từ danh bạ thu hướng từ nguồn tài khoản core CSB/TPCAS [Payment Type = Topup]		Android - Thanh toán hoá đơn bảo hiểm	JAT	V5.0
test_case	Module gửi tiền mừng		iOS - Gửi tiền mừng thành công đến một người từ popup tiền mừng có 1 tên từ nguồn tài khoản core CSB/TPCAS		iOS - Gửi tiền mừng và Gửi tiền mừng từ popup gửi tiền mừng	JAT	V5.0

File 1

A	B	C	D	E	F	G	H	I
TYPE*	APPLICATION_MODULE*	NAME*	CODE	LABELS	TEST_PRIORITY*	PURPOSE	PRECONDITIONS	INPUTS
test_scar	BO	Chuyển tiền nhanh liên ngân hàng	T-457		MEDIUM			
test_scar	Module tiết kiệm	iOS - Install App (Test scenario)	TS_T-312	Mios	HIGH			
test_scar	Module đăng nhập	Android - Install app - Scenario	TS_T-250		HIGH			
test_scar	Module gửi tiền mừng	ANDROID - Gửi tiền mừng nhiều người không kiểm tra điều kiện	T-647	Auto,#functional,#positive,#andro	HIGH			
test_scar	Module gửi tiền mừng	ANDROID - Gửi tiền mừng không kiểm tra điều kiện	T-646	Auto,#functional,#positive,#andro	HIGH			
test_scar	Module gửi tiền mừng	iOS - Gửi tiền mừng không kiểm tra thông tin	T-645	#functional,#positive,#ios,Auto	HIGH			
test_scar	Module gửi tiền mừng	iOS - Gửi tiền mừng nhiều người không kiểm tra thông tin	T-644	#functional,#positive,#ios,Auto	HIGH			
test_scar	Module chuyển tiền nội bộ/liên ngân hàng/tích chuyển tiền	Upapp_Android - Chuyển tiền trong Agribank	android_upapp_transfer	Android,#functional,#positive	HIGH			
test_scar	Module chuyển tiền nội bộ/liên ngân hàng/tích chuyển tiền	iOS - Cài đặt và Chuyển tiền trong Agribank	upapp_transfer	#functional,#positive,#ios,Auto	HIGH			
test_scar	Module thanh toán hoá đơn	iOS - Thanh toán hoá đơn Vnpay từ QR	T-614		HIGH			
test_scar	Module thanh toán hoá đơn	Android - Thanh toán hoá đơn Vnpay từ QR	T-613		HIGH			
test_scar	Module chuyển tiền nội bộ/liên ngân hàng/tích chuyển tiền	Android - Chuyển tiền từ danh bạ thu hướng - LIVE	T-311_NEW	Auto,#functional,#positive,#andro	HIGH			
test_scar	Module nạp tiền điện thoại/nạp data	LIVE - Android - Nạp data điện thoại	LIVE-T-297	Auto,#functional,#positive,#andro	HIGH			

File 2

4. Trên file 1, ở cột SCRIPT_SUITE_CODE, nhập hàm VLOOKUP để tham chiếu giá trị CODE của script/suite với tên script/suite tương ứng từ file 2, áp dụng cho tất cả các test case

Vlookup (N2,['test_case (2).xlsx] test_case'!\$C:\$D,2,0)

Trong đó:

N2: là cột – hàng của test script/suite name tương ứng với test case ở hàng 2
'[test_case (2).xlsx] test_case'! – tên file test auto type test script, sheet test_case

\$C:\$D: từ cột C đến cột D, ở đây cột C là Name Script, cột D là CODE

2: số thứ tự thứ 2 kể từ cột đầu tiên trong vùng lấy giá trị (C-D), ở đây sẽ lấy giá trị cột D

0: Giá trị trả ra là tuyệt đối

=VLOOKUP(N4,'[test_case (2).xlsx]test_case'!\$C:\$D,2,0)					
A	B	C	D	M	N
TYPE*	APPLICATION_MODULE*	CODE	NAME*	SCENARIO_SUITE_CODE	SCENARIO_SUITE_NAME
test_case	BO		Tesst	apt-testing	apt testing
test_case	Module đăng nhập		Android - Upapp - Cài đặt	ts_T-250	Android - Install app - Scenario
test_case	Module đăng nhập		iOS - Upapp - Cài đặt	ts_T-312	iOS - Install App (Test scenario)
test_case	Module tiết kiệm		iOS - Mở - Tắt toán tài khoản tiền gửi trực tuyến từ tài khoản nguồn mặc định		iOS - Mở - Tắt toán tài khoản tiền gửi trực tuyến
test_case	Module tiết kiệm		Android - Mở - Tắt toán tài khoản tiền gửi trực tuyến từ tài khoản nguồn mặc định		Android - Mở - Tắt toán tài khoản tiền gửi trực tuyến

5. Save file và import lên project để tạo hàng loạt test case

1.3 Cấu hình kịch bản kiểm thử tự động (Script, Suite, Shared steps)

Thành phần của một kịch bản kiểm thử tự động bao gồm:

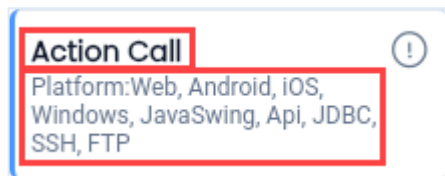
- Metadata: Thông tin chung của kịch bản kiểm thử
- Client: platform, device thực thi kiểm thử
- Keyword: Các khối câu lệnh thể hiện hành động, thuật toán của kịch bản kiểm thử.
- Tham số: input/ output của kịch bản

1.3.1 Keyword

1.3.1.1 Định nghĩa các loại keyword

Keyword là các hành động tương tác của người dùng trên ứng dụng kiểm thử

Các loại keyword bao gồm: Client, Action, Control, Variable, Operator, Assert, File, Event, Security, Other



Một keyword có 2 phần:

1. Tên Keyword
2. Nền tảng được sử dụng keyword

1.3.1.2 Cấu trúc của keyword

Trong mỗi keyword thường có các thông tin:

- Keyword: thể hiện mục đích hành động của người dùng lên phần tử như Click, xóa và nhập, tìm kiếm, lựa chọn...

- Description: Mô tả, diễn giải hành động của bước cấu hình
- Các thông tin một keyword cần xác định là keyword đó sẽ tương tác đến phần tử nào (Locator) và nguồn dữ liệu đầu vào/dữ liệu đầu ra (input/output). Tùy thuộc vào từng keyword thì số lượng và loại thuộc tính sẽ khác nhau

Clear And Send Keys property

×

Step comment

Enter a description

Locator

Element

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

Elements

▼

Enter a xpath

Find value

Show position config

Input

value

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

String

▼

Enter a variable

Cancel

Done

1.3.1.3 Các loại giá trị trong keyword

1.3.1.3.1 Type

Đa phần các thuộc tính của keyword đều mang một dữ liệu dù là người dùng truyền vào hay là kết quả đầu ra của keyword đó. Dữ liệu đó có thể là tham số, có thể là một giá trị cụ thể tùy bài toán đưa ra của kịch bản kiểm thử.

Kiểu giá trị đó người dùng có thể chủ động lựa chọn khi thực hiện cấu hình keyword

Value
Variable
Agent Parameter
Device Parameter
Version Parameter
Environment Parameter

String
Enter a name

Find variable

Các kiểu giá trị mà nền tảng kiểm thử hỗ trợ: Value, Variable, new Variable, Version Parameter, Environment Parameter, Agent Parameter, Device Parameter

Trong đó:

Type	Mô tả
Value	Là một giá trị cụ thể mà người dùng nhập khi cấu hình keyword và không thay đổi khi thực thi kiểm thử. Value không xuất hiện ở thuộc tính output của keyword
Variable	Là tham số, không cố định giá trị, có thể thay đổi tùy thuộc vào ý đồ của người dùng
Version Parameter	Giống như variable, tuy nhiên giá trị của tham số sẽ được thay đổi phụ thuộc vào version được lựa chọn để thực thi. Giá trị của version parameter được định nghĩa ở thiết lập dự án nội dung version và parameter (Xem thêm khởi tạo và cấu hình tham số version parameter)
Environment Parameter	Tương tự như version parameter, nhưng giá trị của tham số sẽ thay đổi phụ thuộc vào môi trường phát triển được lựa chọn khi thực thi kịch bản kiểm thử. Giá trị của Environment parameter được định nghĩa ở thiết lập dự án nội dung environment và parameter
Agent Parameter	Tương tự như version parameter nhưng giá trị của tham số sẽ thay đổi phụ thuộc vào Agent thực thi kịch bản kiểm thử. Giá trị của Agent parameter được định nghĩa ở thiết lập dự án nội dung parameter
New Variable	Giống như Variable, nhưng chỉ xuất hiện ở thuộc tính returnType (trả ra output của keyword).

Type	Mô tả
	Khi chọn new Variable, người dùng tạo ra một tham số mới cho kịch bản kiểm thử.
None	Xuất hiện ở thuộc tính output, không lấy giá trị đầu ra của keyword

1.3.1.3.2 Kiểu dữ liệu (type)

Là kiểu dữ liệu của một value hoặc một variable. Kiểu dữ liệu là cách để phân loại dữ liệu giúp cho công cụ kiểm thử tự động có thể hiểu được dữ liệu đang sử dụng.

Các kiểu dữ liệu được hỗ trợ sử dụng bởi nền tảng:

Data type	Mô tả	Ghi chú
CallOutput	Kiểu dữ liệu lấy ra là output của keyword đó	Sử dụng khi keyword là Action call, Test Script call, API call hoặc khi cần gán giá trị của một tham số bằng output của shared steps/ test script/ api được gọi đến
String	Kiểu dữ liệu của giá trị/ biến là dạng chuỗi ký tự	Sử dụng khi giá trị là một chữ, câu văn, văn bản..
Integer	Kiểu số nguyên: có thể thực hiện các phép toán số học đối với các giá trị có kiểu dữ liệu integer	Sử dụng khi giá trị là số nguyên, có vùng giá trị từ -32,768 đến 32,768 Khi thuật toán của kịch bản kiểm thử yêu cầu phải thực hiện phép toán số học, kiểu dữ liệu cần sử dụng là integer
Long	Kiểu số nguyên , giống integer tuy nhiên có kích thước lưu trữ lớn hơn	Sử dụng khi giá trị là số nguyên, có vùng giá trị từ -2,147,483,648 đến 2,147,483,648

Data type	Mô tả	Ghi chú
Float	Kiểu số thực với độ chính xác đơn, phạm vi giá trị từ 3.4E-38 đến 3.4E+38, có thể lưu được 6 chữ số ở phần thập phân	Sử dụng khi giá trị là số thực, không cần độ chính xác quá cao
Double	Kiểu số thực với độ chính xác kép, phạm vi giá trị từ 1.7E-308 đến 1.7E+308, có thể lưu được 15 chữ số ở phần thập phân	Sử dụng khi cần thực hiện tính toán với số thực và yêu cầu độ chính xác cao
Date	Là kiểu dữ liệu để lưu trữ ngày tháng theo định dạng ‘YYYY-MM-DD’	
DateTime	Là kiểu dữ liệu lưu trữ ngày tháng cùng với thời gian	
Elements	Là một Element mà keyword tương tác đến trên hệ thống/ ứng dụng kiểm thử. Value của kiểu dữ liệu luôn thể hiện vị trí của element	Sử dụng trong thuộc tính Locator của keyword, returnType của keyword có chức năng tìm kiếm element Đặc biệt dữ liệu đầu vào của Foreach là một mảng kết quả từ Find Element/ Find Elements thì type của foreach là Elements
Repository Element	Tương tự như Elements, tuy nhiên giá trị của kiểu dữ liệu này được thiết lập trong chức năng Repository	Sử dụng trong thuộc tính Locator của keyword
Password	Giống kiểu String, tuy nhiên khi thao tác nhập value có kiểu dữ liệu là Password thì kí tự sẽ được mã hoá	Sử dụng cho các trường thông tin cần được bảo mật như Mật khẩu, CCCD...

Data type	Mô tả	Ghi chú
Null	Chỉ gồm một giá trị <i>null</i> , biểu hiện không biết giá trị của tham số	Sử dụng khi người dùng chỉ cần tạo một tham số nhưng chưa biết giá trị của nó
Empty String	Giá trị truyền vào không chứa kí tự nào	
Random Value	Giá trị của truyền vào ở thuộc tính có kiểu dữ liệu Random được sinh tự động, thiết lập ở chức năng Random type của hệ thống	Sử dụng khi giá trị truyền vào là ngẫu nhiên, có thể theo một mẫu nào đó hoặc không Ví dụ các trường thông tin CCCD, số điện thoại...

Lưu ý: Chọn sai kiểu dữ liệu có thể dẫn đến lỗi cấu hình và không thể thực thi được kịch bản kiểm thử.

1.3.1.4 *Sử dụng keyword: xem thêm file [Keyword Document](#)*

1.3.2 *Xác định phần tử*

Locator là vị trí, địa chỉ của phần tử cần được tương tác bởi keyword trên màn hình ứng dụng/ hệ thống được kiểm thử. Nếu không có locator, keyword không thể tìm đến chính xác phần tử cần được tương tác.

Cách xác định locator:

- Sử dụng Find element
- Sử dụng element repository
- Viết xpath thủ công

1.3.2.1 *Sử dụng Find element*

Để sử dụng Find Element, người dùng phải chắc chắn đã Start Debug

Bước 1: Click vào keyword cần xác định locator, lựa chọn Data type là Elements

Click property

Step comment

Enter a description

Locator

Element

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

Repository Element

Search

X

Element Repository

Elements

Repository Element

Password

Null

Empty String

Cancel

Done

Hệ thống hiển thị button Find Element

Click property

Step comment

Enter a description

Locator

Element

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

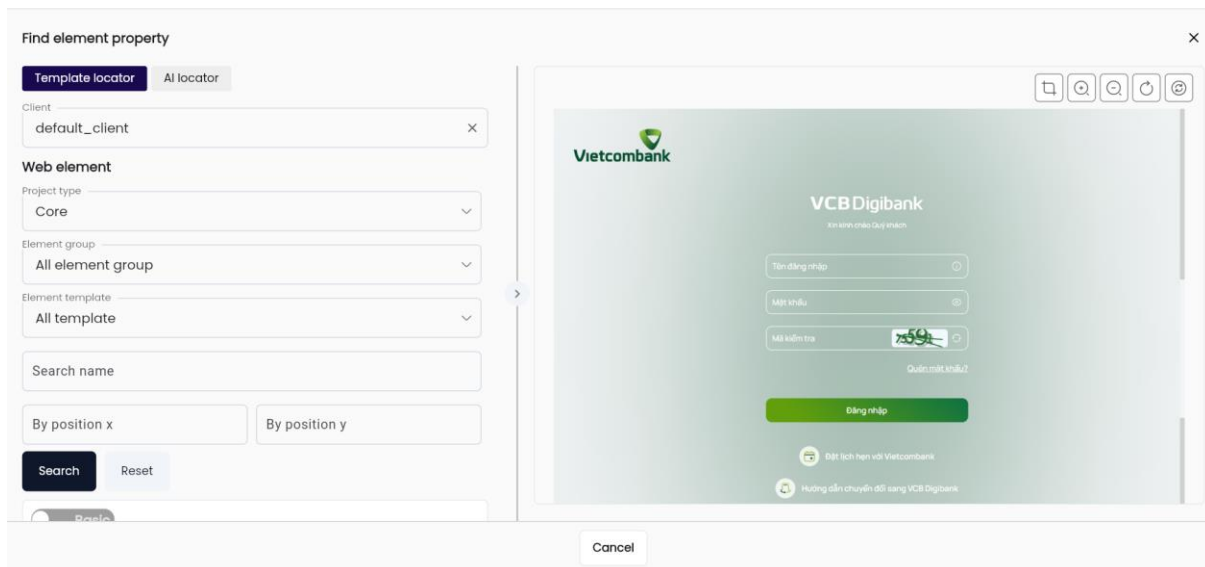
Elements

Enter a value

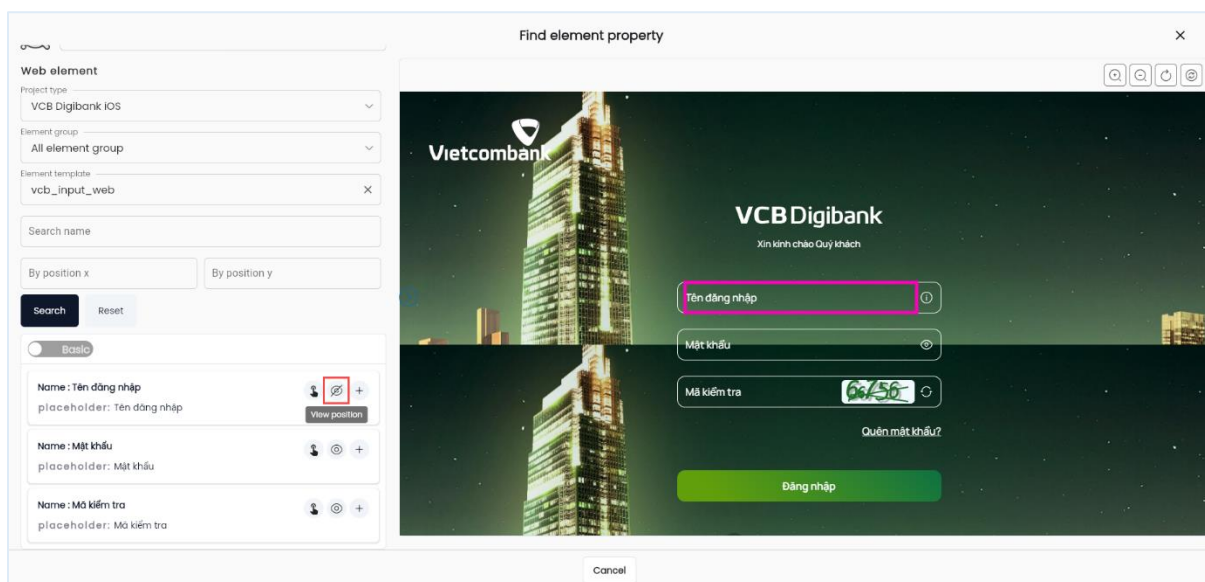
Find element

Find value

Bước 2: Click Find element



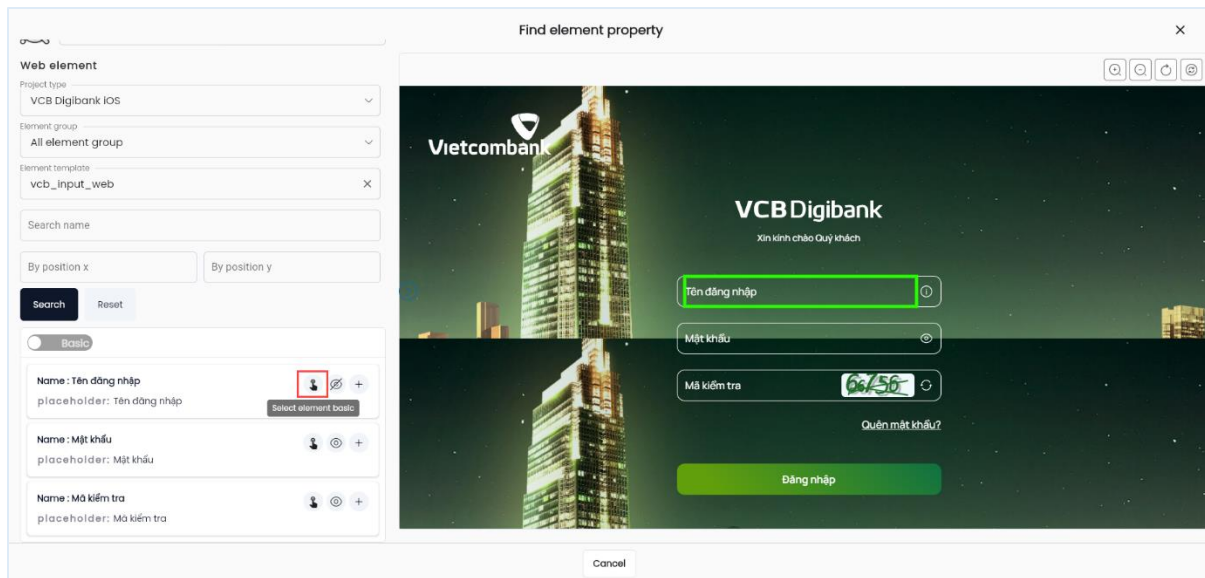
Bước 3: Lựa chọn Project type, Element group và element template để tìm element mong muốn



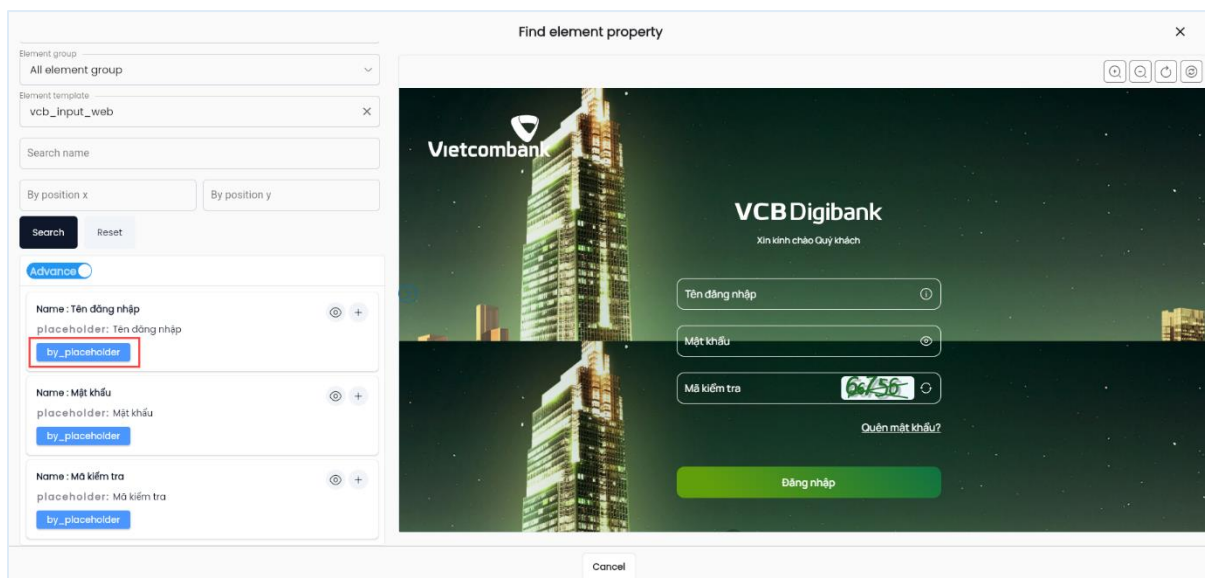
Click view position để xác định vị trí của element trên màn hình của ứng dụng/ hệ thống được kiểm thử

Bước 4: Chọn element

- Với Mode basic: Click Select element basic để lựa chọn element. Với mode basic, hệ thống mặc định lấy giá trị xpath của element đầu tiên mà nó tìm thấy theo thứ tự được sắp xếp trong element template



- Với Mode advance: Click chọn thuộc tính/ name của phương thức lấy xpath từ template. Với mode advance, người dùng có thể chủ động lựa chọn một giá trị xpath cụ thể phù hợp với bài toán kiểm thử



Hệ thống hiển thị thông tin keyword với locator đã được xác định

Click property

Step comment

Enter a description

Locator

Element

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

Elements

Đăng nhập

×

+

☰

Element content

text

Đăng nhập

Find element

Find value

Cancel

Done

Bước 5: Click **Done** để hoàn thiện cấu hình keyword

1.3.2.2 Sử dụng element repository

Trước khi sử dụng element repository trong cấu hình, dự án kiểm thử cần được xây dựng kho repository và lưu trữ phần tử trong đó (thông qua Smart Locator, Find element...)

