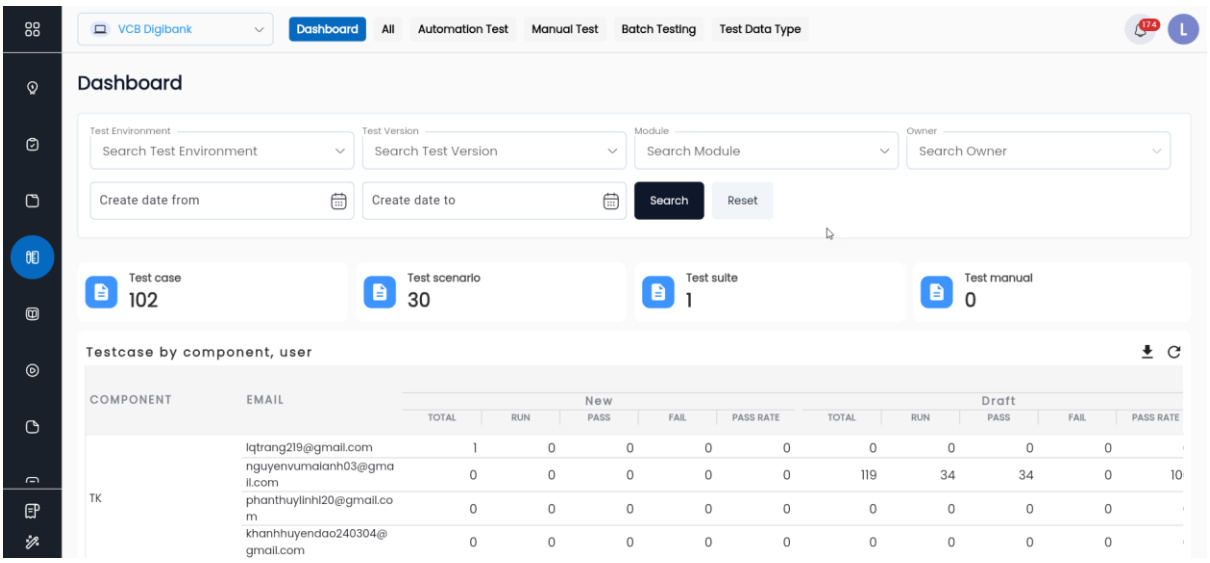
1.1 Test Management

1.1.1 Tổng quan về Test Management

Test Management là thư viện chứa, quản lý toàn bộ các kịch bản kiểm thử của dự án.

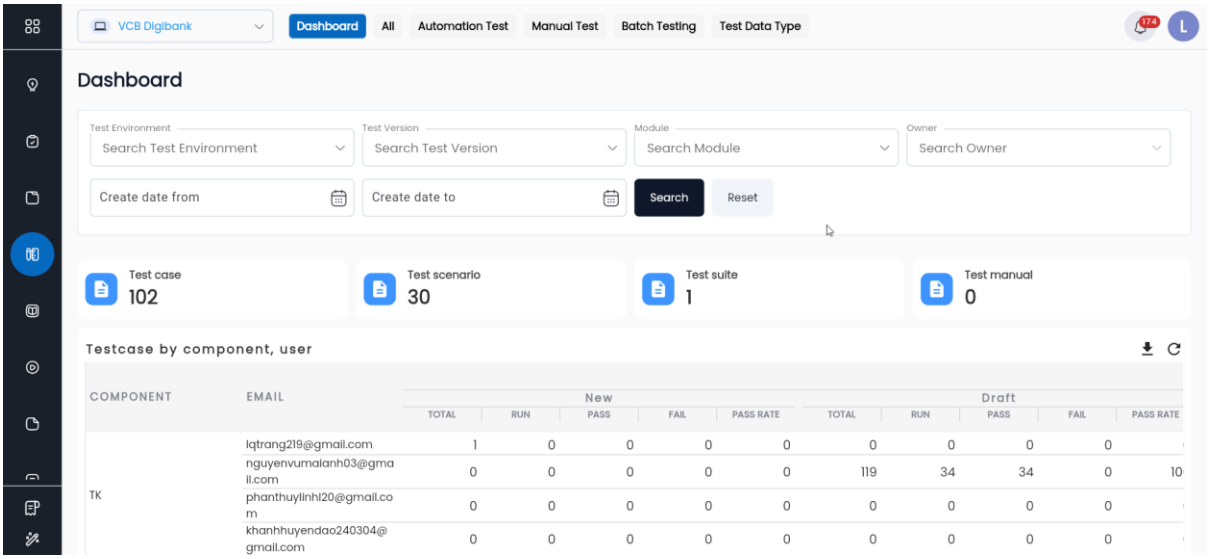


Test Management được chia thành các tab nhỏ để quản lý kịch bản kiểm thử theo tính chất riêng biệt bao gồm:

- Dashboard: Thống kê số lượng các Test Case, Test Script, Test Suit, Test Manual.
- All: Danh sách các kịch bản test theo tất cả các loại kiểm thử trên thanh header.
- Automation Test: Các kịch bản kiểm thử tự động bao gồm: Test case, Test suite, Test Script.
- Manual Test: Các kịch bản kiểm thử thủ công
- Batch Testing: Các kịch bản kiểm thử hàng loạt phục vụ Regression test và Smoke Testing.
- Test Data Type: Các bộ dữ liệu kiểm thử được xây dựng để phục vụ cấu hình kịch bản kiểm thử