Bước 1: Trên màn hình config, click vào keyword cần xác định phần tử, hệ thống mặc định Data type của locator là Repository Element. Click Element Repository để lựa chọn phần tử.

Click property

Step comment

Enter a description

Locator

Element

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

Repository Element

Element Repository

Cancel

Done

Bước 2: Lựa chọn kho repository chứa phân tử cần tương tác

web-login

Select element repository

Elements

Search element name

Project type

All project type

Element group

All element group

Element template

All template

Reset

ELEMENT (4)

Basic

Sort by

Mã kiểm tra

Element template: vcb_input_web

Last update: 23-04-2025 16:56:28

Mật khẩu

Element template: vcb_input_web

Last update: 23-04-2025 16:56:28

Tên đăng nhập

Smart Locator

New repository

New element

Vietcombank

VCBDigibank

Xin kính chào Quý khách

Tên đăng nhập

Quý khách vui lòng nhập tên đăng nhập VCB Digibank

Mật khẩu

Quý khách vui lòng nhập mật khẩu VCB Digibank

Mã kiểm tra

Quý khách vui lòng nhập mã kiểm tra

Quên mật khẩu?

Đăng nhập

Đặt lịch hẹn với Vietcombank

Hướng dẫn chuyển đổi sang VCB Digibank

Bước 3: Click select element basic (ở mode basic) hoặc click name của phương thức xác định xpath (ở mode advance) để lựa chọn element

Bước 4: Click **Done** để hoàn thiện cấu hình keyword

1.3.2.3 *Viết xpath thủ công (không khuyến nghị)*

Để có thể viết xpath thủ công, nhân viên kiểm thử cần hiểu về cấu trúc, cách xác định xpath của element. Người dùng chỉ nên viết xpath thủ công khi element có cấu trúc phức tạp, chưa được khai báo template trên hệ thống LLQ QA Platform.

Tham chiếu mẫu xpath [cơ bản](#) và [nâng cao](#)

1.3.2.4 *Phần tử “động” trong kiểm thử tự động*

Trong kiểm thử tự động, đôi khi xpath của phần tử được tương tác thay đổi theo giá trị truyền vào (input) của kịch bản. Với mỗi giá trị truyền vào khác nhau, keyword sẽ tương tác đến một phần tử khác nhau. Việc xét điều kiện cho giá trị truyền vào (If – else) với tập giá trị của input để thao tác với phần tử nào sẽ khiến cho kịch bản kiểm thử trở nên dài và phức tạp. Trong trường hợp xpath của phần tử liên quan trực tiếp đến input (chứa giá trị truyền vào) thì cần tạo ra xpath “động”, có thể thay đổi khi giá trị truyền vào thay đổi mà không cần xét điều kiện phức tạp.

1.3.2.4.1 Tạo xpath “động” thủ công

Ví dụ: Để lựa chọn mệnh giá, kịch bản kiểm thử cần thực hiện truyền dữ liệu mong muốn cho mỗi test case là 30,000 VND, 50,000 VND, ...



Để có thể tạo ra xpath “động”, nhân viên kiểm thử cần xác định được xpath cụ thể của mỗi phần tử được lựa chọn.

Bước 1: Xác định xpath của phần tử cụ thể

Xpath của phần tử trong ví dụ này là:

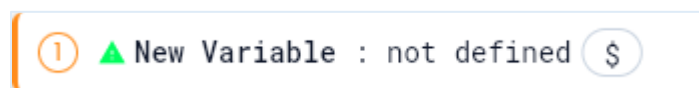
```
//XCUIElementTypeStaticText[@value = “30,000 VND”]
```

Có thể thấy giá trị mệnh giá thay đổi, xpath trực tiếp được thay đổi.

Để tạo xpath “động”, sử dụng keyword: NewVariable với giá trị có cấu trúc: value + Variable – mệnh giá + value, trong đó:

- Value 1 = //XCUIElementTypeStaticText[@value = “
- Variable = mệnh giá – tham số truyền vào
- Value 2 = ”] – phần đóng, hoàn thiện xpath

Cấu hình keyword New Variable:



Bước 2: Click vào “*not defined*” để khai báo tên cho biến xpath “động”

New Variable property



Step comment

Enter a description

Data type

Choose a data type



Name

Enter a name

isList

False



Variable init

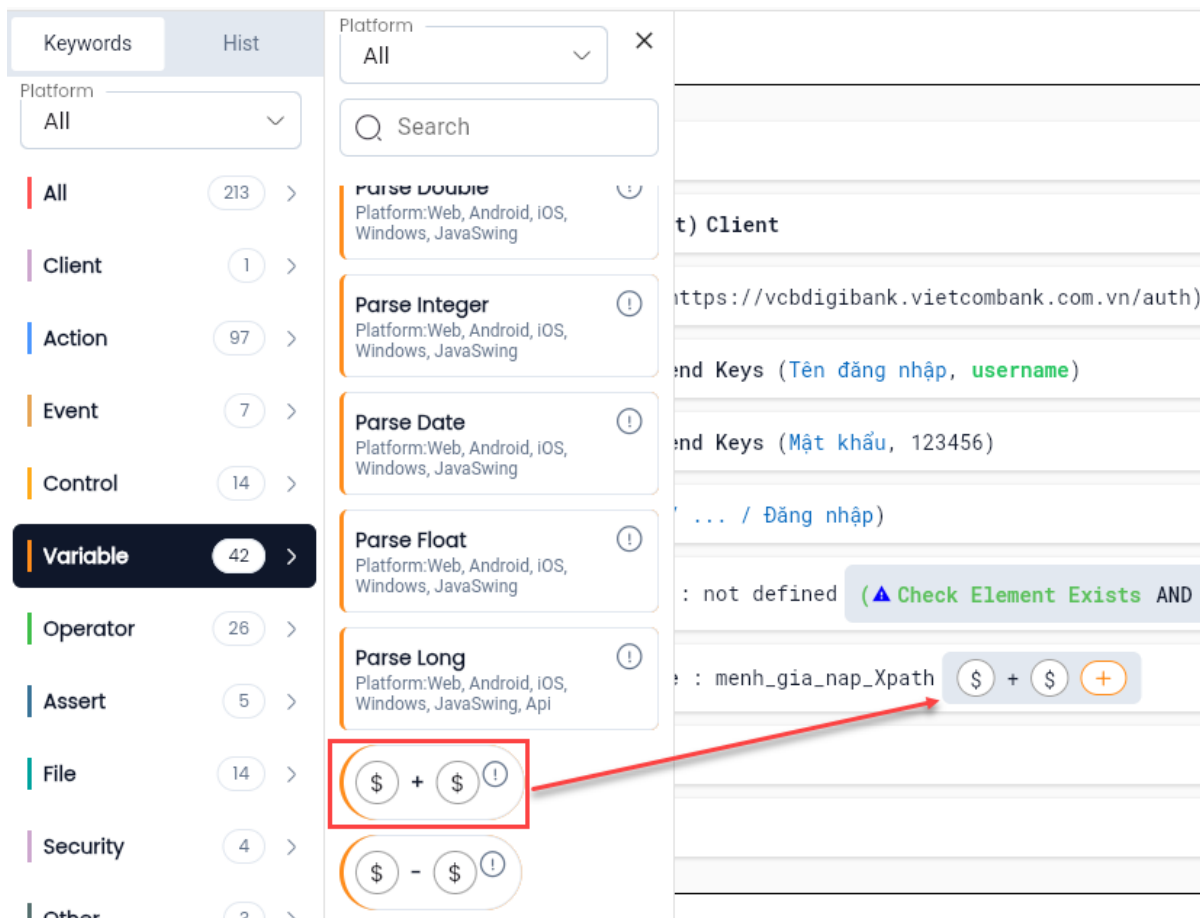
False



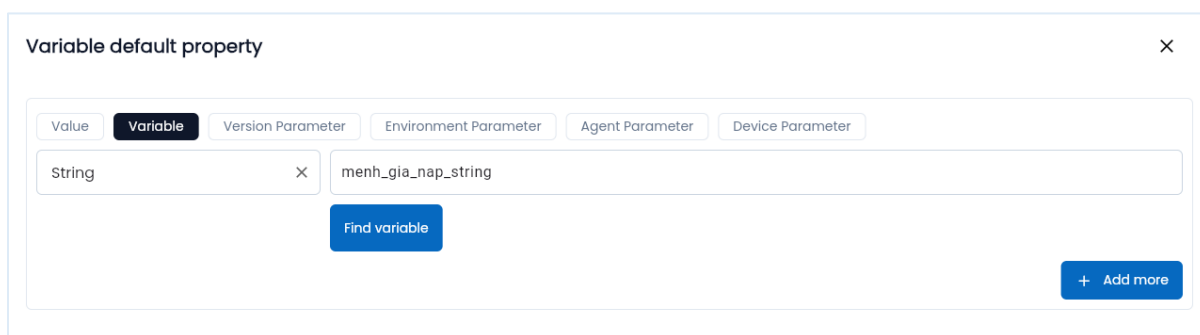
Trong đó:

- Step comment: Nhập ghi chú cho bước thao tác của keyword, giúp kịch bản kiểm thử được hiểu một cách rõ ràng hơn
- Data Type: String
- Name: Nhập tên của biến, đối với biến xpath “động”, nhân viên kiểm thử nên đặt tên liên quan đến phần tử được tương tác cùng với “xpath” để dễ dàng phân biệt các tham số chỉ vị trí và giá trị
- isList: false
- Variable init: false

Bước 3: Kéo keyword toán tử “+” vào vị trí \$, thể hiện giá trị của biến là cộng chuỗi



Bước 4: Nhập giá trị cho từng thành phần của keyword



Khi đó ta có:



Để sử dụng biến xpath đã tạo, trong keyword click có vai trò lựa chọn mệnh giá, thay vì nguồn dữ liệu là Value, ta sử dụng [Variable](#) và [Data type](#) là Elements

Click property

×

Step comment

Click chọn mệnh giá nạp

Locator

Element

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

Elements

▼

menh_gia_nap_xpath

Find variable

+ Add more

1.3.2.4.2 Custom element

Đối với các phần tử không thể xác định qua Find Element, bắt buộc người dùng phải sử dụng New Variable và thao tác click riêng. Tuy nhiên đối với các phần tử “động”, đã được detect template, nhân viên kiểm thử có thể sử dụng chức năng Custom element, giúp giảm bớt thao tác và tận dụng được tài nguyên đã có.

Ở ví dụ trên, trong keyword Click, nhân viên kiểm thử lựa chọn element repository hoặc Find element để chọn element cần sử dụng

Click property

×

Step comment

Click hiển thị danh sách tài khoản nguồn

Locator

Element

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

Repository Element

▼

WEB / 7. Nạp tiền điện thoại / 7.1 Màn khởi tạo / 8880115825

×

⌵

Element Repository

Show position config

Bước 1: Click custom element, hệ thống hiển thị màn hình Create custom element

BIDV - SmartBanking

CREATE CUSTOM ELEMENT

* Often used in cases where the elements that need to interact will change depending on the input value selected

Attribute of element template(Web_Text_005)

text02_text

☐ Enable

Cancel Save

Trong đó: text là các thuộc tính được sử dụng để tạo template cho element.

Bước 2: Click vào checkbox tương ứng với thuộc tính sử dụng để truyền giá trị “động” cho element

BIDV - SmartBanking

CREATE CUSTOM ELEMENT

* Often used in cases where the elements that need to interact will change depending on the input value selected

Attribute of element template(Web_Text_005)

text02_text

☒ Enable

Value Variable Version Parameter Environment Parameter Agent Parameter Device Parameter

String Enter a variable

Find variable

+ Add more

Cancel Save

Bước 3: Sử dụng Find Variable để lựa chọn tham số truyền vào (nếu đã được khai báo trong kịch bản kiểm thử) hoặc nhập tên cho parameter nếu chưa được khai báo và là giá trị được truyền vào khi thực thi kiểm thử.

BIDV - SmartBanking

H

CREATE CUSTOM ELEMENT

* Often used in cases where the elements that need to interact will change depending on the input value selected

Attribute of element template(Web_Text_005)

text02_text

Enable

ValueVariableVersion ParameterEnvironment ParameterAgent ParameterDevice Parameter

String

Find variable

+ Add more

Cancel

Save

Bước 4: Click Save

Click property

Step comment

Locator

Element

ValueVariableAgent ParameterDevice ParameterVersion ParameterEnvironment Parameter

Repository Element

Custom element

text02_text

Element Repository

Show position config

Cancel

Done

Bước 5: Click Done để hoàn thiện cấu hình cho keyword

1.3.3 Tham số

Trong kịch bản kiểm thử luôn tồn tại những tham số, mỗi tham số mang trong mình một giá trị, có thể cố định, có thể thay đổi tùy thuộc vào vị trí nó xuất hiện và vai trò trong kịch bản kiểm thử tự động.

Hệ thống kiểm thử LLQ QA Platform có 03 loại tham số của kịch bản:

- Variable: tham số (biến số) được sinh ra trong quá trình kiểm thử của luồng, được khai báo dưới dạng new variable thông qua keyword New Variable hoặc các keyword có thuộc tính output và có Type là newVariable
- Parameter (input): là giá trị đầu vào của kịch bản kiểm thử
- Output: là đầu ra, kết quả trả về của kịch bản kiểm thử tự động

Một tham số chỉ có thể được khai báo 01 lần trong kịch bản kiểm thử bao gồm dưới dạng new variable, parameter và output.

Khuyến nghị người dùng sử dụng **Find variable** khi sử dụng variable đã khai báo trong kịch bản kiểm thử.

1.3.3.1 Variable

Là tham số (biến số) được sinh ra trong quá trình kiểm thử của luồng, được khai báo dưới dạng new variable thông qua keyword New Variable hoặc các keyword có thuộc tính output và có Type là newVariable.

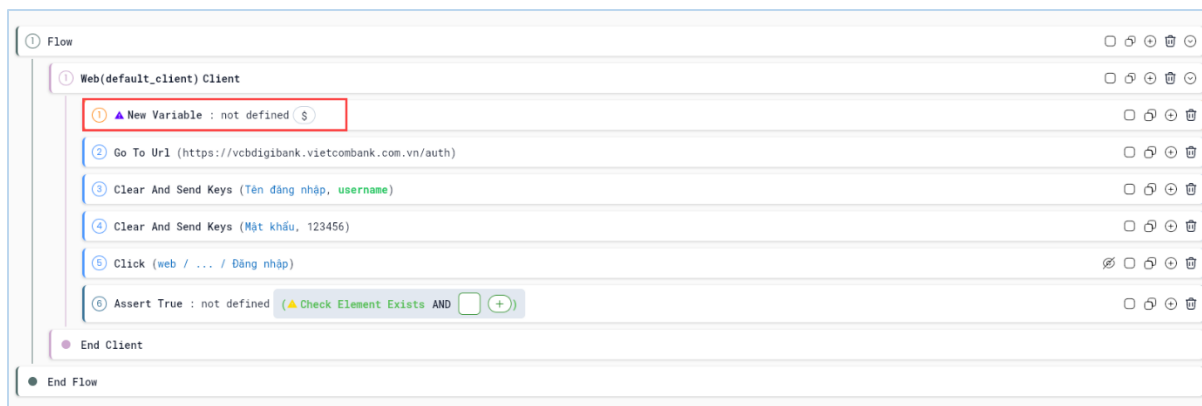
1.3.3.1.1 Đặc điểm:

- Không xuất hiện trong danh sách parameter hay tab output của metadata
- Nhân viên kiểm thử không trực tiếp truyền giá trị cho nó bằng cách nhập liệu
- Variable được khai báo dưới dạng newVariable ở trong Keyword If – else chỉ có thể được sử dụng trong if -else đó, **không thể được gọi đến ở bên ngoài** if – else dù điều kiện của if có xảy ra hay không.
- Tại keyword khai báo new variable, nếu đổi type từ new variable thành Variable hoặc một type khác, tham số sẽ bị xóa khỏi danh sách tham số của script

1.3.3.1.2 Cách tạo variable

Cách 1: Sử dụng keyword New Variable

Bước 1: Trên màn hình config, kéo thả keyword New Variable



Bước 2: Nhấn **not defined** để khai báo tham số

New Variable property



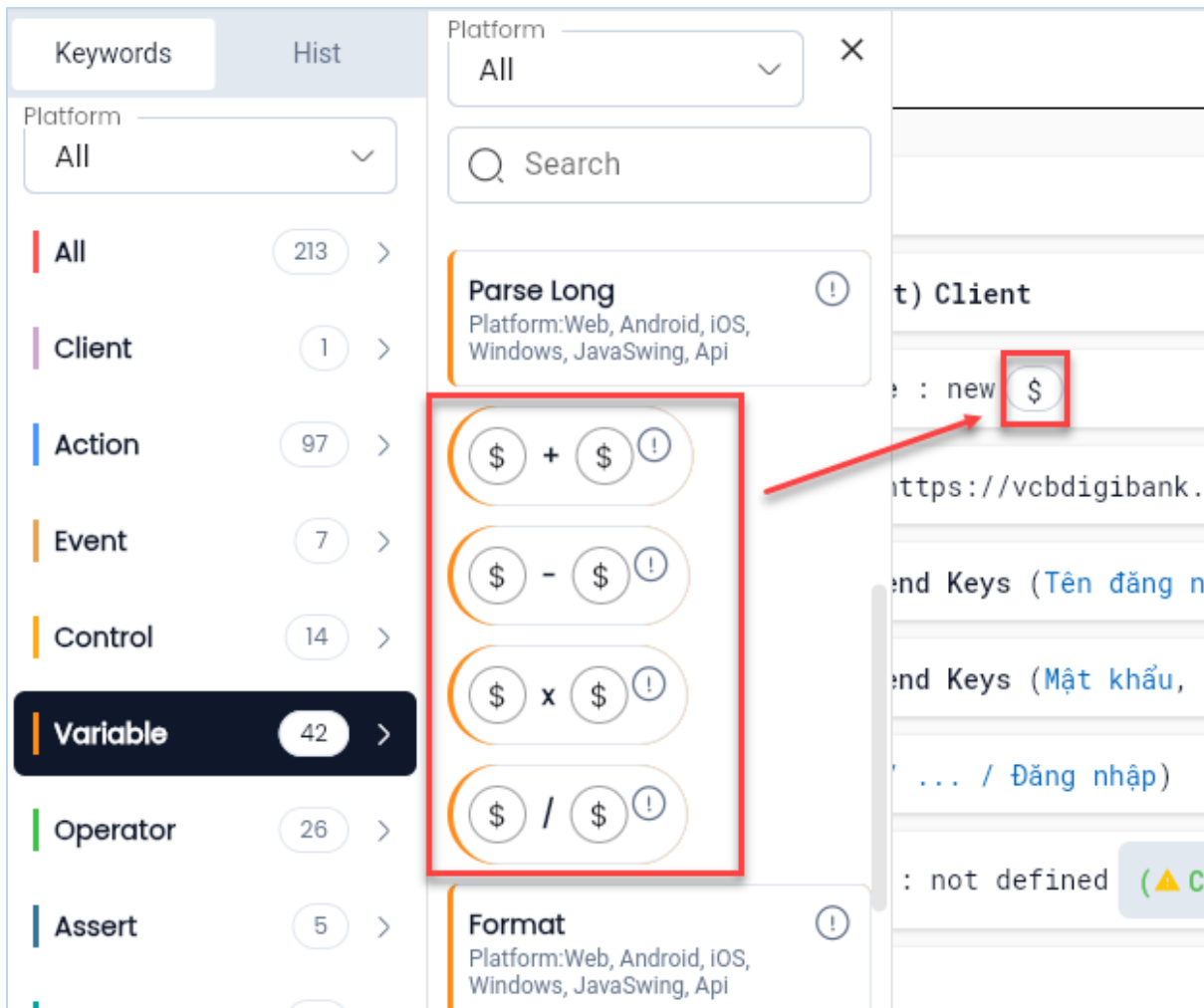
Step comment	Enter a description
Data type	Choose a data type
Name	Enter a name
isList	False
Variable init	False

Trong đó:

- Steps comment: Nhập ghi chú cho bước thao tác của keyword, giúp kịch bản kiểm thử được hiểu một cách rõ ràng hơn
- Data Type: Lựa chọn data type chính xác, phù hợp với mục đích sử dụng của tham số
- Name: Nhập tên của biến – là tên chưa được khai báo trong kịch bản kiểm thử ở bất kỳ loại tham số nào
- isList: false
- Variable init: false

Click **Done**

Bước 3: Nếu variable là giá trị của các phép toán tử (cộng “ + ”, trừ “ – ”, nhân “ x ”, chia “ : ”) hoặc ghép chuỗi string, thực hiện kéo keyword toán tử thuộc variable vào vị trí \$ của keyword



Trong đó:

- Nếu là các phép tính giá trị, bắt buộc kiểu dữ liệu của variable lựa chọn ở bước 2 phải là integer/ long/ double/ float/ date/ datetime
- Nếu là phép cộng chuỗi string, kiểu dữ liệu lựa chọn ở bước 2 là string

Bước 4: Nhập giá trị cho variable

Click vào \$ của keyword để thực hiện nhập giá trị cho variable

Giá trị của variable có thể là Value/ variable

Thường sử dụng new variable khi cần tạo mới một tham số có giá trị là cộng chuỗi string/ kết quả của các phép tính toán tử hoặc có giá trị chưa biết (empty string – với data type là string, null – với data type là integer/ long/ double/ float/ date/ datetime)

Cách 2: Khai báo qua newVariable của thuộc tính output trong keyword

Bước 1: Trong keyword có thuộc tính output, lựa chọn Type output là newVariable

Output

Type	newVariable	×
Data type	Elements	×
Name	Enter a name	

Bước 2: nhập thông tin cho variable

Trong đó:

- Type: newVariable
- Data Type: tùy thuộc vào loại keyword mà lựa chọn data type, thường sử dụng giá trị mặc định của hệ thống trong chính keyword đó
- Name: tên của variable – là giá trị chưa tồn tại trong kịch bản kiểm thử

Bước 3: Click **Done** để hoàn thiện cấu hình keyword

1.3.3.2 Parameter

Là giá trị đầu vào (input) của kịch bản kiểm thử, nhân viên kiểm thử trực tiếp truyền data của test case vào parameter trước khi thực thi kịch bản.

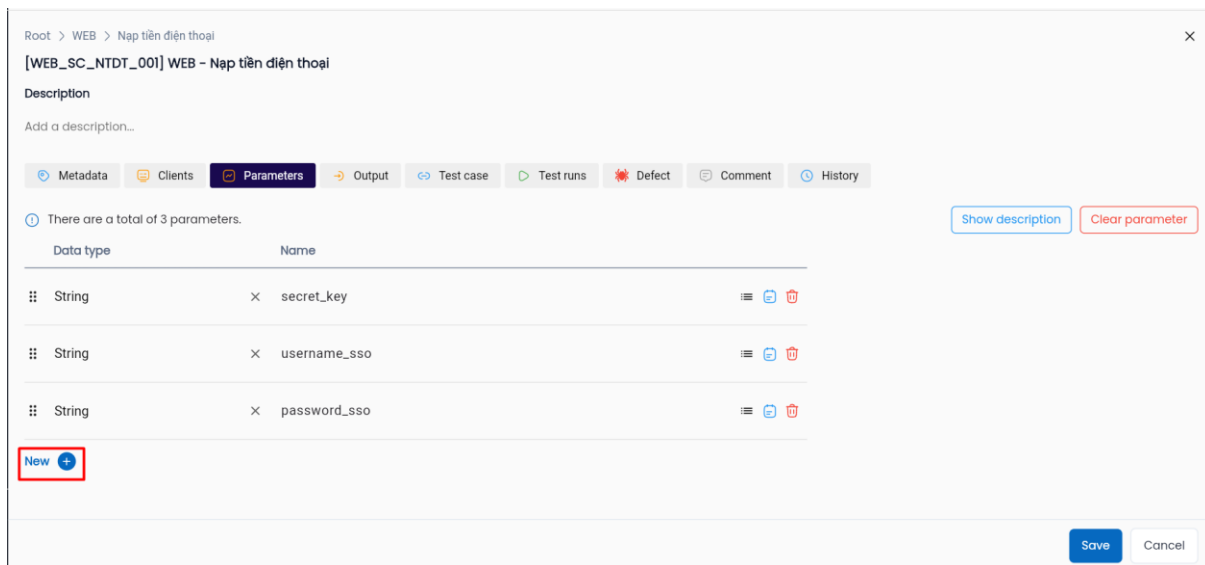
1.3.3.2.1 Đặc điểm:

- Có thể được khai báo trực tiếp trong tab parameter của metadata hoặc khai báo trong các keyword có vai trò nhập liệu (clear and send keys, send keys, select...) hoặc nhập name của parameter trong các keyword so sánh dữ liệu, hệ thống cũng sẽ tự sinh ra parameter tương ứng trong tab parameter của metadata
- Nhân viên kiểm thử trực tiếp truyền data của test case vào parameter trước khi thực thi kịch bản
- Có thể gán giá trị cho parameter bằng output của một keyword bất kỳ có thuộc tính output

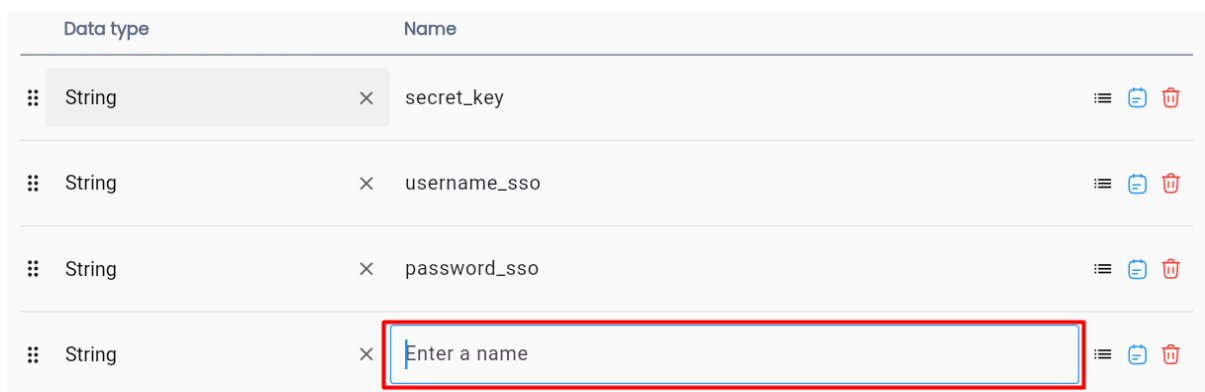
1.3.3.2.2 Cách tạo parameter

Cách 1: Khai báo trong tab parameter của metadata

Bước 1: Trong tab parameter của metadata, click New parameter



Bước 2: Nhập tên cho parameter và lựa chọn data type phù hợp với vai trò và cách sử dụng parameter của nhân viên kiểm thử trong kịch bản kiểm thử tự động



Bước 3: Click **Save** để lưu tham số

Cách 2: Hệ thống tự sinh ra parameter

Hệ thống tự sinh ra parameter khi tham số sử dụng trong các keyword có vai trò nhập liệu (có thuộc tính input) hoặc tham số xuất hiện trong variable default property của keyword và tên tham số chưa từng tồn tại trong kịch bản kiểm thử.

- Các keyword có thuộc tính input: nhập tên của tham số (chưa tồn tại), hoàn thiện cấu hình keyword và click Done

Clear And Send Keys property

×

Locator

Element

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

Repository Element

▼

Element Repository

Input

value

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

String

▼

Enter a name

Find variable

+ Add more

End Key

NONE

×

Cancel

Done

- Tham số xuất hiện trong variable default property: giá trị mặc định của tham số

Variable default property

×

Value

Variable

Version Parameter

Environment Parameter

Agent Parameter

Device Parameter

String

×

Enter a name

Find variable

+ Add more

1.3.3.2.3 Sử dụng tham số dự án trong cấu hình

Input của kịch bản kiểm thử ngoài parameter nhập vào trước khi thực thi qua parameter còn có thể truyền vào qua các tham số [device parameter](#), [environment parameter](#), [version parameter](#).

Đặc điểm:

- Các tham số dự án không xuất hiện trong danh sách variable của kịch bản kiểm thử
- Giá trị của tham số được nhập vào ở chức năng setting của project

- Khi thực thi, giá trị tham số sẽ tự động thay đổi theo môi trường (nếu sử dụng environment parameter), thiết bị (nếu sử dụng device parameter) hoặc version (nếu sử dụng version parameter)
- Khi Play keyword/ Play all hay Run thực thi thì nhân viên kiểm thử đều không cần nhập lại giá trị cho các tham số này.

Bước 1: Trong thuộc tính input/ variable default của keyword click chọn loại tham số dự án mong muốn

Bước 2: Click Choose a name, danh sách parameter của loại tham số dự án tương ứng được xổ ra

Bước 3: Tìm, chọn tham số cần sử dụng

Bước 4: Click **Done**

Lưu ý: Với **device parameter**, hệ thống sẽ nhận tham số thiết bị khi start debug, nếu giá trị của device parameter được nhập sau khi start debug thì giá trị tham số sẽ không được cập nhật khi cấu hình.

1.3.3.3 Output

Là đầu ra, kết quả trả về của kịch bản kiểm thử tự động

1.3.3.3.1 Đặc điểm:

- Có thể được khai báo trực tiếp trong tab output của metadata hoặc khai báo trong các keyword có thuộc tính output (get text, get attribute, parse long...), thậm chí khi sử dụng keyword Assign value, nếu tên của tham số chưa tồn tại trong kịch bản kiểm thử, hệ thống cũng sẽ tự sinh ra output tương ứng trong tab output của metadata
- Output xuất hiện trong kết quả của kịch bản kiểm thử tại test run output, thể hiện kết quả thực tế khi thực thi kiểm thử

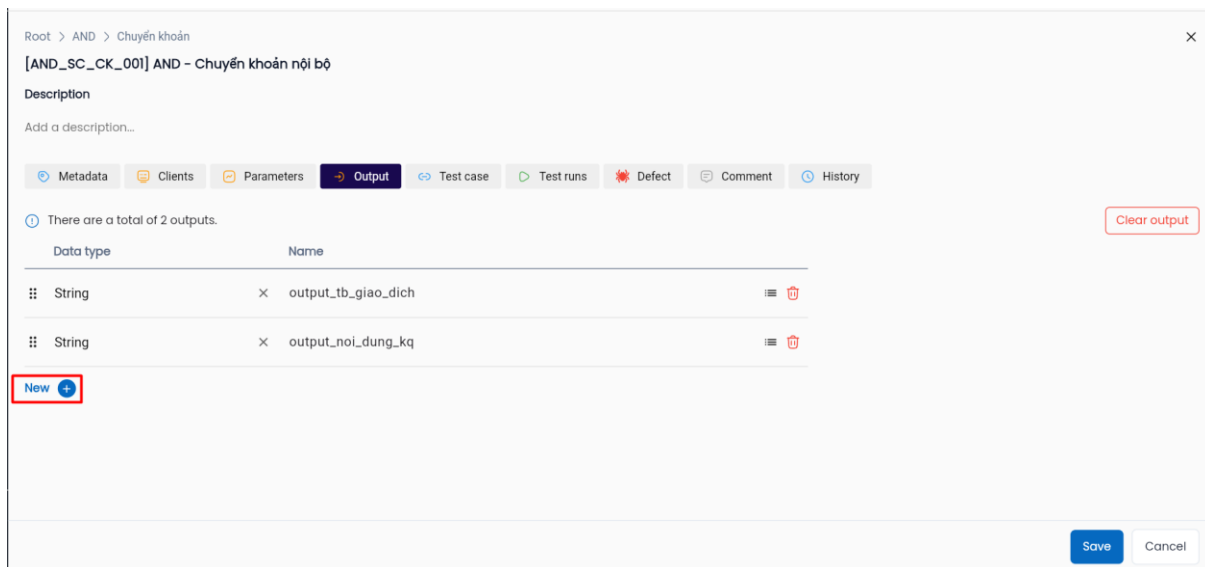
The screenshot displays the 'Data 1 of [AND_TC_CT_002] AND - Chuyển tiền nội bộ cùng chủ' interface. It features three tabs: 'Information', 'Log message', and 'Output File'. Below the tabs, there are three main sections: 'TEST RUN RESULT', 'TEST RUN INFORMATION', and 'TEST RUN DATA'. The 'TEST RUN RESULT' section shows 'Expected result', 'Actual result', and 'Final status', all marked as 'Pass'. The 'TEST RUN INFORMATION' section lists details like Agent (macbook_agemt_public), Start run (10-09-2025 08:55:20), End run (10-09-2025 08:56:24), Run time (44,99 (s)), Version (AND_V.5.2.84), and Environment (SIT). The 'TEST RUN DATA' section shows Task group name (group_nhung), Test case code (AND_TC_CT_002), Test Run Expected Result (Pass), Severity (Critical), and Priority (Medium). At the bottom, the 'Test run output' section is highlighted with a red box, showing a list of output items: output_tb_giao_dich: Giao dịch thành công, output_so_tham_chieu_kq: 039301E1-833EVZPa, output_ngan_hang_nhan:, output_tk_nguon: 8820113001, output_so_du: 205,168,154, output_so_du_double: 205168154, and output_ten_nguoi_nhan: NGUYEN THI THANH HOAN.

- Output của shared steps, script có thể được gọi ra làm input cho một keyword nào đó của script/ suite sử dụng shared steps/ script

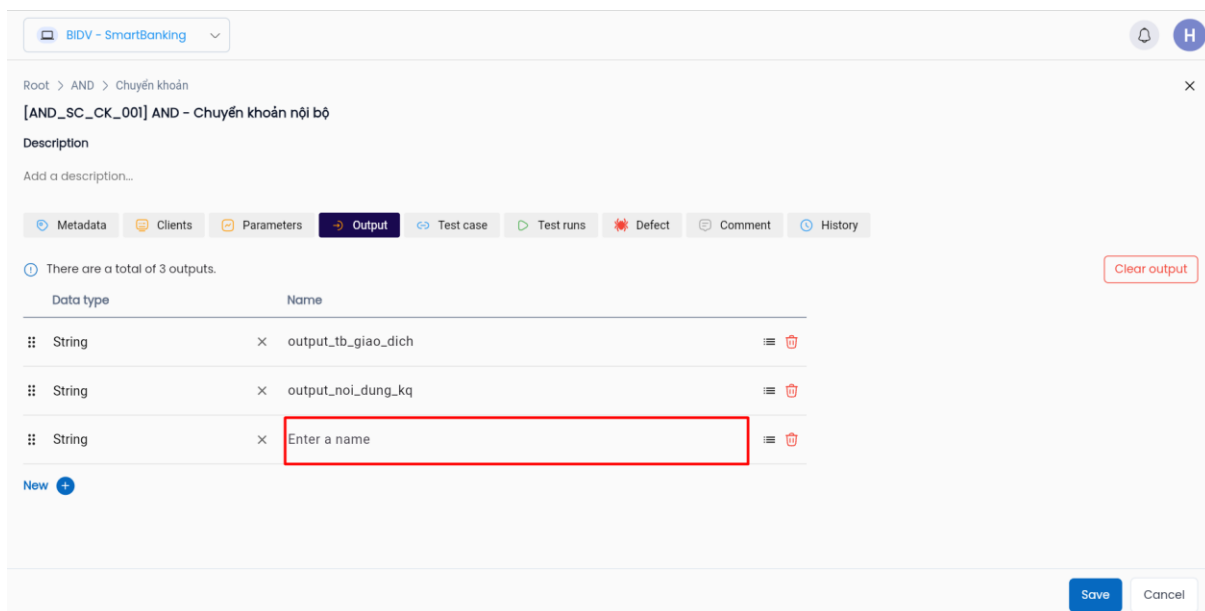
1.3.3.3.2 Cách tạo output của kịch bản kiểm thử

Cách 1: Tạo trực tiếp từ tab output trong metadata

Bước 1: Trên tab output của metadata, click New output



Bước 2: Nhập tên và lựa chọn kiểu dữ liệu của output phù hợp với vai trò và cách sử dụng output trong kịch bản của nhân viên kiểm thử. Tên output là duy nhất, chưa từng tồn tại trong kịch bản kiểm thử



Bước 3: Click *Save*

Cách 2: Tạo tự động từ thuộc tính output của keyword

Bước 1: Trong keyword có thuộc tính output, lựa chọn Type output là variable.

Output

Type	variable	X
Data type	Call Output	X
Name	Enter a name	

Find variable

Cancel Done

Bước 2: Nhập thông tin cho variable

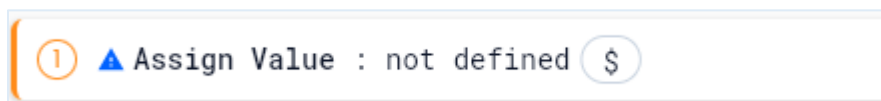
Trong đó:

- Type: Variable
- Data Type: tùy thuộc vào loại keyword mà lựa chọn data type, thường sử dụng giá trị mặc định của hệ thống trong chính keyword đó
- Name: tên của variable – là giá trị chưa tồn tại trong kịch bản kiểm thử

Bước 3: Click Done để hoàn thiện cấu hình keyword

Cách 3: Tạo tự động từ keyword assign value

Bước 1: Trên màn hình config kéo thả keyword assign value



Ta được biết, **assign value** là keyword dùng để gán giá trị cho một tham số, trong trường hợp tham số chưa tồn tại trong list variable của kịch bản kiểm thử, hệ thống tạo tự động một output tương ứng.

Bước 2: Click not defined để khai báo output

Assign Value property

×

Step comment	Enter a description
Data type	Choose a data type ▾
Name	Enter a name
isList	False ×
Variable init	False ×

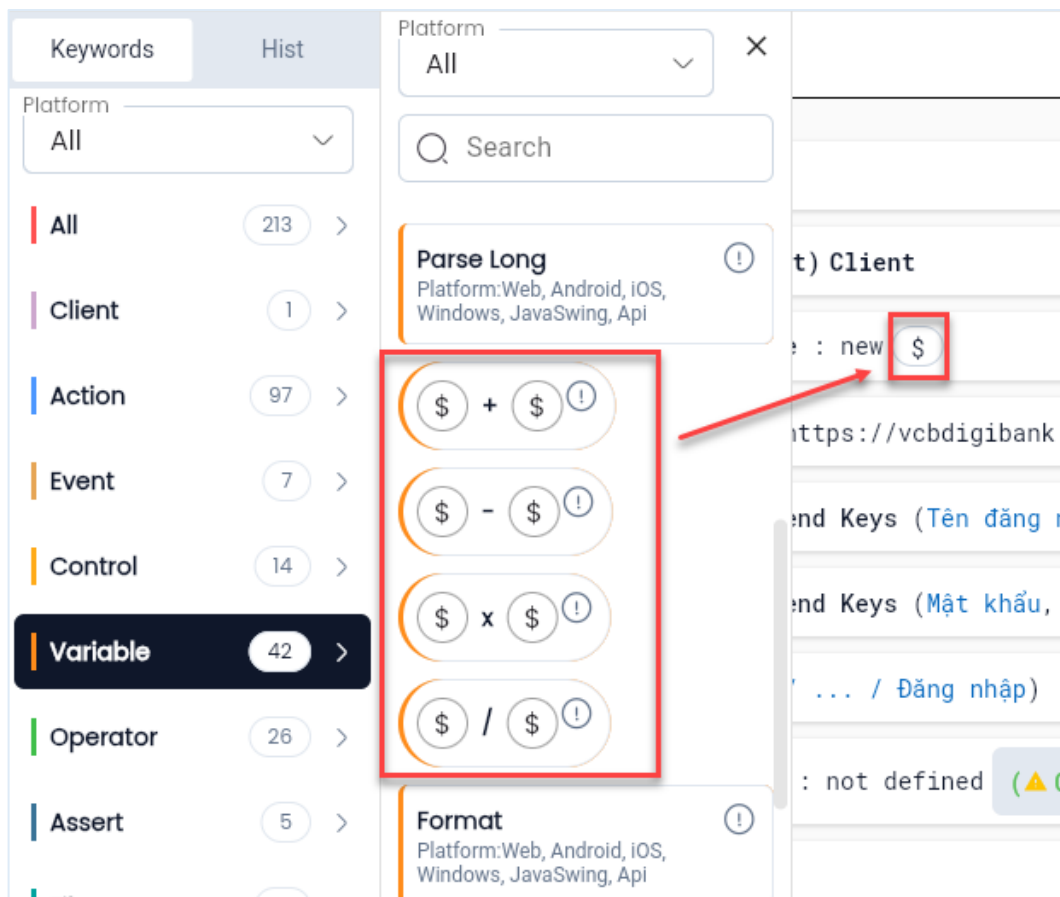
Bước 3: Nhập thông tin của output

Trong đó:

- Steps comment: Nhập ghi chú cho bước thao tác của keyword, giúp kịch bản kiểm thử được hiểu một cách rõ ràng hơn
- Data Type: Lựa chọn data type chính xác, phù hợp với mục đích sử dụng của tham số
- Name: Nhập tên của output – là tên chưa được khai báo trong kịch bản kiểm thử ở bất kỳ loại tham số nào
- isList: False
- Variable init: False

Click **Done**

Bước 4: Nếu output là giá trị của các phép toán tử (cộng “ + ”, trừ “ – “, nhân “ x “, chia “ : “) hoặc ghép chuỗi string, thực hiện kéo keyword toán tử thuộc variable vào vị trí \$ của keyword



Trong đó:

- Nếu là các phép tính giá trị, bắt buộc kiểu dữ liệu của output lựa chọn ở bước 2 phải là integer/ long/ double/ float/ date/ datetime
- Nếu là phép cộng chuỗi string, kiểu dữ liệu lựa chọn ở bước 2 là string

Bước 5: Gán giá trị cho output

Click vào \$ của keyword để thực hiện nhập giá trị cho output

Giá trị của output có thể là Value/ variable

Thường sử dụng assign value khi cần tạo mới một output và có giá trị là cộng chuỗi string/ kết quả của các phép tính toán tử hoặc có giá trị chưa biết (empty string – với data type là string, null – với data type là integer/ long/ double/ float/ date/ datetime)

1.3.4 Cấu hình kiểm thử tự động

Sau khi phân tích kịch bản kiểm thử, thực hiện cấu hình theo bài toán của luồng yêu cầu.

- Nếu bài toán là một bước/ nhóm bước: cấu hình shared steps
- Nếu bài toán là một luồng chức năng hoàn chỉnh: cấu hình script
- Nếu bài toán là một bộ kiểm thử bao một hoặc nhiều chức năng hoàn chỉnh được sắp xếp theo một thứ tự hoặc lựa chọn thực thi theo một điều kiện nào đó: cấu hình test suite

Kịch bản kiểm thử tự động được xây dựng dựa trên các steps đã phân tích trước đó, việc cấu hình kịch bản kiểm thử là lặp đi lặp lại các bước dưới đây cho mỗi lần truy cập vào script.