1.1.2 Filtering Test Management

1.1.2.1 Filtering tại Dashboard

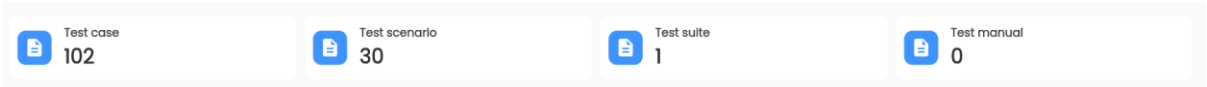


Để thực hiện lọc tại tab Dashboard, người dùng nhập các trường thông tin theo mong muốn vào khu vực tìm kiếm, trong đó:

Trường thông tin	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Test Version	-	Combobox		Tìm kịch bản kiểm thử theo phiên bản của dự án đã có trên hệ thống
Test Environment	-	Combobox		Tìm kịch bản kiểm thử theo môi trường phát triển sản phẩm
Module	-	Combobox		Tìm kịch bản kiểm thử theo cấu phần của dự án
Owner	-	Combobox		Tìm kịch bản kiểm thử theo người tạo
Create date from	-	Date		Tìm kiếm theo ngày tạo từ ngày

Create date to	-	Date		Tìm kiếm theo ngày tạo tới ngày
Search	-	Button		Click "Search" để tìm kịch bản kiểm thử theo điều kiện lọc đã nhập trước đó
Reset	-	Button		Click "Reset" để xoá các điều kiện lọc trước đó

Kết quả tìm kiếm: Số lượng kịch bản kiểm thử theo từng loại có trong dự án:

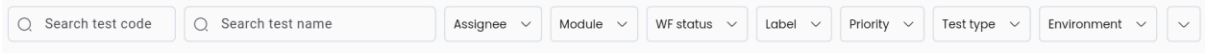


Trong đó:

Trường thông tin	**Mô tả**
TYPE	Loại kịch bản kiểm thử
AMOUNT	Số lượng kịch bản kiểm thử theo từng loại kịch bản kiểm thử

1.1.2.2 Filtering tại tab loại kịch bản kiểm thử

Filtering các tab chức năng Automation Test, Manual Test, Batch Testing, Test Data Type là tương tự nhau.



Để thực hiện lọc, người dùng nhập các trường thông tin theo mong muốn, trong đó:

Trường thông tin	**Bắt buộc**	**Định dạng**	**Ràng buộc**	**Mô tả**
Search test code	-	Text		Tìm kịch bản kiểm thử theo mã
Search test name	-	Text		Tìm kịch bản kiểm thử theo tên

WF status	-	Combobox		Tìm kiếm theo trạng thái của kịch bản kiểm thử: - New - Draft - Review - Active - Reject - Update
Assignee	-	Combobox		Tìm kiếm theo thành viên được gán
Enviroment	-	Combobox		Tìm kiếm theo môi trường phát triển
Module	-	Combobox		Tìm kiếm theo cấu phần
Priority	-	Combobox		Tìm kiếm theo mức độ ưu tiên
Label	-	Combobox		Tìm kiếm theo nhãn của kịch bản kiểm thử

1.1.3 Quản lý theo directory

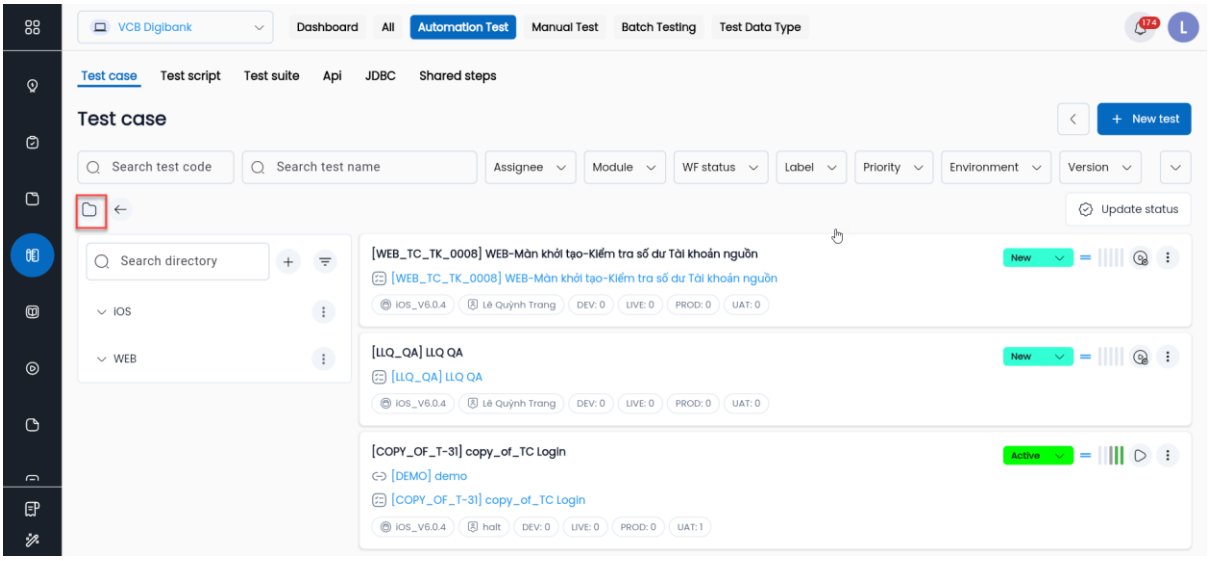
Directory là thư mục chứa các kịch bản kiểm thử, được người dùng gom lại theo người tạo, cấu phần, nền tảng thực hiện (Web, App…)

Người dùng không bắt buộc phải tạo directory để quản lý, tuy nhiên nên gom các kịch bản kiểm thử có đặc điểm, tính chất tương tự vào một directory để dễ dàng quản lý, theo dõi, kiểm soát số lượng kịch bản kiểm thử, đảm bảo thực hiện đủ các luồng nghiệp vụ, data test mà dự án yêu cầu.

1.1.3.1 Ẩn/Hiện tab Directory

Tab directory hiện tại trên hệ thống có thể linh động đóng/mở tùy ý người dùng.