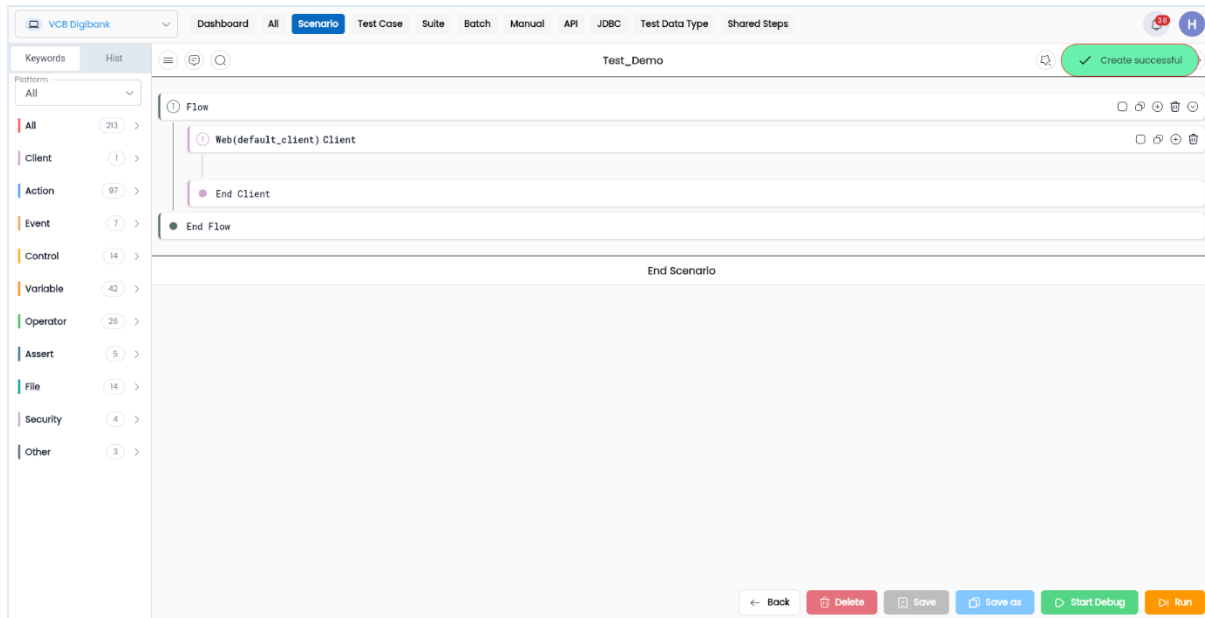
Các bước thực hiện cấu hình:

1. Xác định client cho Script/ Suite/ Shared steps
2. Start debug – không bắt buộc nếu nhân viên kiểm thử đã xác định được các phần tử cần xuất hiện trong kịch bản kiểm thử, tuy nhiên cần start debug để sử dụng các tính năng hỗ trợ của hệ thống, đảm bảo script được cấu hình đúng, đáp ứng được bài toán của kịch bản.
3. Kéo thả keyword phù hợp và cấu hình keyword để có thể đáp ứng được bài toán kiểm thử tự động đồng thời xác định các tham số của kịch bản kiểm thử.
4. Play keyword, Run debug all để kiểm tra tính đúng đắn của script/ suite/ shared steps

Xây dựng script/ suite/ shared steps về cơ bản thao tác giống nhau, ở đây chúng ta sẽ cụ thể đối với script.

1.3.4.1 Xác định client

Sau khi tạo một script, hệ thống mặc định tạo một client với platform là Web



Client là keyword thể hiện nền tảng mà ứng dụng/ hệ thống trong script/ suite thực hiện kiểm thử.

Cần thực hiện xác định client trước khi start debug để có thể lựa chọn profile, device đúng với yêu cầu của script.

Để thực thi câu lệnh, các keyword đều phải được đặt trong một client cụ thể.

1.3.4.1.1 Tạo mới client

Thông thường, chỉ cần tạo mới client khi kịch bản kiểm thử được thực thi trên nhiều hơn 1 client (có thể một platform hoặc nhiều hơn), kiểm thử trên đa môi trường, đa thiết bị.

Cách 1: Thêm mới client trong tab client của metadata

Add client

×

Platform

Choose a platform

▼

Name

Save

Cancel

Bước 3: Nhập thông tin cho client, trong đó:

- Platform: nền tảng của client
- Name: tên client, là duy nhất, không chứa ký tự đặc biệt

Bước 4: Click **Save**

Client property

×

Client

ios-ios_client

×

✎

🗑

+

Step comment

Enter a description

Cancel

Done

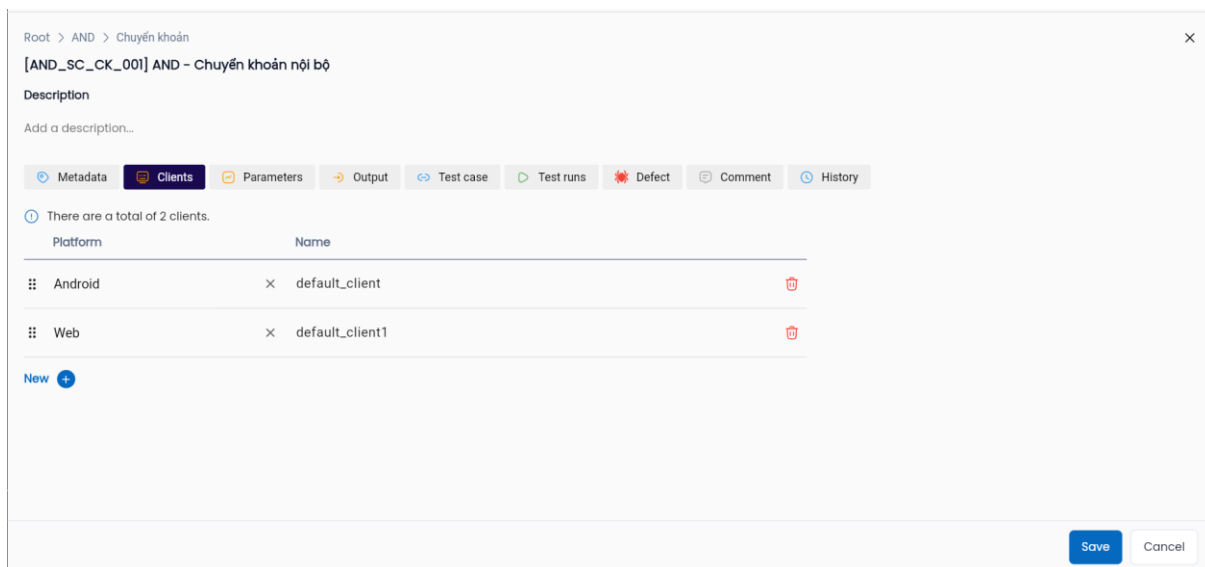
Bước 5: Click **Done**

1.3.4.1.2 Update client

Đôi khi client trong kịch bản kiểm thử khi khởi tạo không đúng nền tảng đang kiểm thử hoặc cần thay đổi tên, nhân viên kiểm thử cần thực hiện update thông tin cho client tương ứng.

Cách 1: Chỉnh sửa thông tin từ tab Client trong metadata

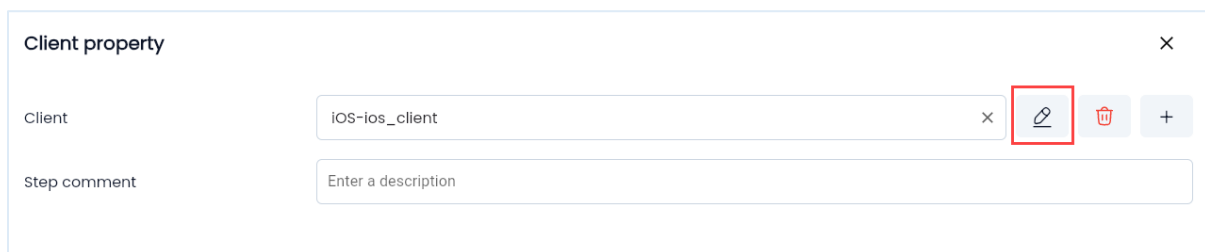
Bước 1: Trên tab Client của metadata, thực hiện chỉnh sửa client bằng cách chọn lại platform, sửa tên client (nếu cần) tương ứng



Bước 2: Click **Save**

Cách 2: Chỉnh sửa trực tiếp trên keyword client

Bước 1: Click vào keyword cần chỉnh sửa client



Bước 2: Click button Edit, hệ thống hiển thị popup edit client

Edit client

×

Platform

ios

×

Name

iso_client

Save

Cancel

Bước 3: Thực hiện chọn lại platform/ sửa tên client (nếu cần)

Bước 4: Click **Save**

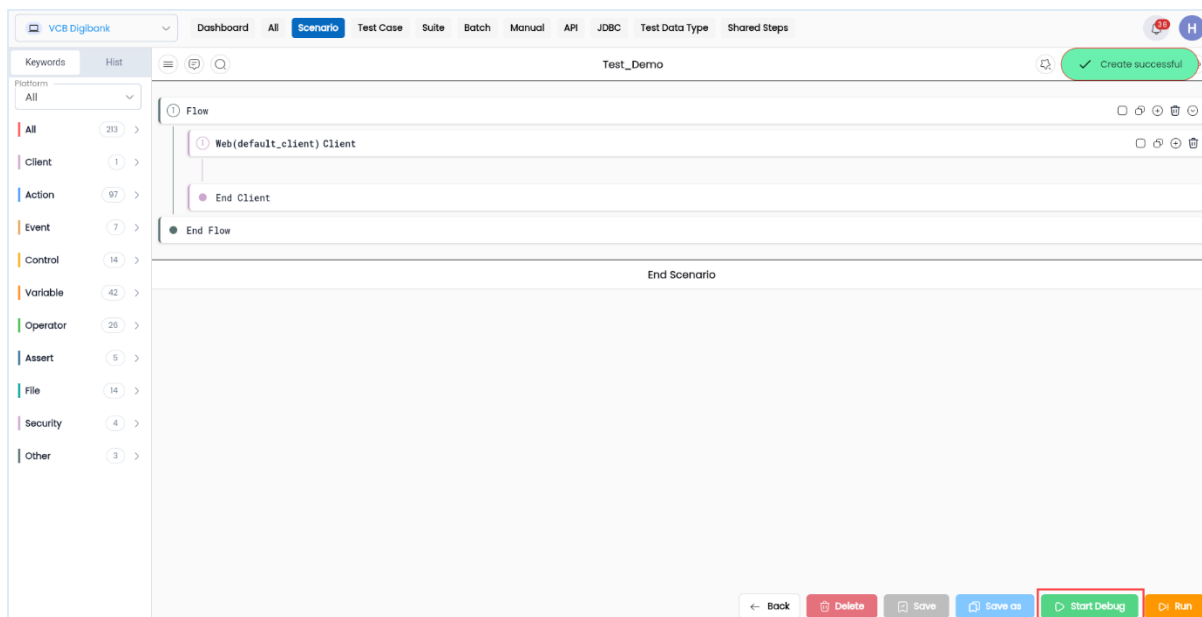
Bước 5: Click **Done**

1.3.4.2 Start debug

Để start debug, nhân viên kiểm thử cần chắc chắn:

- Đã start [agent](#) thành công với nền tảng lựa chọn trong client của script
- Test project đã tạo [profile](#) tương ứng với platform sử dụng trong script

Bước 1: Trên màn hình config, click Start debug



Bước 2: Lựa chọn environment, version, client, Device tương ứng với client (nếu có), agent thực hiện test config

Start Debug

×

Select a profile and version to start debug

SUT Version

ios_V6.0.4

×

SUT Environment

PRJ-UAT

×

Select profile and agent for the client

Client default_client

Web

▼

Client iso_client

ios

▼

Device

Any

▼

Agent type ⓘ

Agent type

Any agent

▼

Start Debug

Bước 3: Click Start debug

Sau khi click start debug thành công, hiển thị button Stop debug và View screenshot ở khối button thao tác của màn hình low – code config



Lưu ý: Đối với kiểm thử ứng dụng trên Mobile/ Window app, nhân viên kiểm thử cần thực hiện [Add device](#) vào Project để có thể Start debug/ Run trên thiết bị đó

1.3.4.3 Cấu hình Keyword

1.3.4.3.1 Truy cập môi trường kiểm thử

Thông thường, để bắt đầu một script cần truy cập vào môi trường kiểm thử.

- **Môi trường Web:** Sử dụng keyword Go to URL

Go To Url property [X]

Step comment:

Input

url

Value Variable Agent Parameter Device Parameter Version Parameter Environment Parameter

String

Find value

Cancel Done

Bước 1: Click vào keyword, hệ thống hiển thị màn hình **Go To Url property**

Bước 2: Nhập thông tin cho keyword, trong đó:

- Steps comment: nhập mô tả cho bước thực hiện, ở đây là truy cập trang web được kiểm thử
- url: địa chỉ trang web được kiểm thử, thường có type là Value. Trong trường hợp url thay đổi theo môi trường phát triển/ version, hệ thống khuyến nghị sử dụng environment parameter/ version parameters cho tham số url

Bước 3: Click **Done** để hoàn thiện cấu hình keyword

- **Môi trường Mobile:** Sử dụng keyword Restart Application

Restart Application property

Step comment: Enter a description

Input

applid

Value Variable Agent Parameter Device Parameter Version Parameter Environment Parameter

String Enter a value

Find value

Cancel Done

Bước 1: Click vào keyword, hệ thống hiển thị màn hình Restart Application property

Bước 2: Nhập thông tin cho keyword, trong đó:

- Steps comment: nhập mô tả cho bước thực hiện, ở đây là khởi động lại ứng dụng được kiểm thử
- appId: id của ứng dụng từ nhà phát triển, giúp hệ thống có thể start đúng ứng dụng trên thiết bị

Bước 3: Click **Done** để hoàn thiện cấu hình keyword

1.3.4.3.2 Sử dụng keyword trong script

Để sử dụng đúng, hiệu quả keyword trong cấu hình kiểm thử tự động, nhân viên kiểm thử cần xác định được từ khóa, steps auto ở từng bước, từ đó lựa chọn đúng keyword.

Các loại keyword bao gồm: Client, Action, Control, Variable, Operator, Assert, File & others

- Action: Các keyword dạng Action trong kịch bản kiểm thử tự động được sử dụng để thực hiện một hành động cụ thể trên giao diện hoặc hệ thống/ứng dụng đang được kiểm thử.
- Control: keyword “điều khiển” các hành động ở trong nó thực hiện theo một/ một số điều kiện nào đó
- Variable: keyword thực hiện hành động lên các tham số
- Assert: keyword khẳng định một kết luận dựa trên một/ một số điều kiện nào đó (tương đương If – else kết hợp Error)
- Operator: keyword thể hiện điều kiện nào đó, không thể dùng độc lập mà chỉ có thể kết hợp sử dụng với control hoặc assert
- File: là các keywords được thiết kế để thao tác hoặc kiểm tra các tập tin (files) trong hệ thống.
- Client and other: các keywords khác
- Security: keywords bảo mật trong hệ thống
- Event: keywords các sự kiện trong hệ thống

Tham chiếu Document keyword để xác định được cách sử dụng của từng keyword có trong hệ thống.

Lựa chọn, kéo thả các keyword và sắp xếp theo thứ tự phù hợp với trình tự kiểm thử của kịch bản.

Cấu hình keyword theo các thuộc tính có trong keyword đó.

Bước 1: Xác định phần tử cần tương tác của keyword (nếu có thuộc tính locator)

Bước 2: Xác định tham số của keyword (nếu có thuộc tính input/ output/ variable default)

Ví dụ, cấu hình keyword Clear and send keys trong script đăng nhập

Clear And Send Keys property



Step comment

Enter a description

Locator

Element

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

Repository Element

Element Repository

Input

value

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

String

Enter a name

Find variable

+ Add more

End Key

Cancel

Done

Bước 1: Nhập steps comment của keyword giúp kích bản kiểm thử được hiểu một cách rõ ràng hơn, ví dụ: Nhập dữ liệu vào trường thông tin username trên màn hình đăng nhập

Bước 2: Nếu đã có element trong repository, click element repository để lựa chọn element cho keyword

Bước 3: Dùng Find variable để tìm tham số đã được khai báo chứa thông tin username hoặc trực tiếp nhập tên tham số username (nếu chưa có)

Bước 4: Click Done

Clear And Send Keys property

Step comment

Nhập username vào trường username

Locator

Element

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

Repository Element

web-login / Tên đăng nhập

Element Repository

Input

value

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

String

username

Find variable

+ Add more

End Key

Cancel

Done

Sau khi click **Done**, trên màn hình config, keyword vừa cấu hình hiển thị thông tin xpath (tùy thuộc vào việc sử dụng repository element hay find element/ viết xpath) và tham số sử dụng trong keyword.

2

Clear And Send Keys (web-login / Tên đăng nhập, username)
Nhập username vào trường username

Sử dụng các phương pháp [xác định phần tử](#) và [tham số](#) của kịch bản kiểm thử để cấu hình keyword, ta đã có thể xây dựng một kịch bản kiểm thử tự động hoàn chỉnh.

chuyen_khoan_ngoi_vcb

Flow

Select

1 iOS(default_client) Client

1 New Variable : count 0

2 While ((count <= so_lan_nhap_ptxt) AND

+

)

1 Assign Value : count count + 1

+

2 Send Keys Without Element

3 Click

4 If ((

\$

 ==

\$

) AND

+

) then

1 Assert Equals : not defined

End If

End While

3 Start Application (com.vcb.VCB)

4 Clear And Send Keys (Mật khẩu, password)

5 Click (loginBtnNext - Đăng nhập)

6 Action Call : [LOGIN] Login (nhap_mat_khau)
onFinally

7 Action Call : [IOS_ST_015] IOS-ST-Chuyển tiền trong nước-Đi đến màn hình khởi tạo
onFinally

8 Click (ios / ... / Mẫu chuyển tiền - Mẫu chuyển tiền - Mẫu chuyển tiền)

9 Sleep 4000

10 Click (//XCUIElementTypeWindow/XCUIElementTypeOther[@visible='true']//XCUIElementTypeScrollView/XCUIElementTypeStaticText[@name='Chuyển tiền nhanh 24/7 ngoài VCB'])

Click (
//XCUIElementTypeWindow/XCUIElementTypeOther[@visible='true']//XCUIElementTypeStaticText[@name='Ngân hàng thụ hưởng']/parent::XCUIElementTypeOther/
XCUIElementTypeButton
)

12 Clear And Send Keys (//XCUIElementTypeTextField[@value='Tìm kiếm'], bank_name)

13 Click (//XCUIElementTypeTable/XCUIElementTypeStaticText)

14 Clear And Send Keys (//XCUIElementTypeTextField[contains(@value,'chọn tài khoản nhận')], bank_account)

15 Clear And Send Keys (//XCUIElementTypeTextField[contains(@value,'Số tiền')], amount)

16 Click (//XCUIElementTypeButton[@name='Tiếp tục'])

17 Click (//XCUIElementTypeButton[@name='Xác nhận'])

18 Send Keys Without Element (pin)

19 Click (//XCUIElementTypeButton[@name='Tiếp tục'])

20 Click (//XCUIElementTypeButton[@name='Xác nhận'])

21 Sleep 15000

If

22 (Check Element Exists (//XCUIElementTypeWindow/XCUIElementTypeOther[@visible='true']//XCUIElementTypeStaticText[@name='CHUYỂN KHOẢN THÀNH CÔNG']) AND

+

)
then

1 ma_giao_dich = Get Text (ma_giao_dich_element)

Else

1 Error (Không thành công)

End If

23 Click (
//XCUIElementTypeStaticText[@name='Ngân hàng nhận']/parent::XCUIElementTypeOther/following-sibling::XCUIElementTypeImage/parent::XCUIElementTypeOther/preceding-
sibling::XCUIElementTypeButton
)

24 Click (Ngân hàng nhận - Ngân hàng nhận - Ngân hàng nhận)

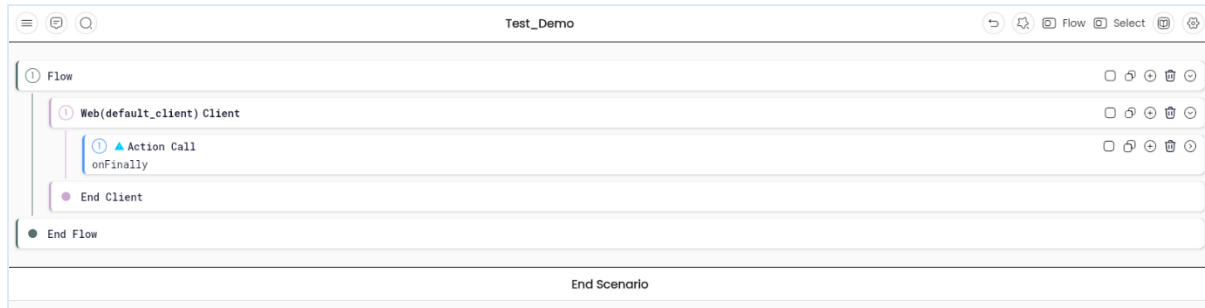
End Client

End Scenario

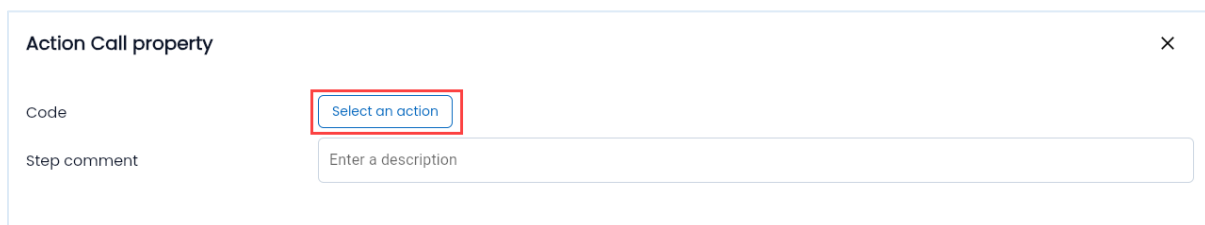
1.3.4.4 Sử dụng shared steps trong script

Để sử dụng shared steps trong script, shared steps cần được set Workflow status là Active.

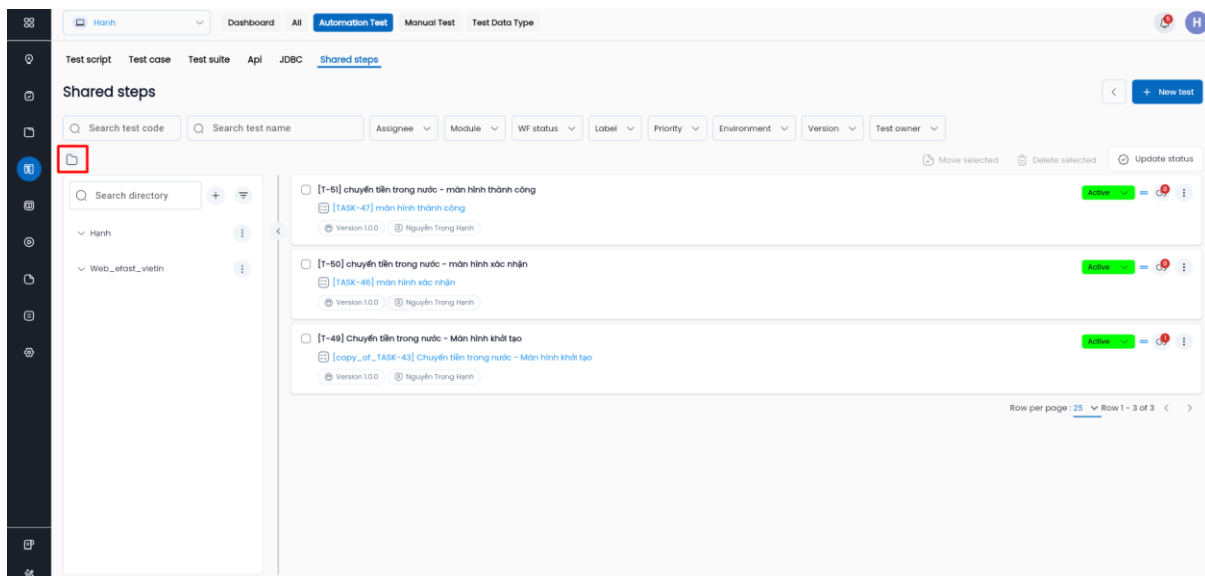
Bước 1: Trên màn hình config kéo thả keyword Action call vào vị trí cần sử dụng.



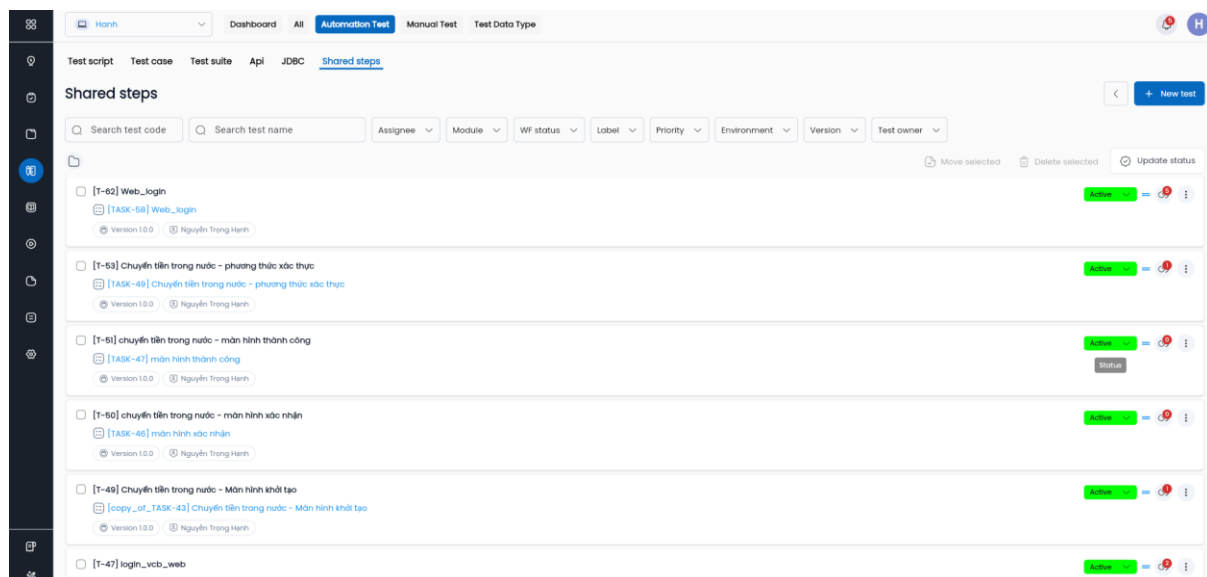
Bước 2: Click vào keyword Action call, hệ thống hiển thị màn hình Action call property, click Select an action



Nhấn vào biểu tượng **Directory** để hiển thị toàn bộ các Shared steps:



Bước 3: Click chọn shared steps cần sử dụng



Hệ thống hiển thị màn hình **Action call property** cùng với các thông tin của shared steps vừa chọn bao gồm code, name, client list, parameter...

Action Call property

Code

LOGIN_WEB

Select an action

Name

Login

Step comment

Enter a description

Client list *

default_client

web_client

iso_client

iso_client

Input

ten_dang_nhap

Value

Variable

Version Parameter

Environment Parameter

Agent Parameter

String

ten_dang_nhap

Find variable

+ Add more

mat_khau

Value

Variable

Version Parameter

Environment Parameter

Agent Parameter

String

mat_khau

Cancel

Done

Bước 4: Nhập Steps comment cho action call

Bước 5: Lựa chọn lại client trong Client list cho Action Call

Hệ thống mặc định map client default của script cho shared steps, trong trường hợp script có nhiều hơn 01 client cần lựa chọn client phù hợp với client của shared steps.

The screenshot shows the 'Action Call property' dialog box. The 'Client list' dropdown is open, showing 'web_client' selected and highlighted with a red box. Below the dropdown, the 'Find variable' button is also highlighted with a red box. The dialog box includes fields for Code (LOGIN_WEB), Name (Login), Step comment (Enter a description), and a search bar. The 'Find variable' button is located below the search bar.

Bước 6: Map parameter của shared steps với các tham số đã có của script có giá trị, vai trò sử dụng tương ứng (bao gồm variable, parameter và output). Trong trường hợp các tham số với giá trị sử dụng tương ứng chưa xuất hiện, hệ thống mặc định tạo parameter với name là tên tham số của shared steps.

Cách map tham số của parameter với một tham số đã có trong script

1. Click **Find variable** tương ứng với parameter cần map

The screenshot shows the 'Input' dialog box. The 'Find variable' button is highlighted with a red box. The dialog box includes tabs for Value, Variable, Version Parameter, Environment Parameter, and Agent Parameter. The 'Variable' tab is selected, and the 'String' dropdown is open, showing 'mat_khau' selected. The 'Find variable' button is located below the dropdown.

2. Click chọn tham số có giá trị sử dụng tương ứng với parameter của shared steps

Find variable

Variable name

String : ten_dang_nhap

String : mat_khau

Tham số đã được map tương ứng giữa shared steps và script

Action Call property

Step comment

Enter a description

Client list *

default_client

default_client

Input

mat_khau

Value

Variable

Version Parameter

Environment Parameter

Agent Parameter

String

matkhau

Find variable

+ Add more

Bước 7: Call output của shared steps (nếu cần)

Output

Type

none

Cancel

Done

Thuộc tính output của Action call mặc định là none – không thực hiện gọi output, để call output ta thực hiện:

1. Lựa chọn type – newVariable

Output

Type

newVariable

Data type

Call Output

Name

Enter a name

Cancel

Done

2. Data Type mặc định là Call Output
3. Nhập tên cho tham số, tham số này sẽ là tập hợp toàn bộ các output có trong shared steps, nên đặt tên là output của shared steps tương ứng, ví dụ out_put_xac_nhan_giao_dich

Click **Done** để hoàn thành cấu hình Action call

1.3.4.5 Sử dụng output của shared steps trong script

Để sử dụng được output của shared steps trong script, Action call gọi đến shared steps đó **phải được call output**.

- **Gán giá trị cho tham số là một output của shared steps**

Bước 1: Trên keyword cần lấy input/ variable default là output của shared steps, nhấn **Find variable**

Input

mau_chuyen_tien_hint

Value Variable Version Parameter Environment Parameter Agent Parameter

Call Output

Find variable

+ Add more

Bước 2: Lựa chọn output của shared steps tương ứng (data type là call output)

Find variable

Variable name

> CallOutput : chuyen_tien_hint

String : mau_chuyen_tien_hint

String : test

String : newOutput

Thông tin tham số được hiển thị cùng data type Call output của shared steps

Input

mau_chuyen_tien_hint

Value **Variable** Version Parameter Environment Parameter Agent Parameter

Call Output

Find variable

+ Add more

Bước 3: Nhấn **Add more**

Bước 4: Nhập tên output cần lấy trong shared steps đồng thời chọn data type đúng với output đó

Input

mau_chuyen_tien_hint

Value **Variable** Version Parameter Environment Parameter Agent Parameter

Call Output

Find variable

Any

+ Add more

Nhấn **Done** để hoàn thành. Hệ thống hiển thị *call output* của shared steps cùng với tên output đã nhập:

```
var chuyen_tien_hint = Action Call : [WEB_ST_054] WEB-ST-Kiểm tra hiển thị thông tin (chuyen_tien_hint.  
example_output)  
onFinally
```

- **Map tham số trong action call có giá trị là output của một shared steps khác:** thao tác tương tự

Action Call property

Client list *

default_client

Input

username

Value

Variable

Version Parameter

Environment Parameter

Agent Parameter

Call Output

▼

thanh_cong

Find variable

Any

▼

username

+ Add more

- ❖ Để gán giá trị output của script là output của shared steps, thực hiện call output như trên, tuy nhiên keyword gán giá trị cho output của script cần được đặt trong onFinally của Action call tương ứng để đảm bảo lấy được giá trị output của shared steps ngay cả khi thực thi failed tại shared steps đó.

```

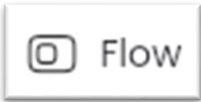
1 var chuyen_tien_hint = Action Call : [WEB_ST_054] WEB-ST-Kiểm tra hiển thị thông tin (chuyen_tien_hint.
  example_output)
onFinally
2 var new_Output = Action Call : [WEB_ST_054] WEB-ST-Kiểm tra hiển thị thông tin (mau_chuyen_tien_hint)
  onFinally
3 End Action Call

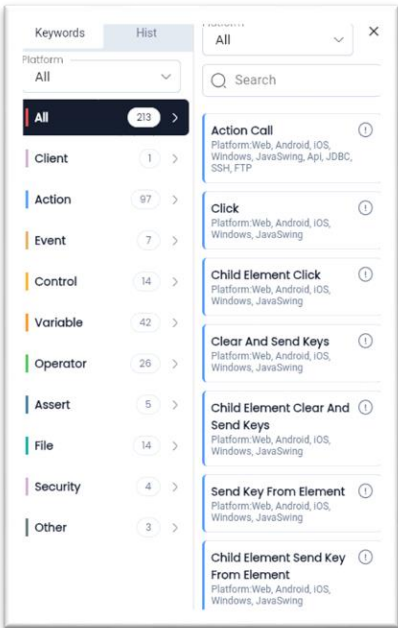
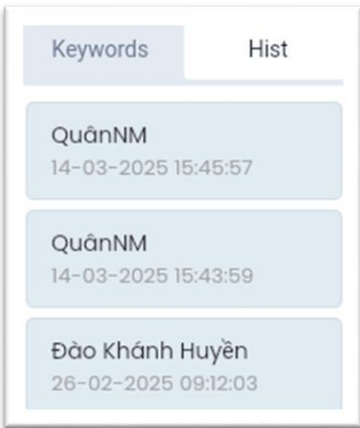
```

1.3.4.6 Các tính năng hỗ trợ trong cấu hình

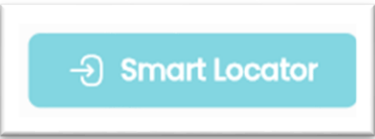
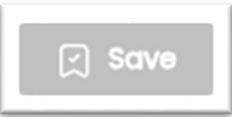
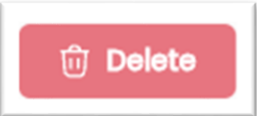
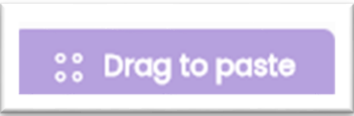
Các chức năng:

Chức năng	Mô tả
	Click vào icon này để ẩn/ hiển thị danh sách keyword
	Comment: Chức năng cho phép người dùng (bao gồm toàn bộ user được phân quyền sử dụng test script) bình luận
	Search: Chức năng cho phép người dùng tìm kiếm các keyword đã sử dụng trong test script
	Agent logs: Khi thực thi test case, LLQ Agent sẽ ghi lại toàn bộ quá trình xử lý thông qua các dòng log, kèm theo mã định danh và timestamp. Những log này

Chức năng	Mô tả
	giúp kiểm thử viên theo dõi tiến trình thực thi chi tiết, để từ đó dễ dàng kiểm tra luồng chạy và xác định lỗi nếu có.
	Play step by step: Chức năng cho phép người dùng debug toàn bộ script hoặc một nhóm các keyword tùy ý, từ đó kiểm tra tính đúng đắn về logic, thuật toán của script trước khi thực thi
	Chức năng cho phép người dùng chuyển từ hiển thị chi tiết cấu hình của script sang flow và ngược lại, giúp người review có thể xem các flow có trong script (nếu có) mà không cần xem toàn bộ cấu hình
	Chế độ basic(select: off): màn hình config hiển thị đầy đủ các tính năng hỗ trợ cấu hình thông thường Chế độ select (select: on): cho phép chọn nhiều keyword
	Chức năng cho phép người dùng chuyển sang màn hình metadata của script từ màn hình low-code
	AI conversation: Người dùng nhấn để mở ra giao diện sử dụng trí tuệ nhân tạo hỗ trợ tạo mới kịch bản kiểm thử

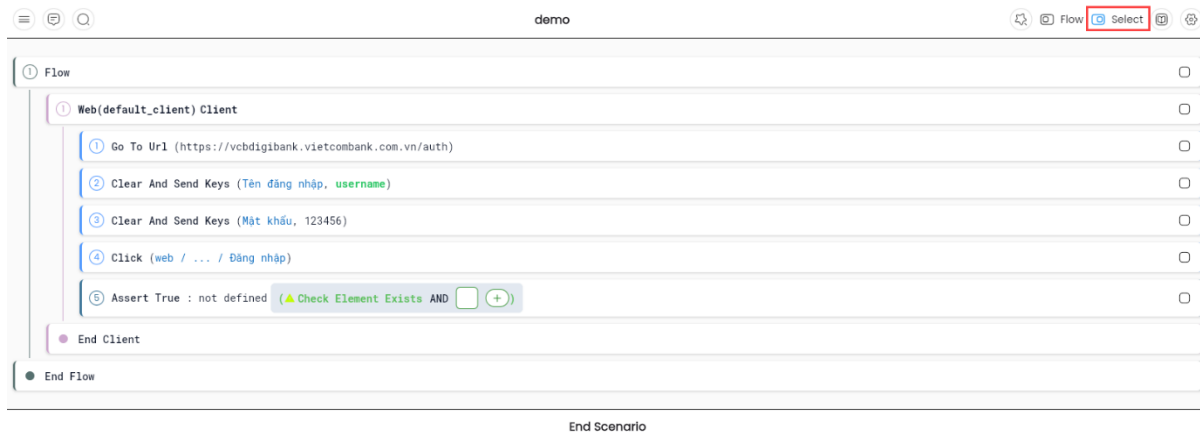
Chức năng	Mô tả
	Keyword: các hành động, điều kiện kiểm soát cho phép người dùng xây dựng thuật toán thực hiện test steps của script, được tách ra thành 6 loại keyword: Client: Nền tảng thực hiện Action: Hành động tương tác với các phần tử trên hệ thống kiểm thử Control: Giúp người dùng kiểm soát các action trong test steps theo một yêu cầu nào đó Variable: Giúp người dùng tạo và thực hiện kiểm soát variable Operator: Điều kiện để thực hiện các control Assert: Giúp người dùng đưa ra được các output mong muốn với các điều kiện cụ thể File: Các hành động tương tác với File trên hệ thống kiểm thử Security: keyword về bảo mật Other: Loại khác, không ảnh hưởng đến steps của script
	History: Chức năng cho phép người dùng xem lại toàn bộ lịch sử chỉnh sửa cấu hình của test script, từ đó có thể tái sử dụng một số câu lệnh đã bị xóa/ chỉnh sửa trước đó
	Play action: Chức năng cho phép người dùng thực hiện điều hướng trình duyệt Agent bằng cách thực hiện từng action của script, từ đó kiểm tra được tính đúng đắn của action Chỉ hiển thị khi đã start debug

Chức năng	Mô tả
	Disable/Enable: Chức năng cho phép người dùng thực hiện thao tác vô hiệu hóa/cho phép dòng cấu hình liệu có được thao tác trong lúc thực thi kịch bản kiểm thử hay không? Button có tác dụng trên tất cả các keyword trên hệ thống.
	Copy: Chức năng cho phép người dùng sao chép 1 flow, group, control, action... trong cùng script hoặc script khác (thao tác trên cùng 01 tab của trình duyệt) từ một phiên bản history bất kỳ của script đó.
	Duplicate: Chức năng cho phép người dùng nhân bản 1 key word xuống vị trí ngay phía dưới key word đã chọn. Thông tin cấu hình phải lựa chọn lại. Button có tác dụng trên tất cả các keyword trên hệ thống.
	Delete: Chức năng cho phép người dùng xóa 1 flow, group, control, action... của script
	Record: Tại keyword Group, hệ thống hỗ trợ người dùng tạo ra 1 dòng low-code bên trong group với chính xác những thông tin người dùng đã cấu hình tại màn hình View Screenshot.
	Add Multiple: Tại keyword Group, hệ thống hỗ trợ người dùng tạo ra hàng loạt các key word giống nhau với các Locator được sử dụng theo danh sách từ Element Repository hoặc Find Element. Thứ tự các keyword được tạo ra theo thứ tự lựa chọn các Element của người dùng.
	Collapse: Chức năng cho phép thu gọn script theo flow, client, group, control...

Chức năng	Mô tả
	Expand: Chức năng cho phép người dùng hiển thị toàn bộ action, control đã bị thu gọn trước đó của script
	Start Debug: Chức năng bắt đầu cho phép QA platform điều khiển trình duyệt qua Agent.
	Hiển thị sau khi Start debug, chức năng tắt trình duyệt Agent, kết thúc debug
	Công cụ định vị element xuất hiện trên giao diện Client
	Save as: Chức năng cho phép người dùng lưu script hiện tại thành một script mới mà vẫn giữ nguyên script cũ (một hình thức duplicate)
	Save: Lưu những thay đổi của script bao gồm phần cấu hình và metadata Chỉ enable khi cấu hình có sự thay đổi
	Delete: Chức năng cho phép người dùng xóa một action/ group... khi kéo action/ group...đó và thả vào button Delete
	Sau khi copy một action/ group...người dùng click chuột trái vào “Drag to paste” và giữ chuột, kéo đến vị trí cần chèn action/ group... đã copy và thả
	Back: Click button “Back” để quay lại màn hình thư viện script

1.3.4.6.1 Sử dụng select mode

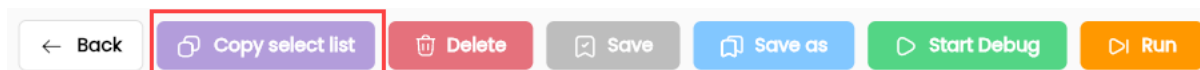
Khi cần thực hiện copy/ xóa đồng thời nhiều keyword, thực hiện click chuyển từ Basic sang select mode



Click vào check box tương ứng với keyword cần lựa chọn



- Để copy các keyword đã chọn, click Copy select list



Di chuyển đến script cần sử dụng các keyword đã copy (nếu một script tái sử dụng keyword của một script khác)



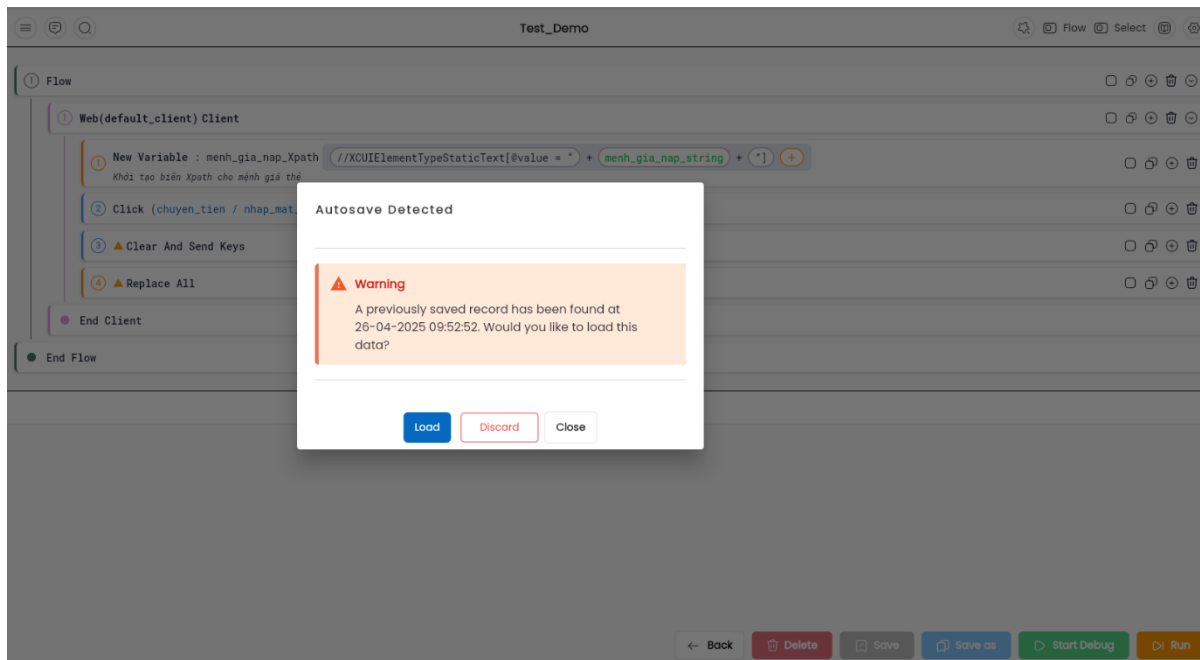
Giữ chuột trái ở button Drag to paste và kéo – thả ở vị trí cần sử dụng

- Để xóa các keyword đã chọn, giữ chuột trái vào nhóm keyword đã select, kéo xuống vị trí button Delete và thả



1.3.4.6.2 Auto save

Trong quá trình cấu hình kiểm thử tự động, người dùng vô tình thoát ra khỏi màn hình config mà chưa kịp save, khi quay lại màn hình config của script đó, hệ thống hiển thị popup xác nhận phiên bản auto save.