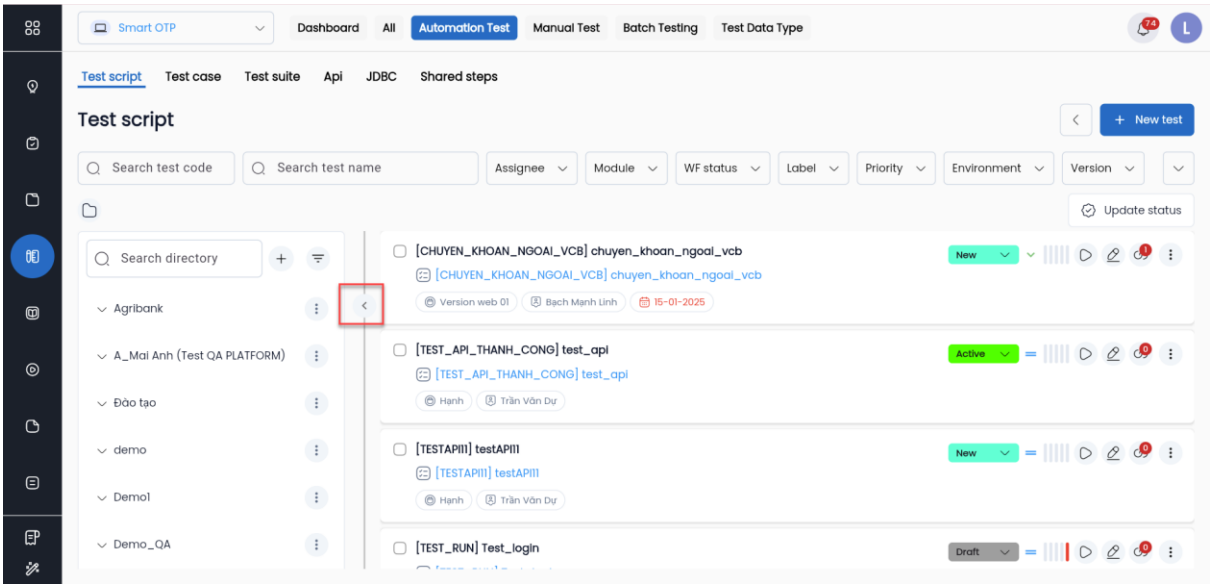
Cách 1: Click để ẩn / hiện directory



Cách 2:

Click button “

”



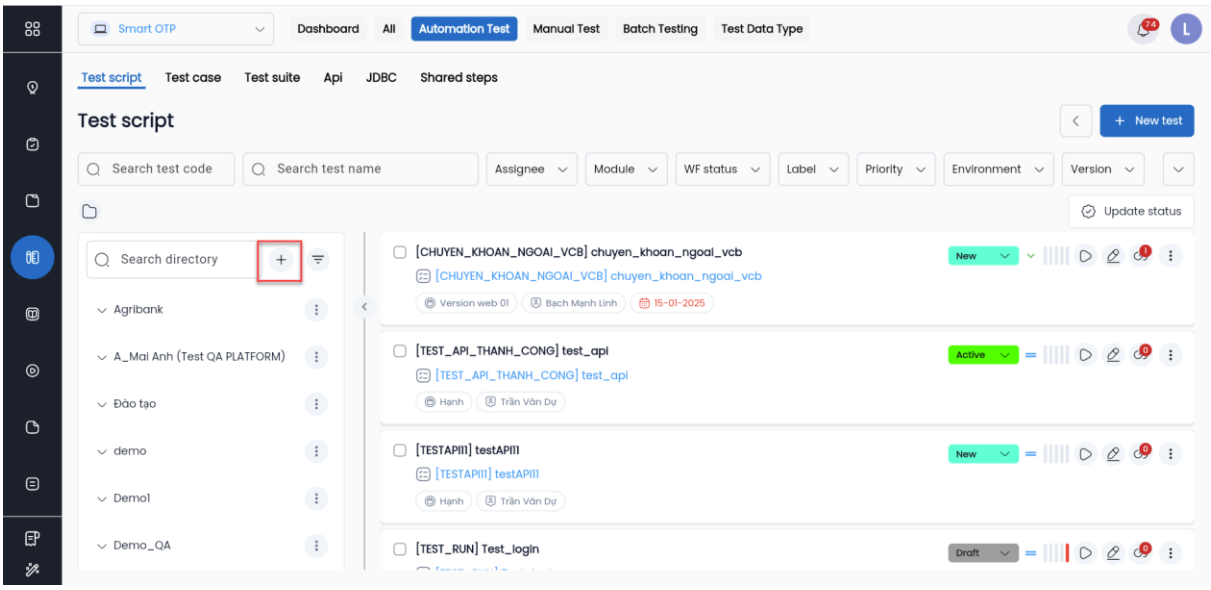
1.1.3.2 Tạo Directory

Tạo Directory trong tab Script

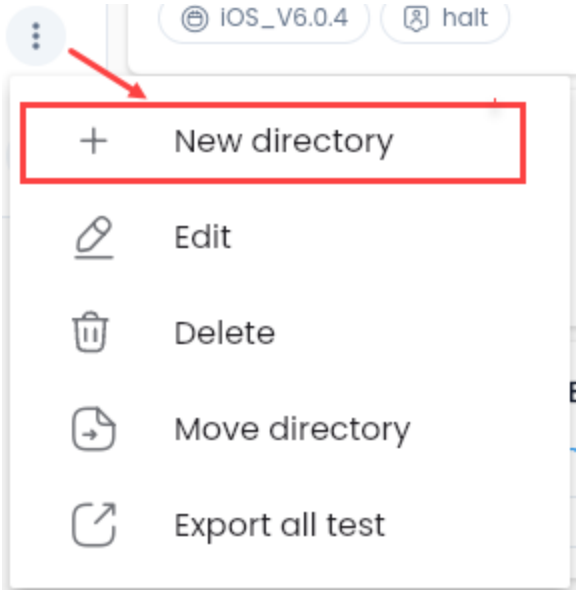
Bước 1: Thực hiện hiển thị tab Directory (nếu directory đang bị ẩn). Ẩn/Hiện tab Directory

Bước 2: Thực hiện tạo mới Directory

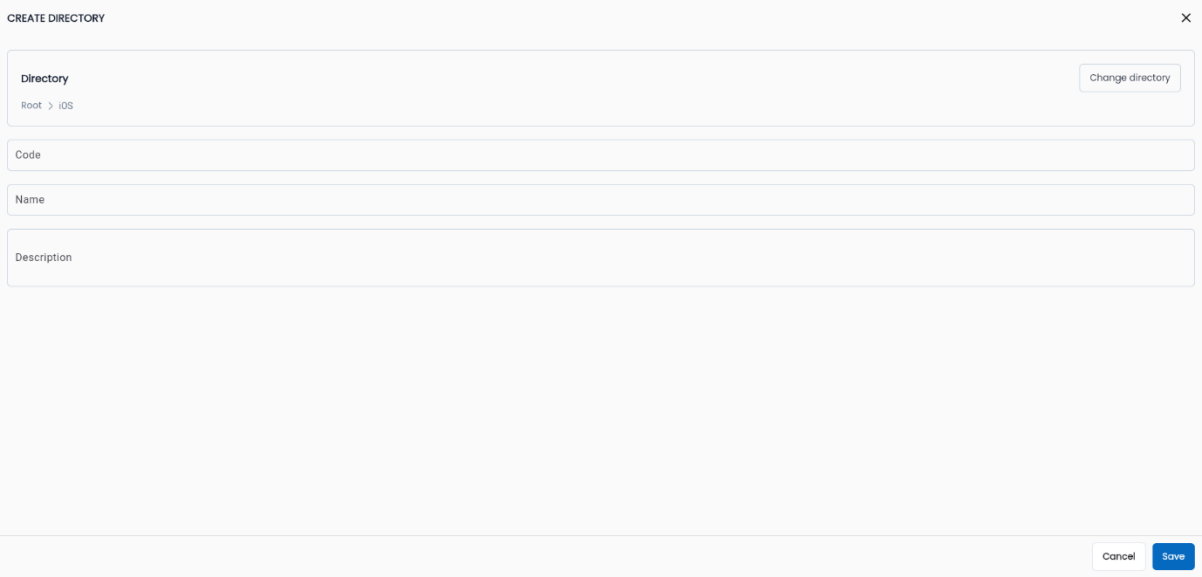
Nhấn “ + ” để tạo mới Directory



Để tạo một directory con (subdirectory) trong một directory đã có sẵn, người dùng nhấn button “ ⋮ ” tại directory muốn tạo subdirectory và chọn “New directory”.



Sau đó, hệ thống hiển thị màn hình “CREATE DIRECTORY”



Bước 3: Nhập các trường thông tin, trong đó:

Trường thông tin	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Directory	Có	Label		Là vị trí directory trong hệ thống. Root là vị trí gốc nếu tạo mới directory không nằm trong 1 directory cha khác.
Code	-	Text		Là mã của directory. Trường thông tin được dùng để phân biệt giữa các directory. Mã có thể tự sinh.
Name	Có	Text	Không được phép đặt trùng tên giữa các directory	Là tên của directory. Người dùng nên đặt theo Module hoặc chức năng của kịch bản kiểm thử để thuận tiện theo dõi, kiểm soát.

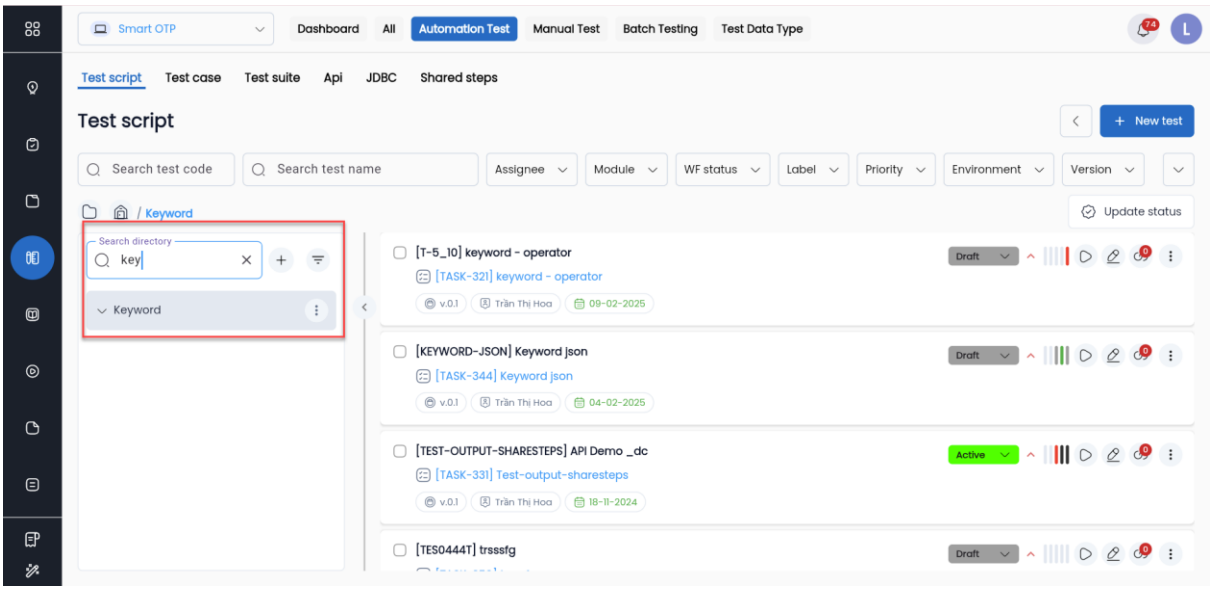
			trong cùng 1 tab	
Description	-	Textarea		Mô tả về Directory, thể hiện nội dung của directory giúp người dùng cũng như những user khác trong hệ thống có thể nắm bắt được tổng thể về directory.
Save	-	Button		Click "Save" để lưu Directory
Cancel	-	Button		Click "Cancel" để huỷ thao tác tạo Directory