- Để lấy các bước đã cấu hình chưa được save trước đó, click **Load**
- Click **Discard** để hủy bỏ các bước đã cấu hình chưa kịp save
- Click **Close** để đóng popup

1.3.4.7 Cấu hình API trong script

1.3.4.7.1 Client

Client sử dụng trong API testing có nền tảng là Api

Click vào client mặc định, click Edit

Client property

Client

Web-default_client

×





+

Step comment

Enter a description

Lựa chọn Platform là Api

Edit client

Platform

Api

×

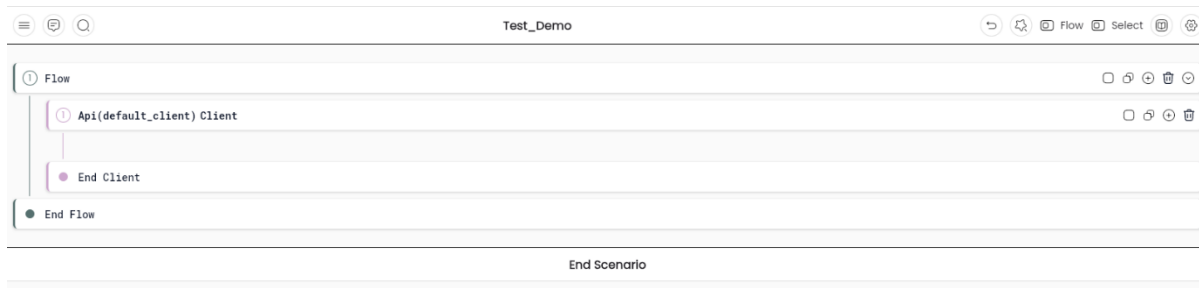
Name

default_client

SAVE

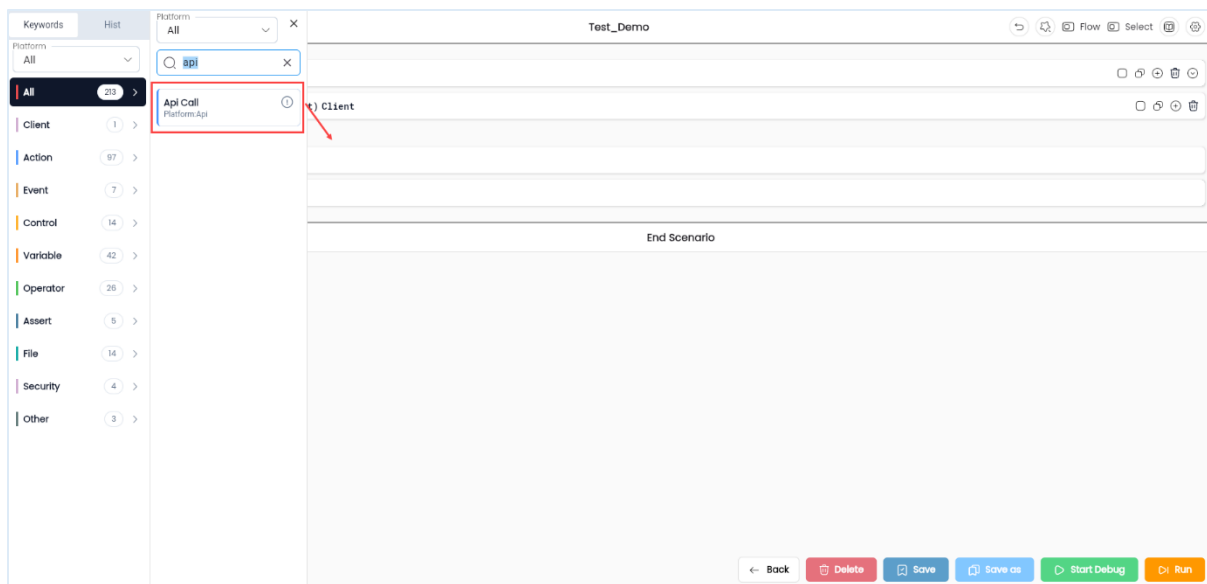
CANCEL

Client thẻ hiện trong script:



1.3.4.7.2 Sử dụng Api Call

Để gọi đến một API đã được cấu hình từ API Explore, thực hiện kéo thả keyword “Api Call” vào trong client



Cấu hình keyword “Api Call”

Bước 1: Click vào keyword “Api Call”, hệ thống hiển thị màn hình Api call property

Api Call property

Code

Select test api

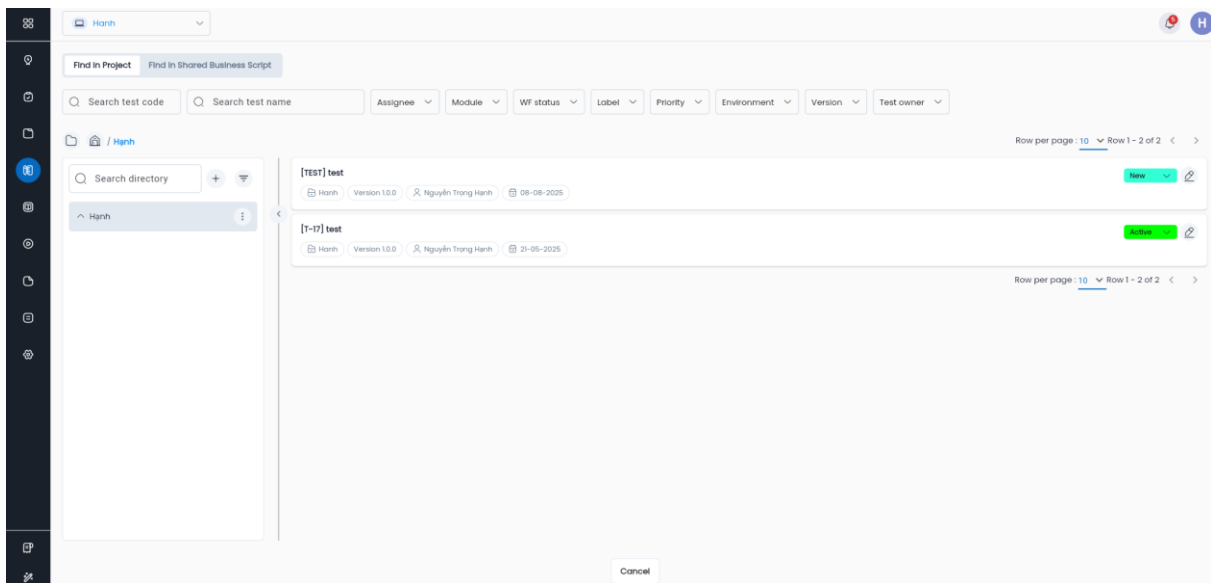
Step comment

Enter a description

Cancel

Done

Bước 2: Click Select API, hệ thống hiển thị danh sách API đã được cấu hình trong project



Bước 3: Click chọn API cần gọi đến

Bước 4: Cấu hình các trường thông tin của Api như cấu hình action call, bao gồm output



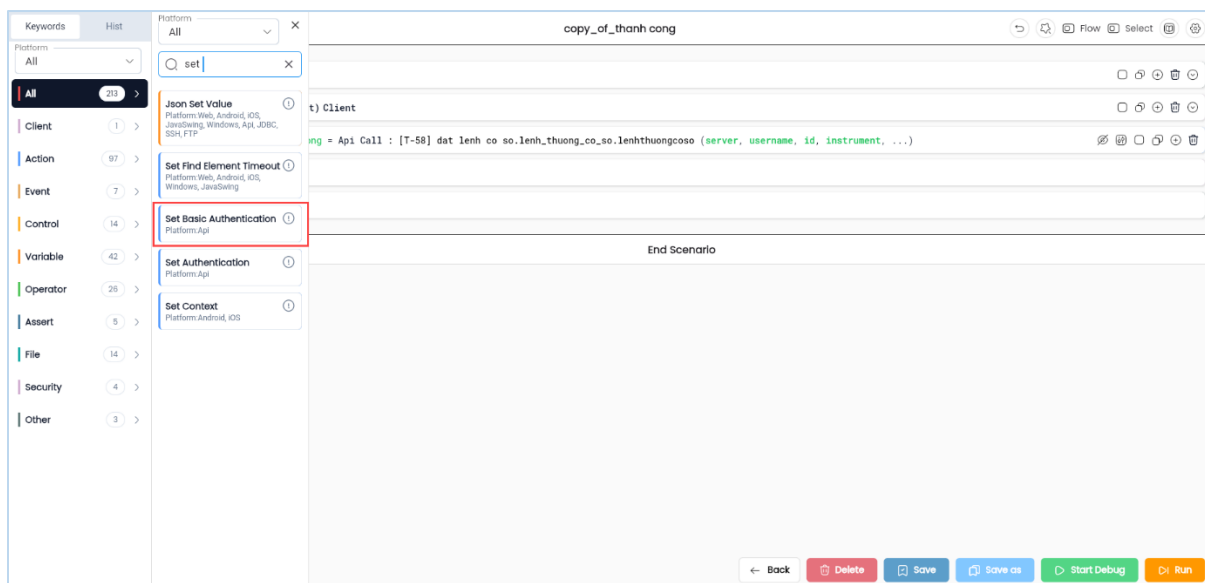
1.3.4.7.3 Set Authen khi kiểm thử Api

Tùy thuộc vào phương thức xác thực của Api được kiểm thử mà sử dụng keyword set Authen tương ứng.

Trước khi set Authen, cần thực hiện Call Api Exchange Token

- Basic Token (Basic Authentication)

Kéo thả keyword Set Basic Authentication vào trong client



Click vào keyword và thực hiện cấu hình

Set Basic Authentication property

×

Step comment

Enter a description

Input

username

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

String

Enter a value

Find value

password

Value

Variable

Agent Parameter

Device Parameter

Version Parameter

Environment Parameter

Password

Enter a value

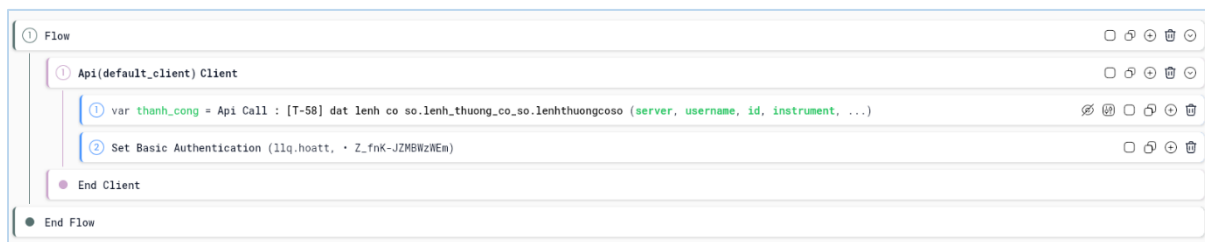
Find value

👁

Cancel

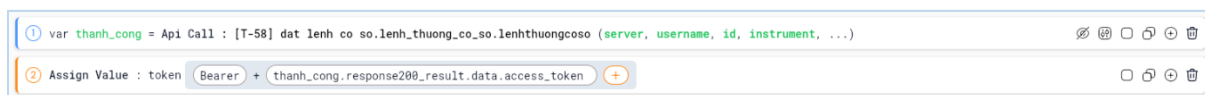
Done

Click **Done**.

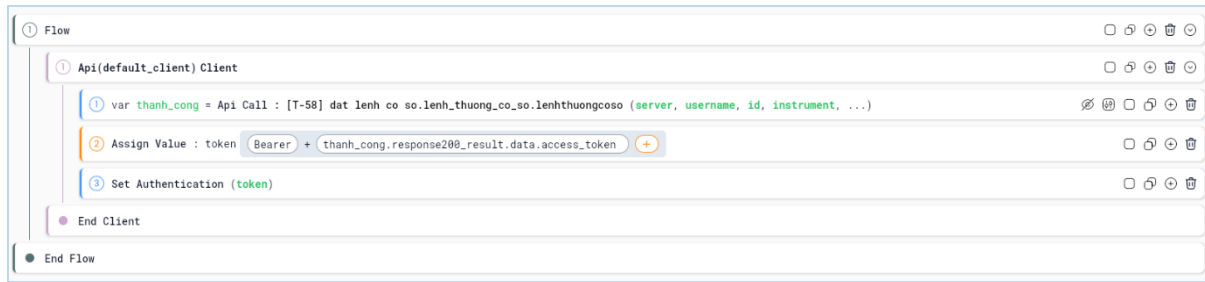


- Bearer Token

Sử dụng keyword Assign value để cộng chuỗi giá trị token, trong đó Token là output của Api call Exchange Token



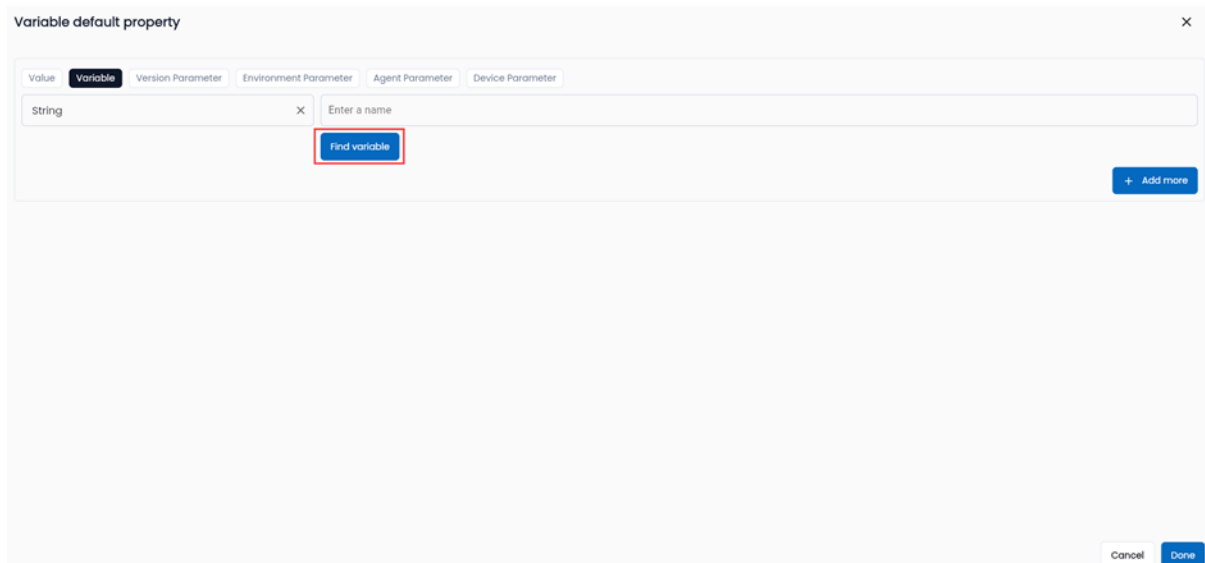
Kéo thả keyword Set Authentication vào trong client, trong đó token là giá trị của keyword Assign value



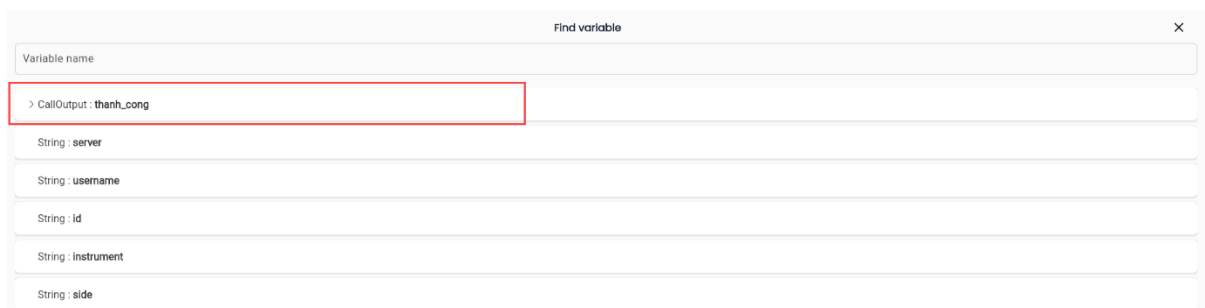
1.3.4.7.4 Lấy giá trị tham số của Api call

Để lấy giá trị của một trường thông tin trong response Api được gọi đến, Api call phải thực hiện call output (tương tự call output của shared steps)

Bước 1: Trong thuộc tính input (hoặc variable default property) của keyword, click Find variable



Bước 2: Click chọn output của api call



Bước 3: Click button +

Variable default property

Value **Variable** Version Parameter Environment Parameter Agent Parameter Device Parameter

Call Output × thanh_cong

Find variable

+ Add more

Bước 4: Nhập tên response của endpoint được gọi đến trong api call theo cấu trúc:

Response_name + “_” + “result”

Bước 5: Click button “+” để nhập tên field

Variable default property

Value **Variable** Version Parameter Environment Parameter Agent Parameter Device Parameter

Call Output × thanh_cong

Find variable

Any ▾ response

+ Add more

Bước 6: Nhập tên trường thông tin và click Done

Variable default property

Value **Variable** Version Parameter Environment Parameter Agent Parameter Device Parameter

Call Output × thanh_cong

Find variable

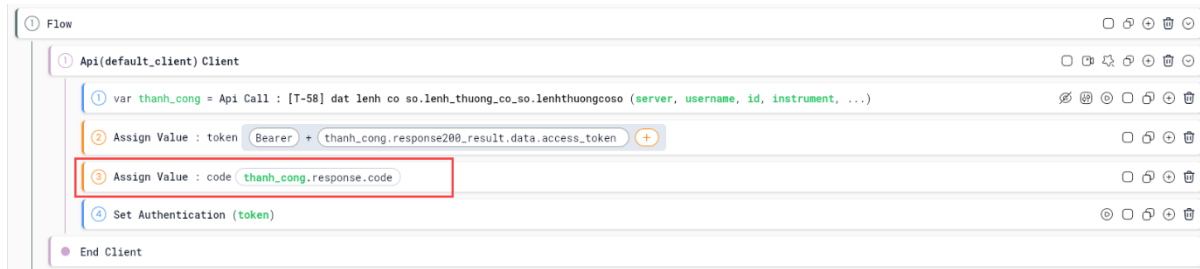
Any ▾ response

Any ▾ code

+ Add more

Cancel Done

Màn hình config script hiển thị:



Nếu trong trường thông tin nhập ở bước 6 là một schema và cần lấy giá trị của trường thông tin con của schema thì lặp lại thao tác bước 5 và 6 cho đến phân cấp của trường thông tin đó.

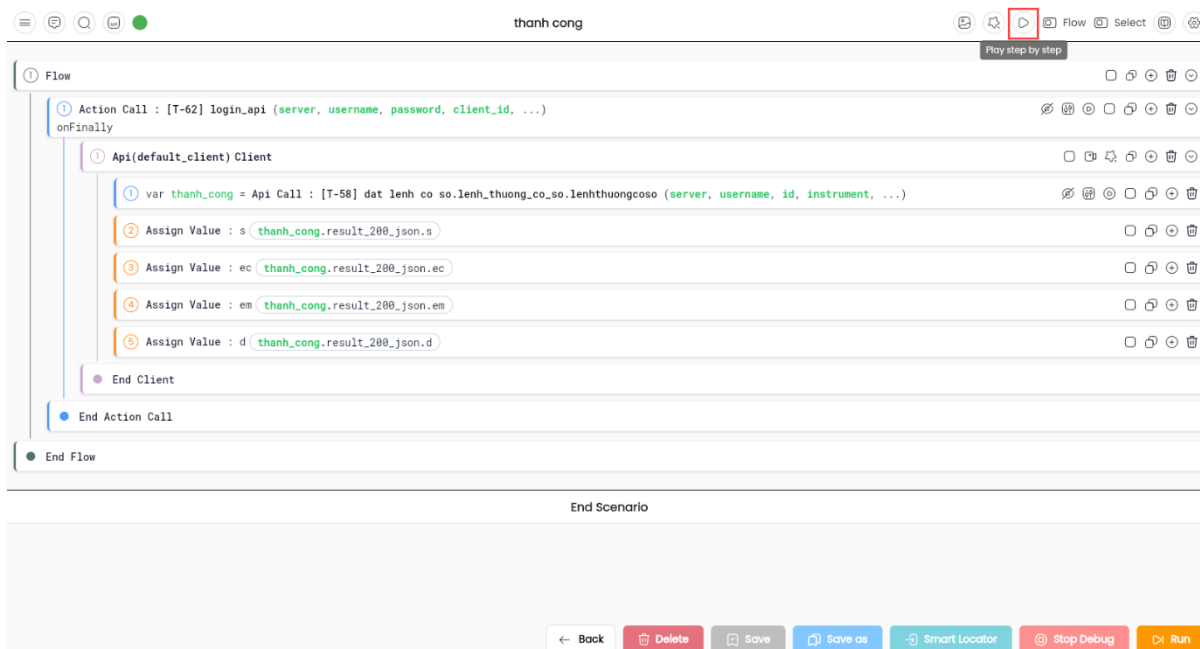
Lưu ý: Để lấy status và content của API response, tại bước 4 nhập response_tatus hoặc response_content

1.3.4.7.5 Show raw log API

Show raw log thể hiện log của API khi Play bao gồm các thông tin: Phương thức, path của endpoint, Authorization, Content-type, Request-content, response-status, response-content...