Bước 4: Nhấn "**Save**", hệ thống thông báo **"Create Successful"**

1.1.3.3 Tìm kiếm Directory

Bước 1: Thực hiện hiển thị tab Directory (nếu directory đang bị ẩn). Ẩn/Hiện tab Directory

Bước 2: Nhập tên của directory cần tìm vào trường thông tin Search directory, hệ thống sẽ thực hiện tìm kiếm ngay khi người dùng nhập vào trường thông tin tìm kiếm.

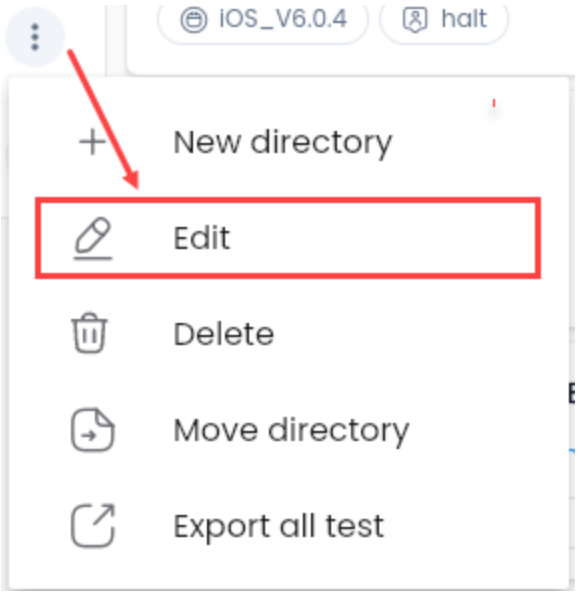


1.1.3.4 Chỉnh sửa Directory

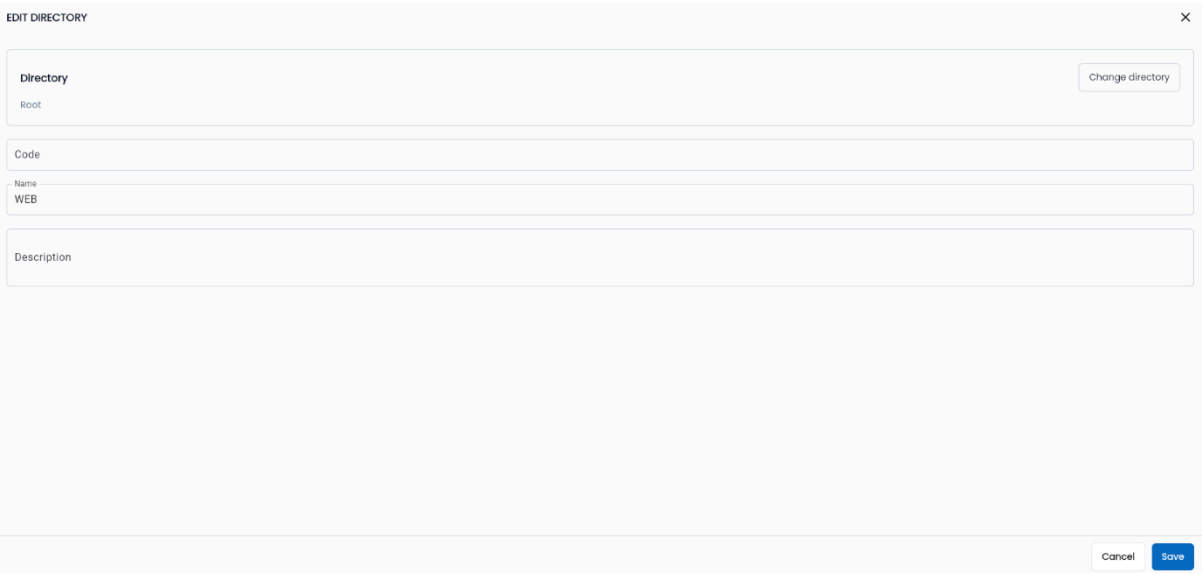
Chỉnh sửa Directory trong tab Test case

Bước 1: Thực hiện hiển thị tab Directory (nếu directory đang bị ẩn). Ẩn/Hiện tab Directory

Bước 2: Nhấn button “ ” trên directory muốn chỉnh sửa, chọn “**Edit**”



Hệ thống hiển thị màn hình **“EDIT DIRECTORY”**



Bước 3: Nhập các trường thông tin, trong đó các trường thông tin tương tự tại màn hình CREATE DIRECTORY. Tạo mới Directory.