- Trên màn hình debug script

Bước 1: Trên màn hình config script, click Play list



Bước 2: Thực hiện chọn các bước sẽ play

Play thanh cong

×

Enter value for parameters

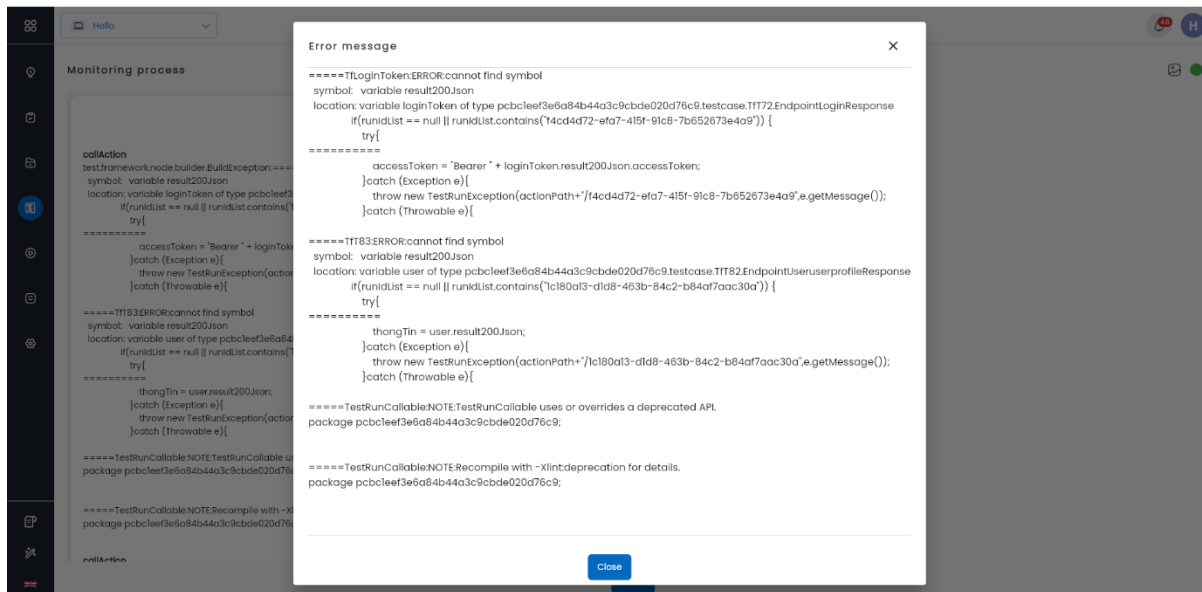
server	Enter a server
username	Enter a username
id	Enter a id
instrument	Enter a instrument
side	Enter a side
type	Enter a type
limitprice	Enter a limitprice
qty	Enter a qty
stopprice	Enter a stopprice
trailingstoplimitoffset	Enter a trailingstoplimitoffset

Ok

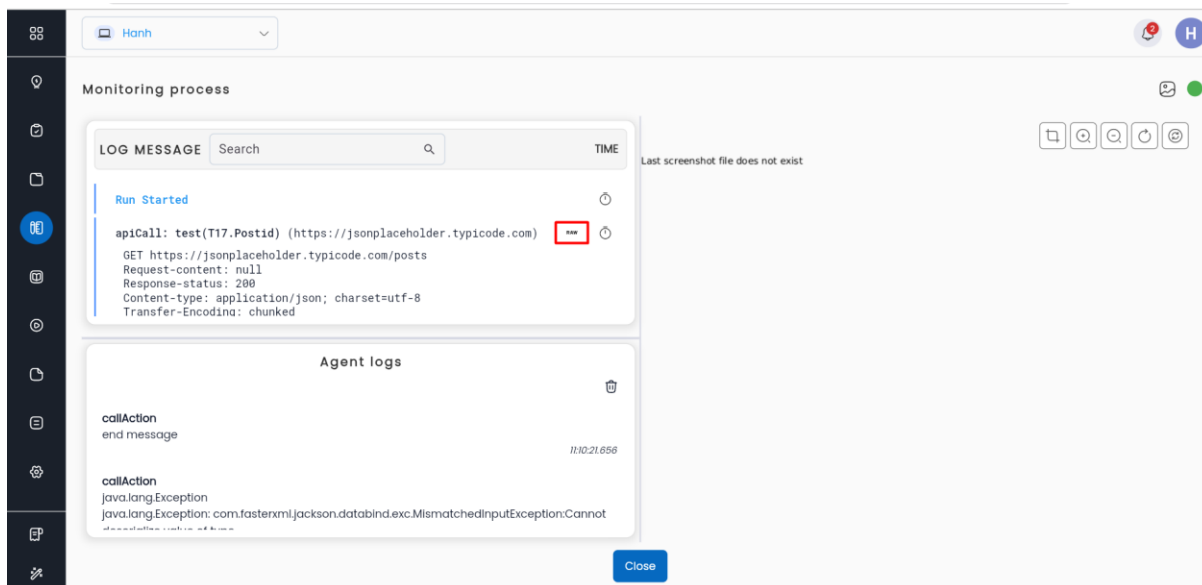
Cancel

Bước 4: Click OK

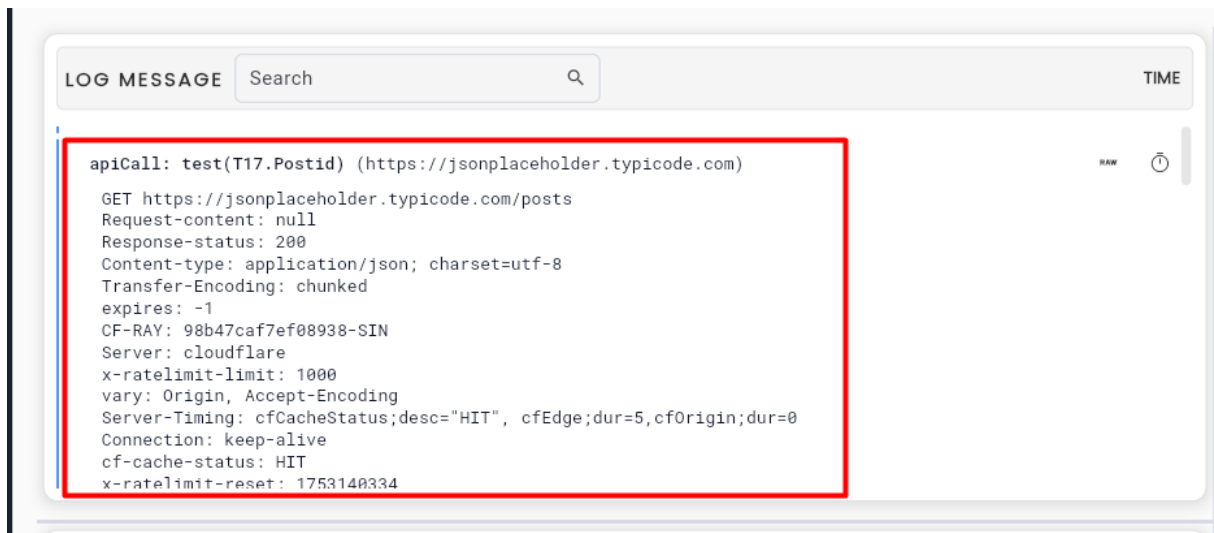
Bước 5: Hệ thống hiển thị popup kết quả của script khi play debug, click “Close” để tắt popup



Bước 6: Trên màn hình test log, click vào raw để xem log của API tương ứng

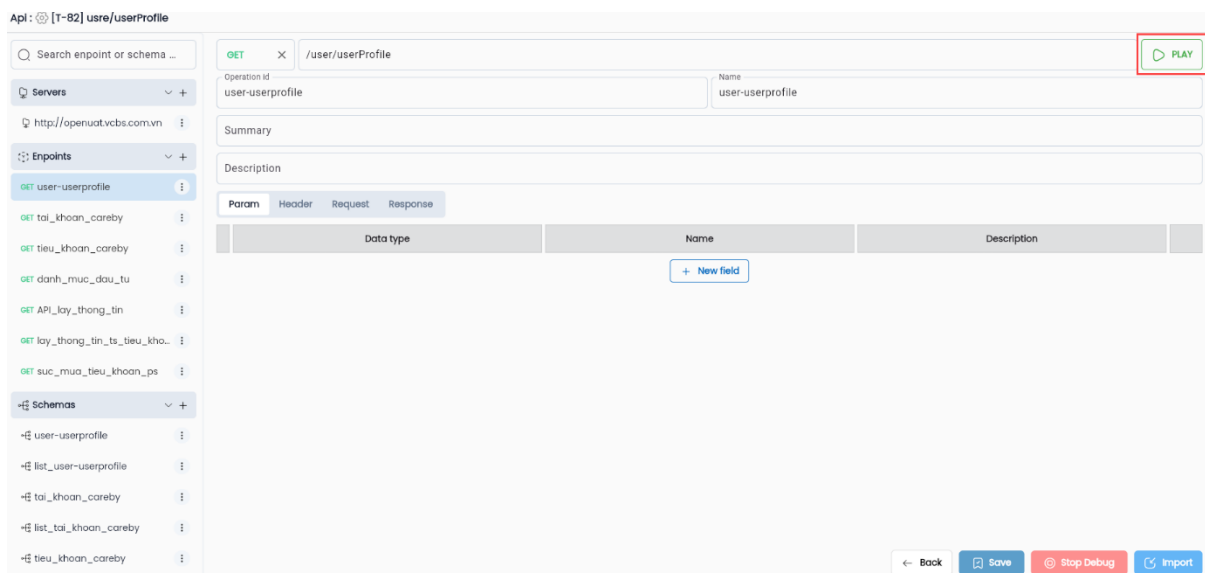


Bước 7: Xem thông tin log để xác định lỗi (nếu có)



- Trên màn hình config API

Play Endpoint



Nhập data và click OK, popup Output message hiển thị debugContent cùng các thông tin request, response của endpoint

Output message

×

```
responseStatus : 200

responseContentType : application/json; charset=utf-8

requestHeadersMap : {accept: [application/json]}

responseHeadersMap : {access-control-allow-origin: [*], date: [Tue, 29 Apr 2025 08:20:04 GMT], access-control-max-age: [-1], transfer-encoding: [chunked], access-control-allow-headers: [*], track-log-id: [f3ee0e33-0e70-4c75-a72a-064740dd419f], content-type: [application/json; charset=utf-8], access-control-allow-methods: [*]}

requestMethod : GET

requestUrl : http://openuat.vcbs.com.vn/user/userProfile

requestContent : null

responseContent : {"em":"Request chưa được xác thực hoặc access-token không hợp lệ","em_details":"","ec":"INVALID_ACESSTOKEN","s":"error"}

userResult : {s: error, ec: INVALID_ACESSTOKEN, em: Request chưa được xác thực hoặc access-token không hợp lệ, em_details: }

debugContent : {requestMethod: GET, requestUrl: http://openuat.vcbs.com.vn/user/userProfile, requestHeaders: {accept: [application/json]}, requestContent: null, responseStatus: 200, responseContentType: application/json; charset=utf-8, responseHeaders: {access-control-allow-origin: [*], date: [Tue, 29 Apr 2025 08:20:04 GMT], access-control-max-age: [-1], transfer-encoding: [chunked], access-control-allow-headers: [*], track-log-id: [f3ee0e33-0e70-4c75-a72a-064740dd419f], content-type: [application/json; charset=utf-8], access-control-allow-methods: [*]}, responseContent: {"em":"Request chưa được xác thực hoặc access-token không hợp lệ","em_details":"","ec":"INVALID_ACESSTOKEN","s":"error"}
}
```

Close

Xem thông tin từ debugContent để xác định lỗi (nếu có)

1.3.5 Mapping test case

Để mapping test case, nhân viên kiểm thử có 03 cách

1.3.5.1 [Mapping qua file excel](#) (hệ thống khuyến nghị)

1.3.5.2 Mapping qua tính năng test case của script

Bước 1: Trên màn hình edit test script, click vào tab Test case

Root > WEB > Mô tiết kiệm

[WEB_SC_TK_014] WEB-Mở tài khoản tiết kiệm - Kiểm tra giao diện màn Khởi tạo

Description

Add a description...

Metadata Clients Parameters Output **Test case** Test runs Issue Comment

Directory

Root > WEB > Mô tiết kiệm

Change directory

Summary

Type Test script

Test code WEB_SC_TK_014

Test name WEB-Mở tài khoản tiết kiệm - Kiểm tra giao diện màn Khởi tạo

Information

Task code WEB_SC_TK_014

Task name WEB-Mở tài khoản tiết kiệm - Kiểm tra

Task priority Medium

Task difficulty Medium

Task Status Draft

Save Update to Test Library Cancel

Hệ thống hiển thị danh sách test case đã tham chiếu đến script

Root > WEB > Mô tiết kiệm

[WEB_SC_TK_014] WEB-Mở tài khoản tiết kiệm - Kiểm tra giao diện màn Khởi tạo

Description

Add a description...

Metadata Clients Parameters Output **Test case** Test runs Issue Comment

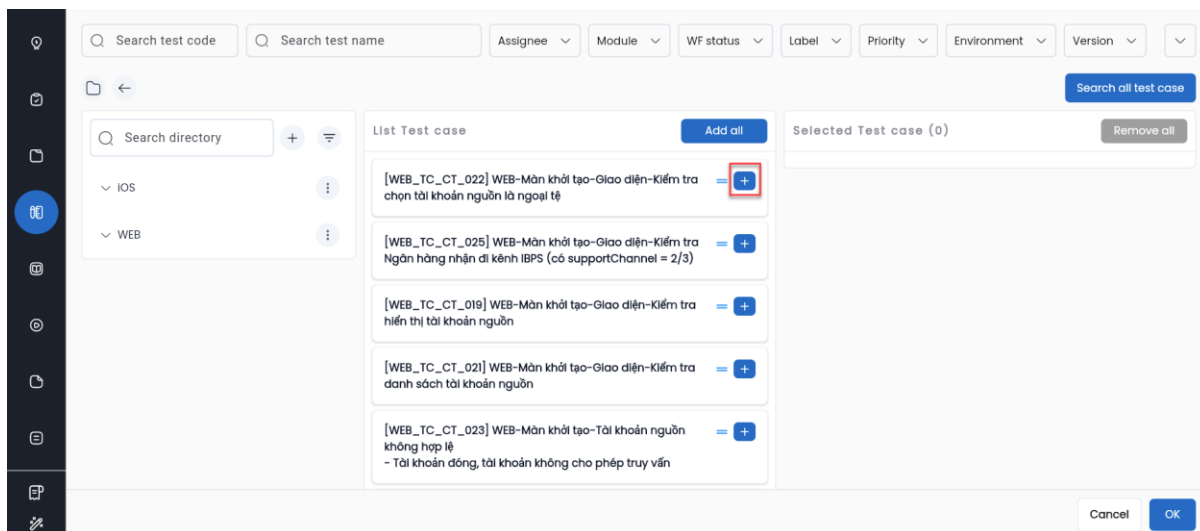
List Test case of [WEB_SC_TK_014] WEB-Mở tài khoản tiết kiệm - Kiểm tra giao diện màn Khởi tạo

Select Test case + New test

Save Cancel

Bước 2: Rà soát danh sách test case của script, nếu còn thiếu, click “Select test case”, hệ thống hiển thị danh sách các test case chưa được tham chiếu đến bất kỳ script nào.

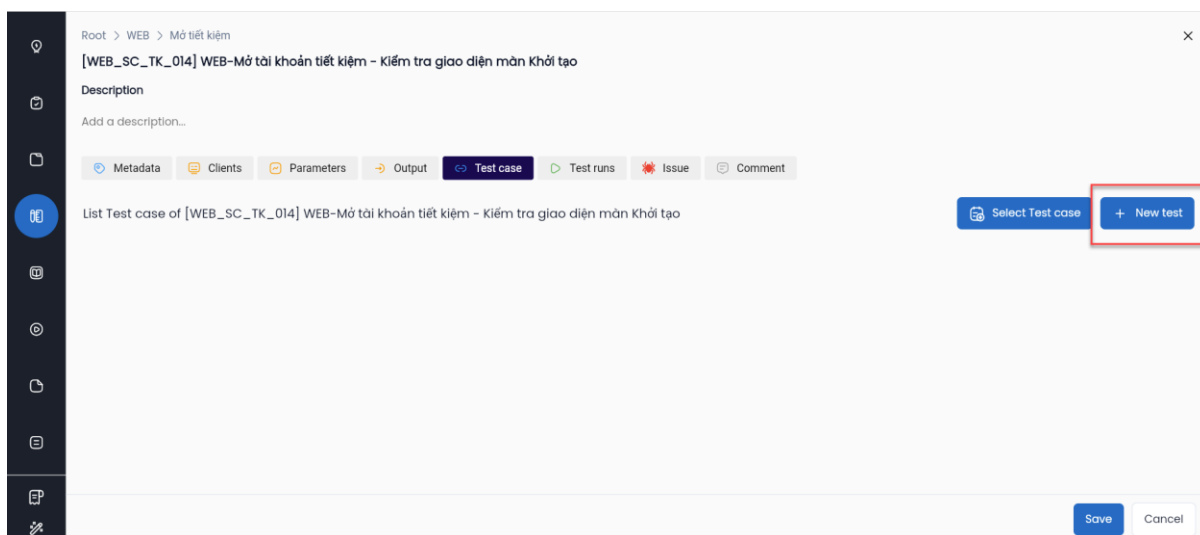
Bước 3: Click button “+” tương ứng với test case cần chọn, có thể chọn nhiều test case trong 01 lần thao tác



Bước 4: Click OK, hệ thống hiển thị thông báo “Add successful”

Trong quá trình nâng version của ứng dụng/ hệ thống kiểm thử có thể phát sinh test case hoặc danh sách kịch bản kiểm thử cần bổ sung thêm test case.

Trên màn hình tab **test case** của script, click “NEW TEST”



Hệ thống hiển thị màn hình metadata của test case, trong đó tại tab Data script/suite được chọn mặc định là script mà nhân viên kiểm thử đang thao tác.

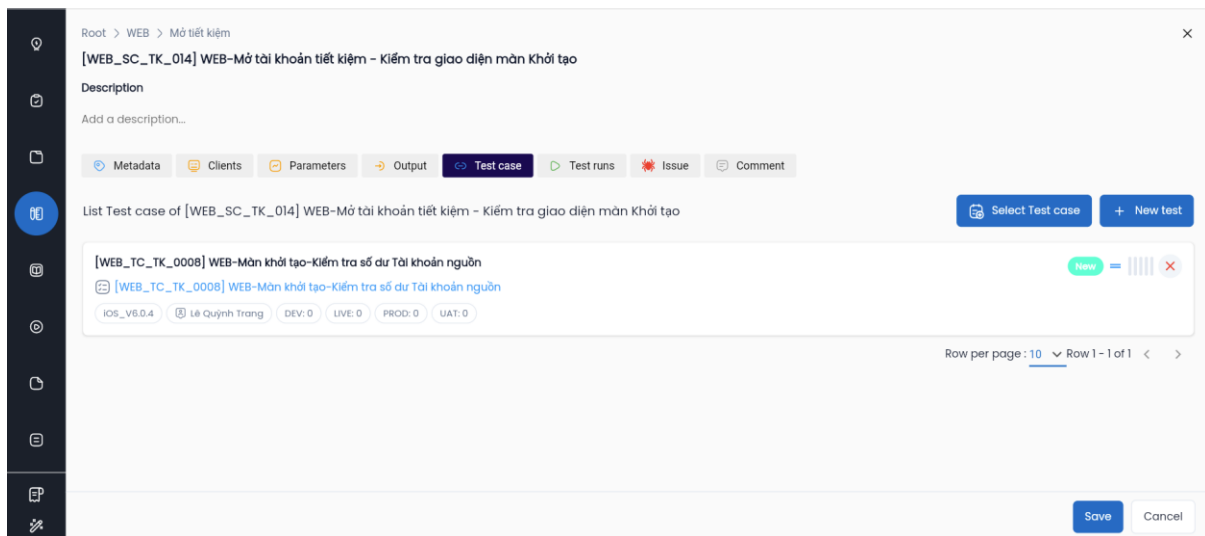
The screenshot shows the 'CREATE TEST CASE' form with the 'Metadata' tab selected. The form includes a 'Directory' section with a 'Root' dropdown and a 'Change directory' button. Below this is a 'Summary' section with fields for 'Type' (set to 'Test case'), 'Test code', and 'Test name'. To the right is an 'Information' section with fields for 'Task code', 'Task name', 'Task priority', 'Task difficulty' (set to 'Medium'), and 'Task Status'. At the bottom right are 'Next' and 'Cancel' buttons.

Nhập đầy đủ và đúng thông tin vào metadata sau đó click **Next**.

Màn hình hiển thị tab Data. Nhập đầy đủ và đúng thông tin vào các tab config, data.

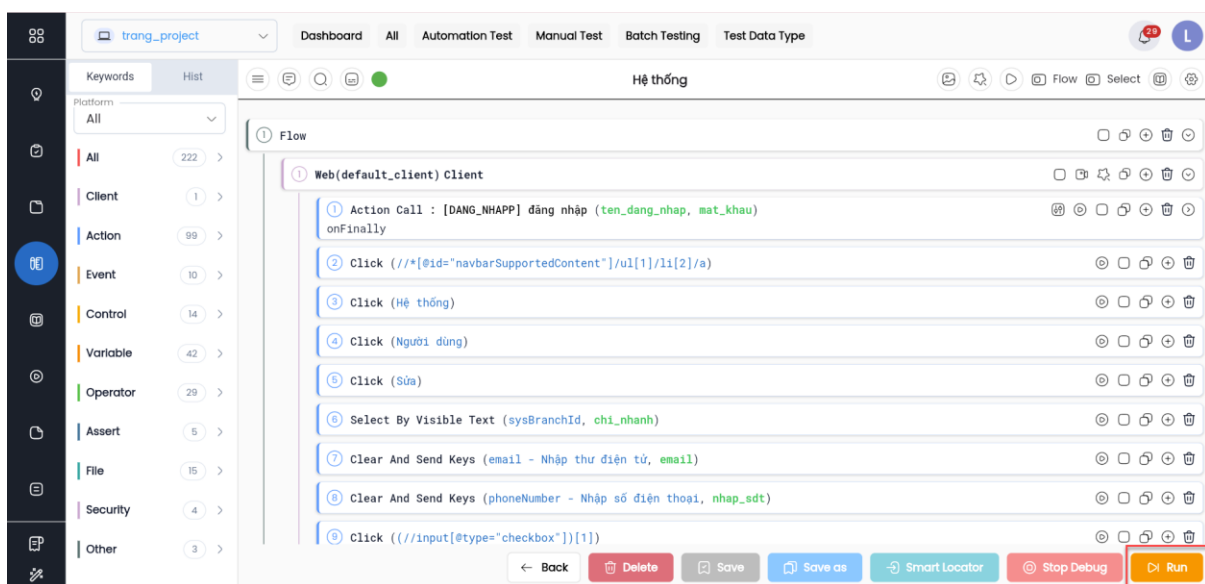
The screenshot shows the 'CREATE TEST CASE' form with the 'Data' tab selected. The form displays the 'Test case name' as 'WEB-Màn khởi tạo-Kiểm tra số dư Tài khoản nguồn' and a detailed description '[WEB_SC_TK_014] WEB-Mô tài khoản tiết kiệm - Kiểm tra giao diện màn Khởi tạo'. Below the description are project selection buttons: 'PRJ-DEV', 'PRJ-LIVE', 'PRJ-PROD', and 'PRJ-UAT'. There are also buttons for 'Select test script' and 'Select Test suite'. At the bottom right, the 'Save' button is highlighted with a red box, next to a 'Cancel' button.

Sau khi click Save, test case mới được tạo trên project đồng thời cũng xuất hiện trong list test case của script.



1.3.5.3 Sử dụng tính năng Generate test case

Để sử dụng tính năng Generate test case, trên màn hình cấu hình kiểm thử tự động của Test script, click Run.