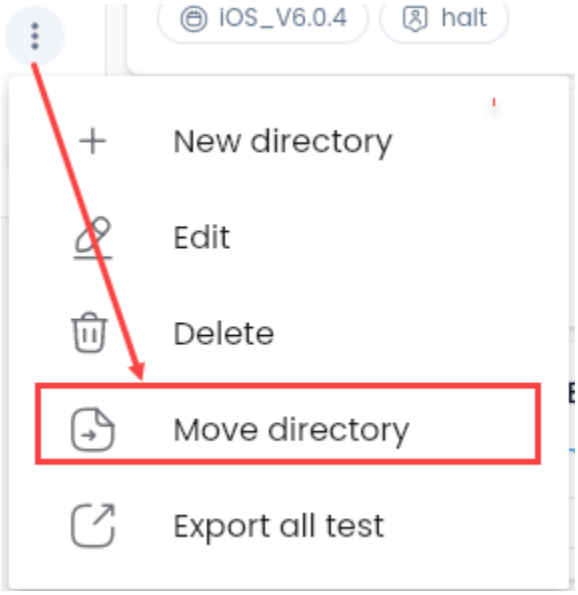
Bước 4: Nhấn "**Save**", hệ thống thông báo **"Update Successful".**

1.1.3.5 Move to directory

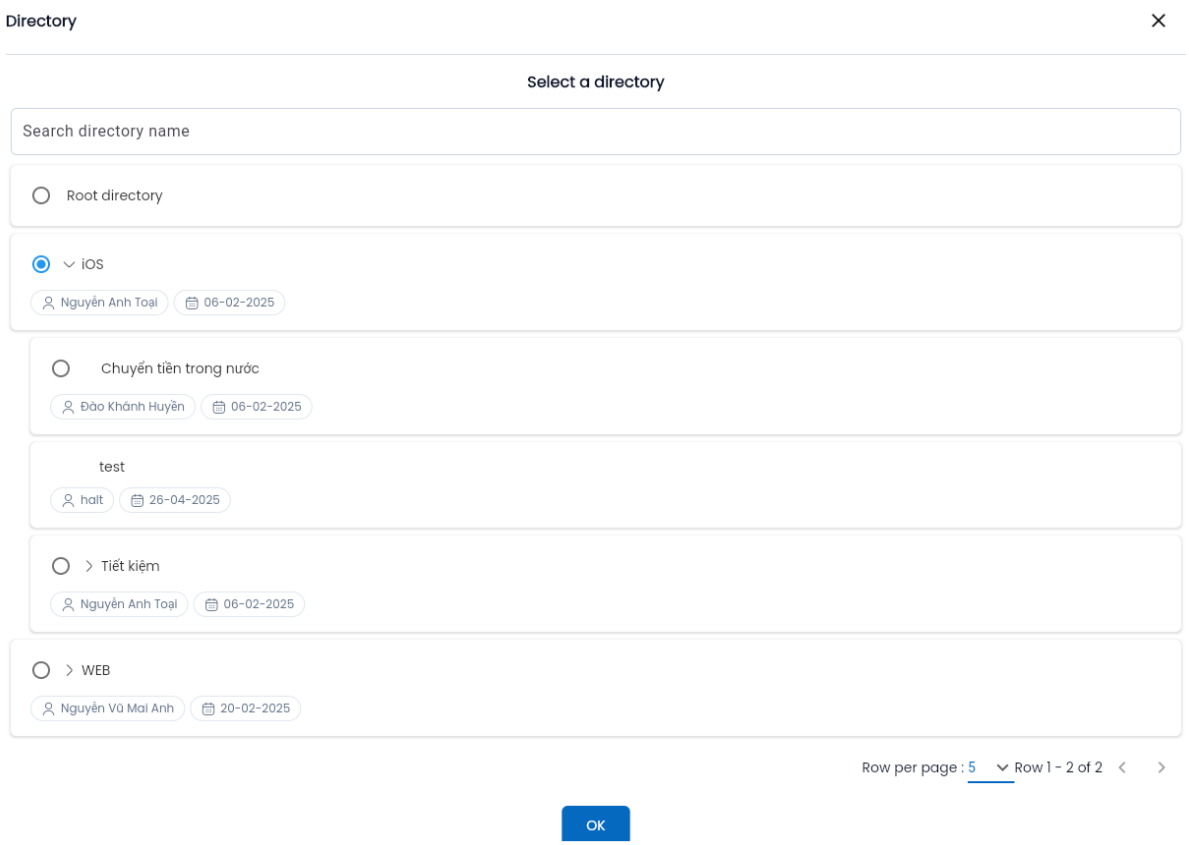
Di chuyển Directory trong tab Test case

Bước 1: Thực hiện hiển thị tab Directory (nếu directory đang bị ẩn). Ẩn/Hiện tab Directory

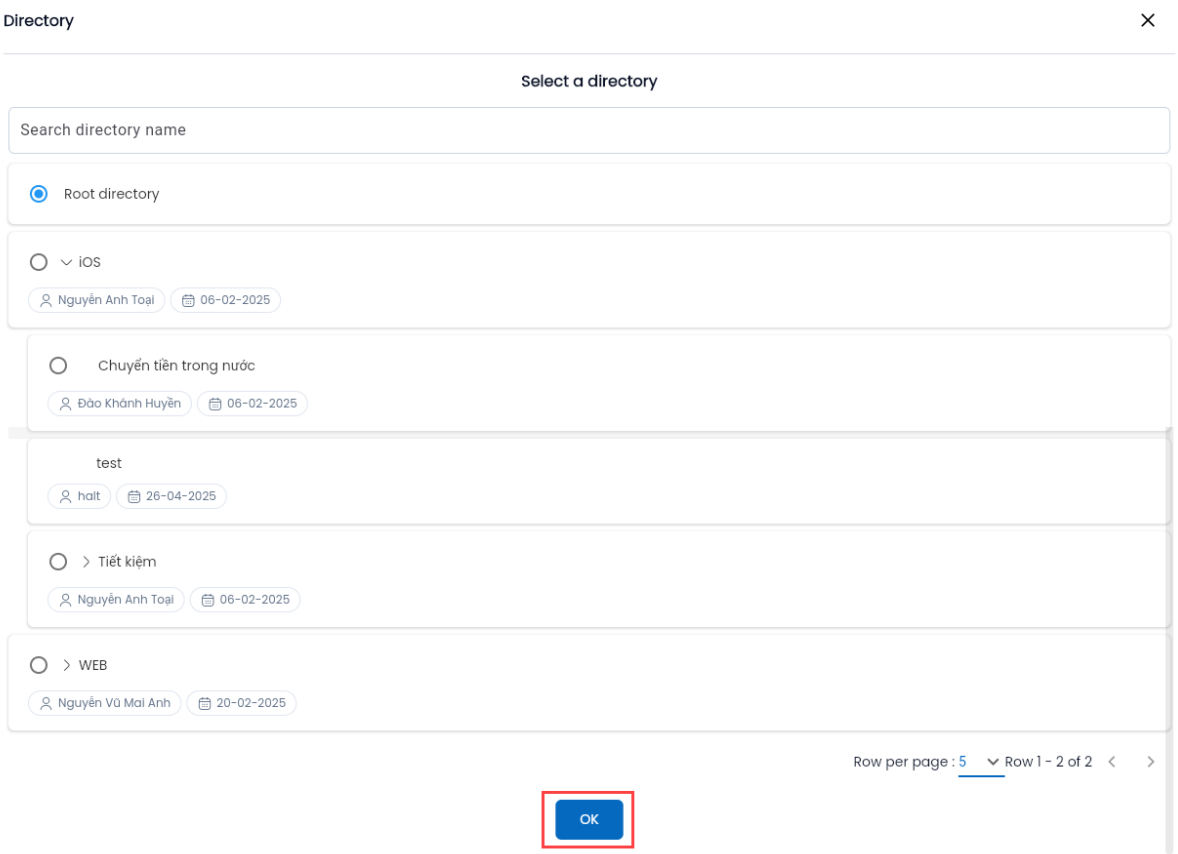
Bước 2: Nhấn button "⋮" trên directory muốn di chuyển vị trí, chọn "**Move directory**"



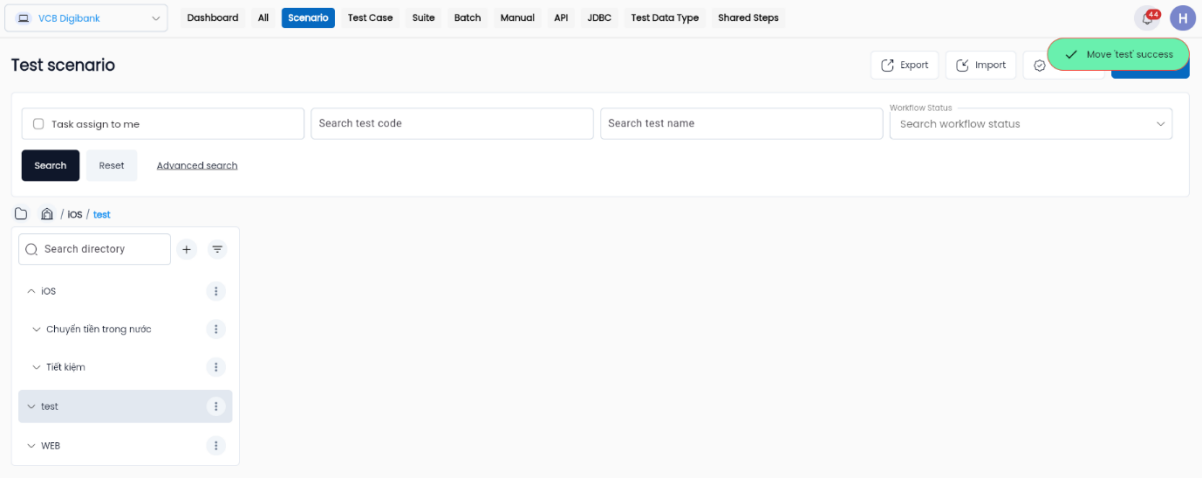
Hệ thống hiển thị màn hình **"Select a directory"** cho phép chọn một directory đích