Khi màn hình chạy kiểm thử hiển thị, nhân viên kiểm thử cần nhập dữ liệu vào trường thông tin test case (giá trị là test case code).

trung_project Dashboard All Automation Test Manual Test Batch Testing Test Data Type

Test Run: Draft Schedule time: 07-08-2025 09:33:58

Test run description

Test data

+ Add

Data 1

Test case code Enter a test case

Test case name Enter a test case

Task group name Enter a task group name

Test Run Expected Result Pass X

Severity Critical X

Priority Medium X

Cancel Run

Thông tin này có thể được nhập thủ công hoặc chọn từ danh sách có sẵn bằng cách sử dụng tính năng **Select test case**

trung_project Dashboard All Automation Test Manual Test Batch Testing Test Data Type

Test run description

Test data

+ Add

Data 1

Test case code Enter a test case

Test case name Enter a test case

Task group name Enter a task group name

Test Run Expected Result Pass X

Severity Critical X

Priority Medium X

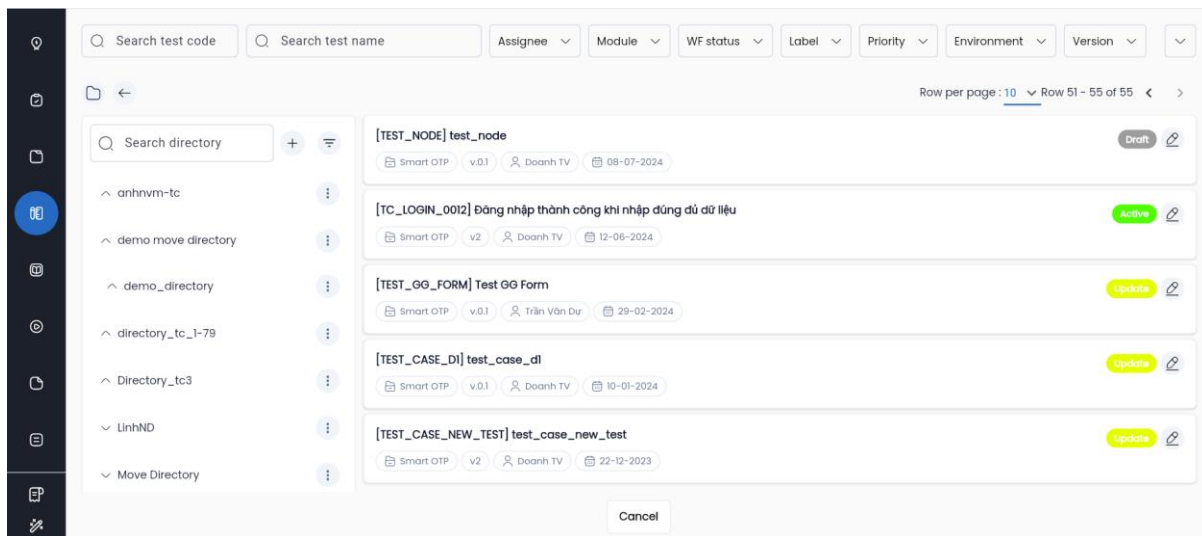
ten_dang_nhap Enter a ten_dang_nhap

mat_khau Enter a mat_khau

Cancel Run

- Sử dụng tính năng select test case

Bước 1: Click Select test case trên trường thông tin Test case code



Bước 2: Tìm, chọn test case cần mapping, hệ thống tự động điền giá trị test case code và test case name vào trường thông tin data của script

Test data

[+ Add](#)

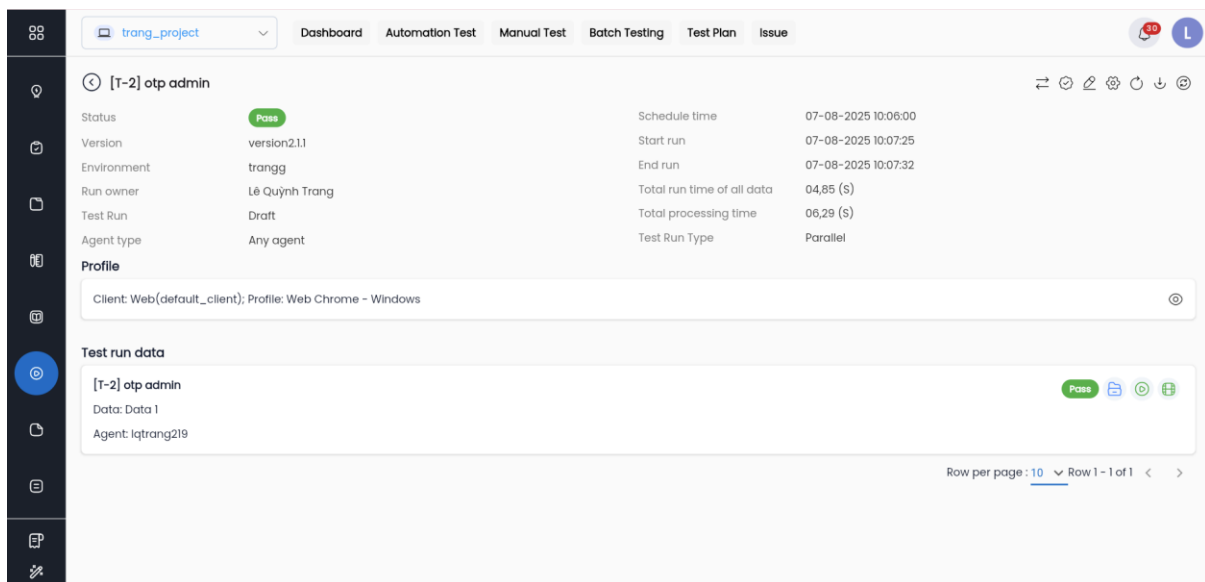
Data 1

Test case code	TC_01	
Test case name	Login thành công	
Task group name	Enter a task group name	
Test Run Expected Result	Pass	
Severity	Critical	

Thao tác tương tự với các test case khác của script

Bước 3: Click Run và đợi hệ thống thực thi xong toàn bộ data được nhập trong lần run

Bước 4: Trên màn hình kết quả thực thi click chức năng **Generate test case**



Hệ thống hiển thị thông báo “***Generate successful***” đồng thời mapping test case với script tương ứng và nhập data cho test case theo môi trường đã thực thi.

1.4 Active script

Active là trạng thái ghi nhận test script/ suite/ test case đã hoàn thành và sẵn sàng cho công cuộc thực thi kiểm thử chính thức.

1.4.1 Check list before active

Một test case/ test script/ suite được coi là đã hoàn thành khi đạt được đầy đủ các hạng mục trong check list

1.4.1.1 Test script, suite

Trước khi active test script/ suite, nhân viên kiểm thử cần đảm bảo một số điểm của test case:

Hạng mục	Bắt buộc	Mục đích
Đã select đủ test case	Có	Đảm bảo test script bao quát đầy đủ các tình huống cần kiểm thử. Khi có danh sách test case đầy đủ, việc xây dựng script sẽ chính xác và không bị bỏ sót step.

Steps của script đã bao phủ toàn bộ test case của script	Có	Khi có steps bao phủ phù hợp, nhân viên kiểm thử có thể thực hiện cấu hình theo steps mà không cần quan tâm đến danh sách test case hay steps của từng test case trong quá trình cấu hình.
Cấu hình đầy đủ các bước trong steps	Có	Đảm bảo có thể kiểm thử toàn bộ luồng nghiệp vụ như yêu cầu.
Đã run với đầy đủ data của luồng	Có	Đảm bảo luồng đi đúng với tất cả data hợp lệ
Script được cấu hình output đầy đủ	Có	Đảm bảo kết quả thực thi phản ánh đúng trạng thái thực tế của script, phục vụ kiểm tra và đánh giá kết quả.
Last run status Pass	-	Trong điều kiện lý tưởng, không phát sinh lỗi element not found Khi không có lỗi từ ứng dụng được kiểm thử và với bộ data hoàn chỉnh, script phải được chạy thông luồng và đạt được expect (pass/ trả ra output theo required) Khi last run status là fail cần nêu ra được nguyên nhân fail khách quan: - Ứng dụng chưa thông luồng/ chỉ có thể đi đến bước này - Lỗi từ ứng dụng

1.4.1.2 Test case

Hạng mục	Bắt buộc	Mục đích
Chọn đúng test script/ suite	Có	Chọn đúng test script/ suite cho test case để kiểm thử đúng luồng
Nhập đúng, đủ data cho từng môi trường	Có	Đảm bảo quá trình kiểm thử phản ánh đúng tình trạng trên từng môi trường (Trong trường hợp thực thi trên nhiều hơn một môi trường phát triển ứng dụng/ phần mềm)

Chọn đúng mức độ nghiêm trọng khi test case run có status là fail	Có	Chọn đúng mức độ nghiêm trọng khi kết quả thực thi là fail sẽ giúp phân loại, đánh giá kết quả chính xác, thuận tiện hơn, đặc biệt khi run batch
Chọn đúng kết quả mong đợi của test case	Có	Chọn đúng kết quả thực thi mong muốn của test case Với mỗi test case positive thông thường expect result là Pass, đối với test case negative cần xác định theo cấu hình của script và data tương ứng
Last run status: Pass	-	Trong điều kiện lý tưởng (không có lỗi từ ứng dụng/ phần mềm được kiểm thử) với data hoàn chỉnh test run status phải đạt được kết quả là Pass Trường hợp fail - cần nêu ra nguyên nhân dẫn đến fail và không phải do cấu hình script

1.4.2 Quy trình active test case/ script/ suite

Quy trình active của các loại test là giống nhau, tùy thuộc vào quy định của tổ chức (đối tượng sử dụng hệ thống LLQ QA Platform) mà quy trình trải qua 1 bước (nhân viên kiểm thử tự active) hoặc 2 bước (nhân viên kiểm thử - người phụ trách đánh giá test của project).

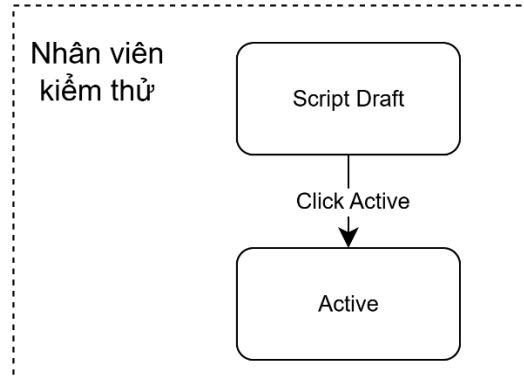
Một số trạng thái của script trong suốt quá trình triển khai dự án kiểm thử:

- **Draft:** Nháp, thể hiện script chưa hoàn thành, là trạng thái mặc định sau khi được khởi tạo của script
- **Review:** thể hiện script đang được kiểm tra, đánh giá bởi người phụ trách, xuất hiện khi người thực hiện click **“Send to Review”**
- **Active:** thể hiện trạng thái đã hoàn thành của script. Script chuyển trạng thái Active khi được Approve bởi người phụ trách (assign to review)
- **Reject:** thể hiện trạng thái bị từ chối active bởi người đánh giá. Script chuyển trạng thái từ Review sang Reject khi người đánh giá click Reject
- **Update:** thể hiện trạng thái script đang được cập nhật, chỉ xuất hiện khi người dùng click “Update” từ trạng thái Active

Script cần được Active để ghi nhận đã hoàn thành cấu hình.

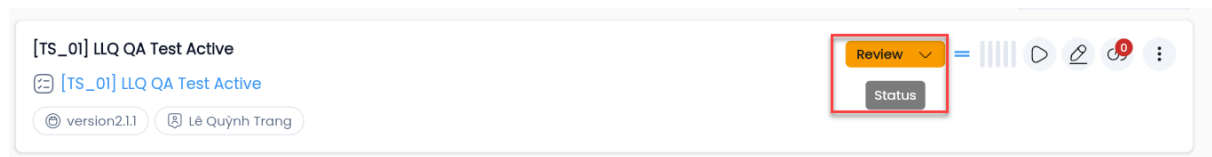
Dưới đây là 2 quy trình mà LLQ QA Platform khuyến nghị sử dụng, người dùng có thể thay đổi quy trình Active test theo quy định của cơ quan, tổ chức triển khai dự án kiểm thử.

1.4.2.1 Quy trình 01 bước

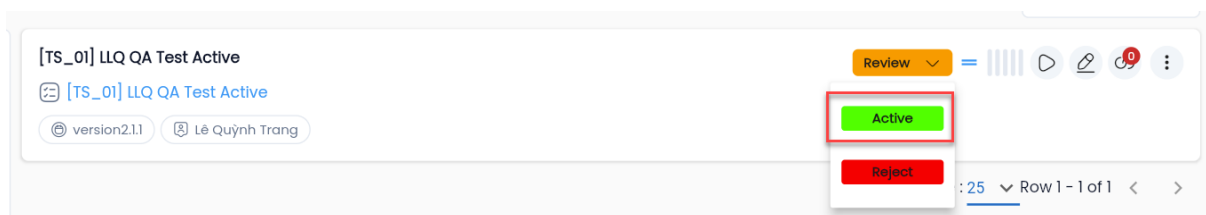


Bước 1: Kiểm tra, hoàn thành [check list before active](#) của script

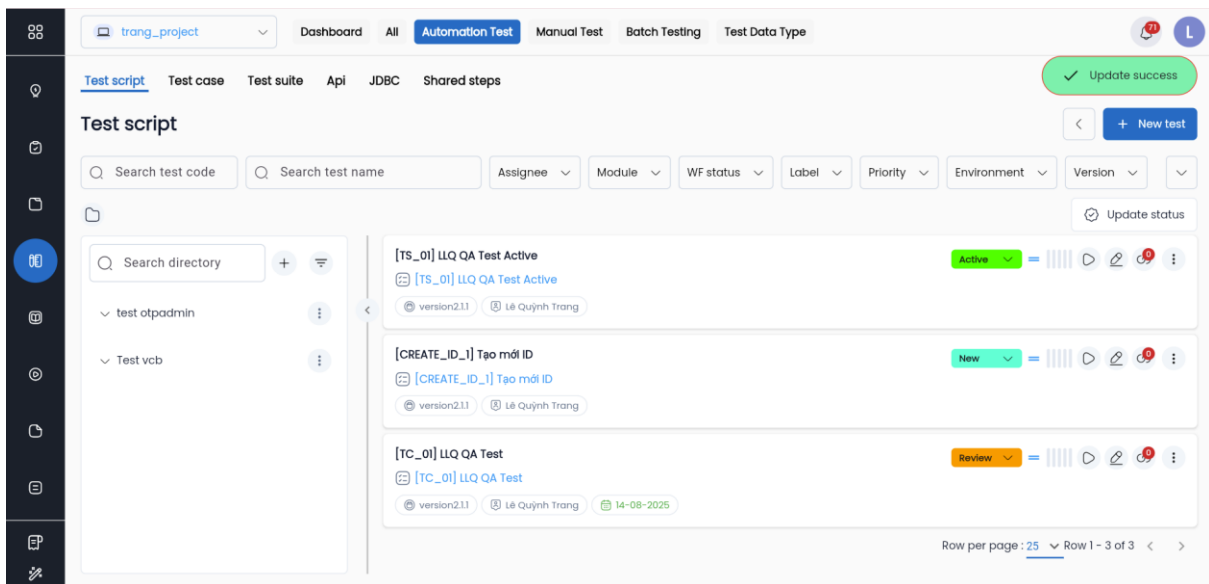
Bước 2: Sau khi hoàn thành cấu hình, nhân viên kiểm thử click vào trạng thái hiện tại tương ứng với script cần active



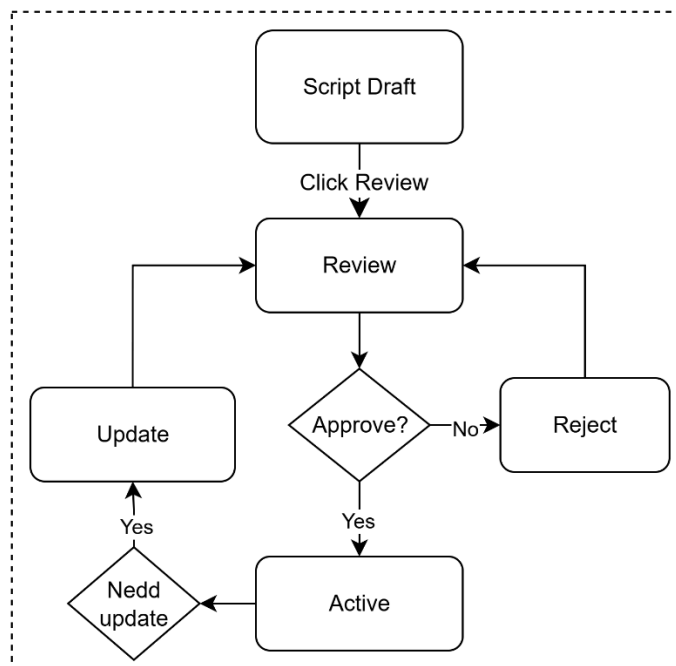
Bước 3: Click Active



Bước 4: Hệ thống hiển thị màn hình thông tin của test, click Active, trạng thái của test được cập nhật là Active



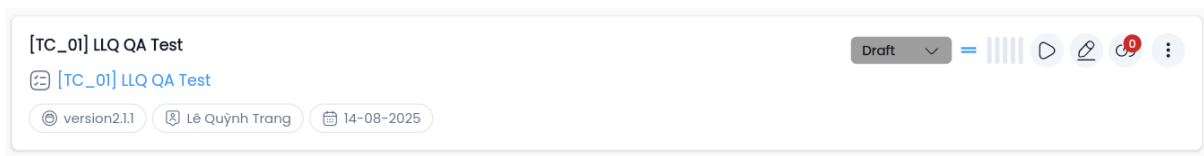
1.4.2.2 Quy trình 02 bước



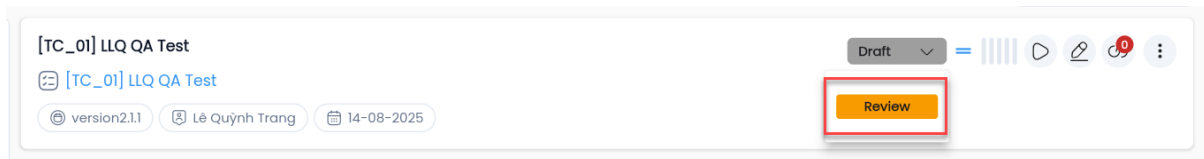
Quy trình 02 bước được thực hiện bởi nhân viên kiểm thử (user được assign thực hiện kiểm thử) và người đánh giá chất lượng kiểm thử (user mang role test management/ test lead và được “*assign to review*” script)

Bước 1: Kiểm tra, hoàn thành [check list before active](#) của script

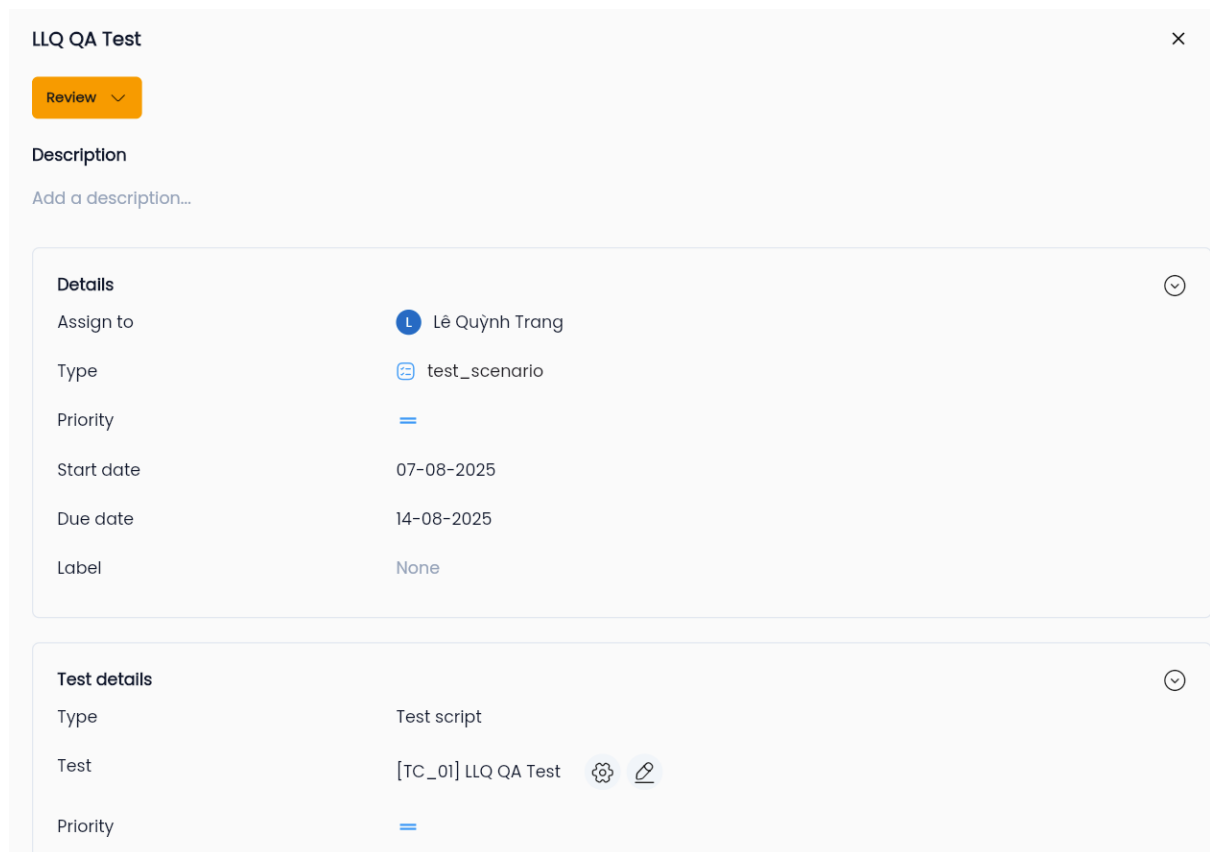
Bước 2: Sau khi hoàn thành cấu hình, nhân viên kiểm thử click vào icon Draft tương ứng với script cần active



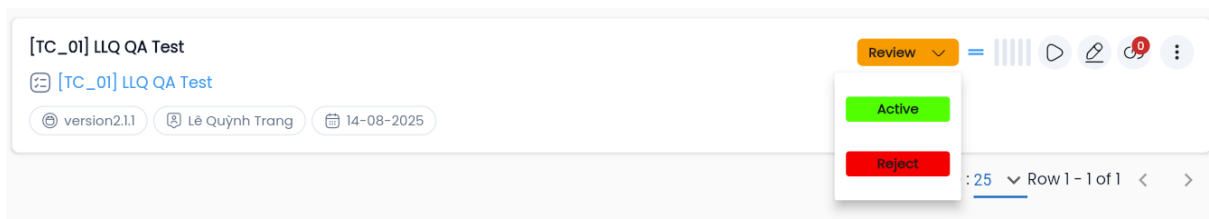
Bước 3: Click Review



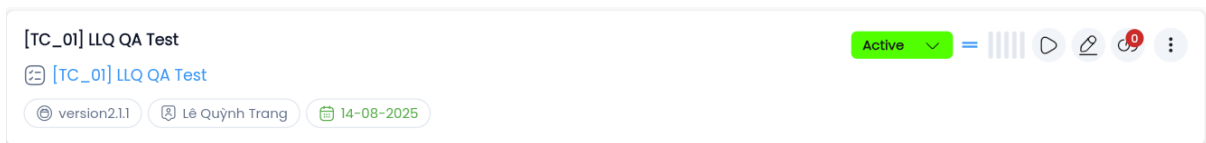
Bước 4: Hệ thống tự động cập nhật trạng thái của script là Review. Màn hình hiển thị thông tin của test ở trạng thái **Review**, người dùng có thể nhập comment (nếu có), hệ thống gửi yêu cầu đến cho user có role “**Test management/ Test lead**”.



Bước 5: User có Role “**Test management/ Test lead**” của project thực hiện đi đến script được request review, click Review tương ứng



Bước 6: Click **Active**, script được cập nhật status **Active**



Lưu ý: Tại bước 6, nếu người dùng click Reject, cần phải nhập thông tin, lý do reject active và click Reject, hệ thống cập nhật trạng thái Reject cho script