Bước 3: Chọn một directory đích, nhấn **"OK"**



Sau đó, hệ thống hiển thị thông báo trên màn hình**: "Move "…" success.**

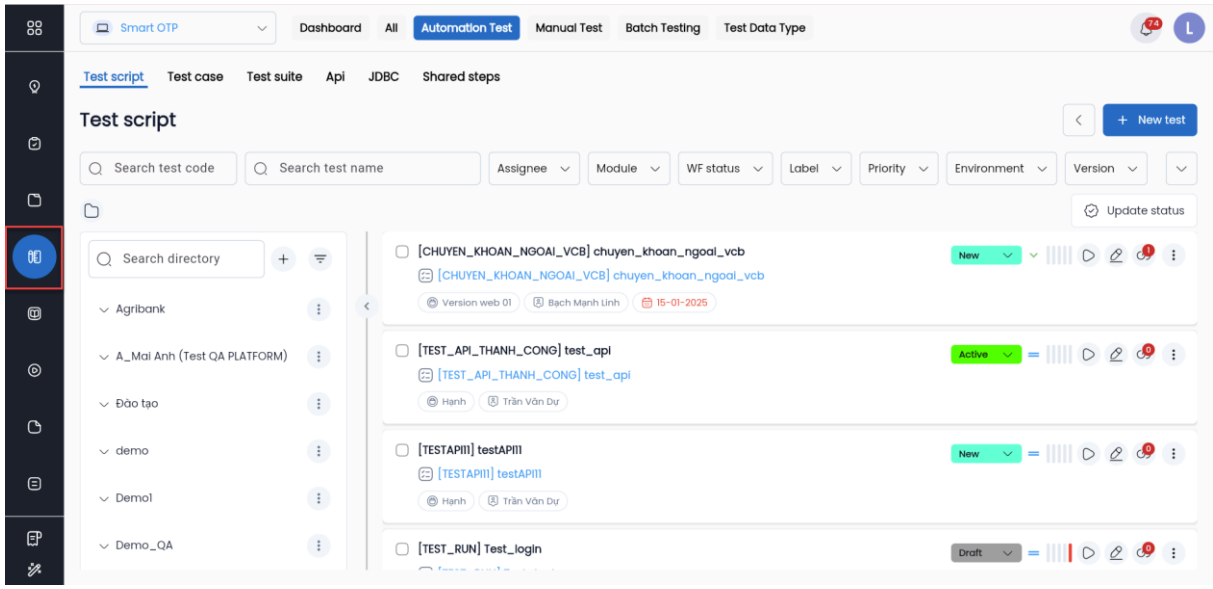


1.1.4 Khởi tạo kịch bản kiểm thử trong LLQ QA Platform

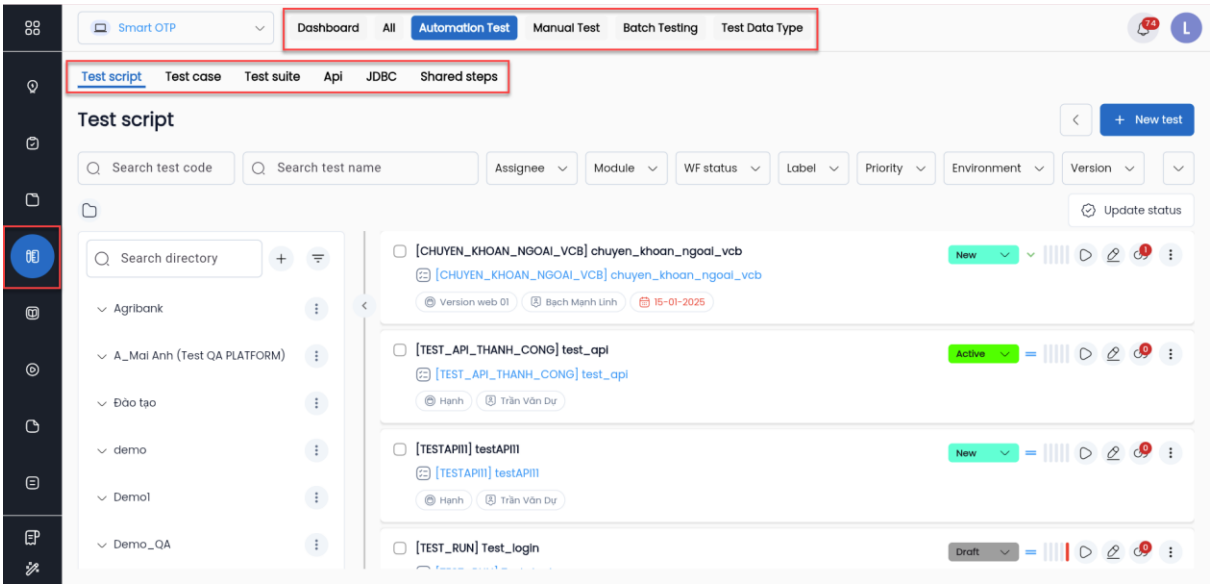
1.1.4.1 Khởi tạo thủ công

Để tạo mới một loại test trực tiếp trên hệ thống, thực hiện như sau:

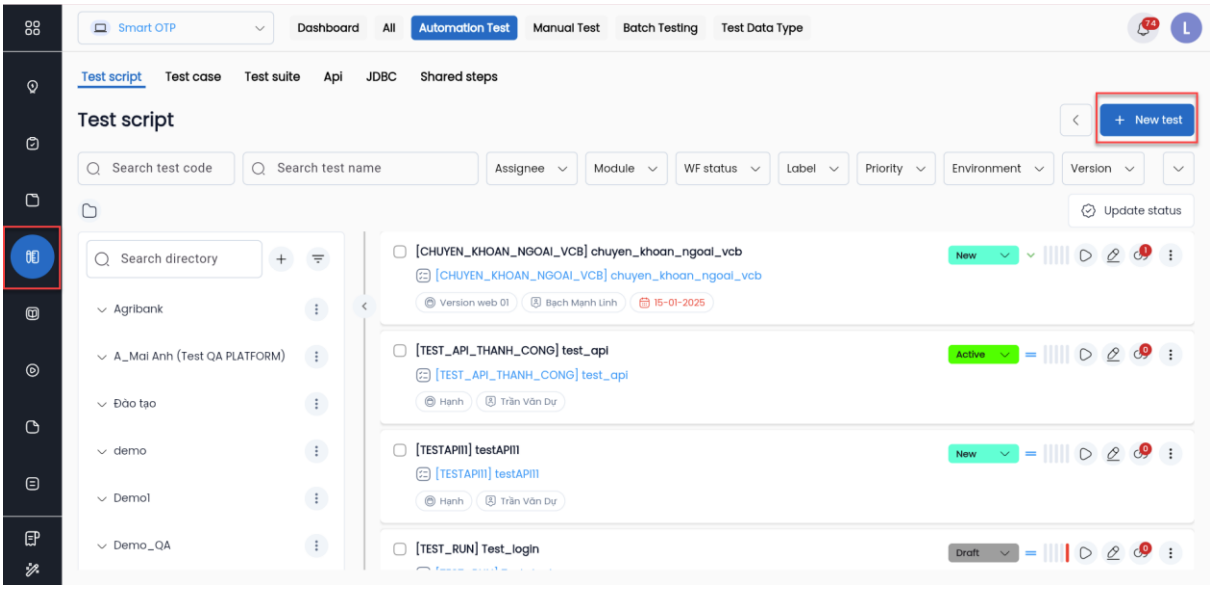
Bước 1: Chọn **Test Management** trên danh sách menu của dự án



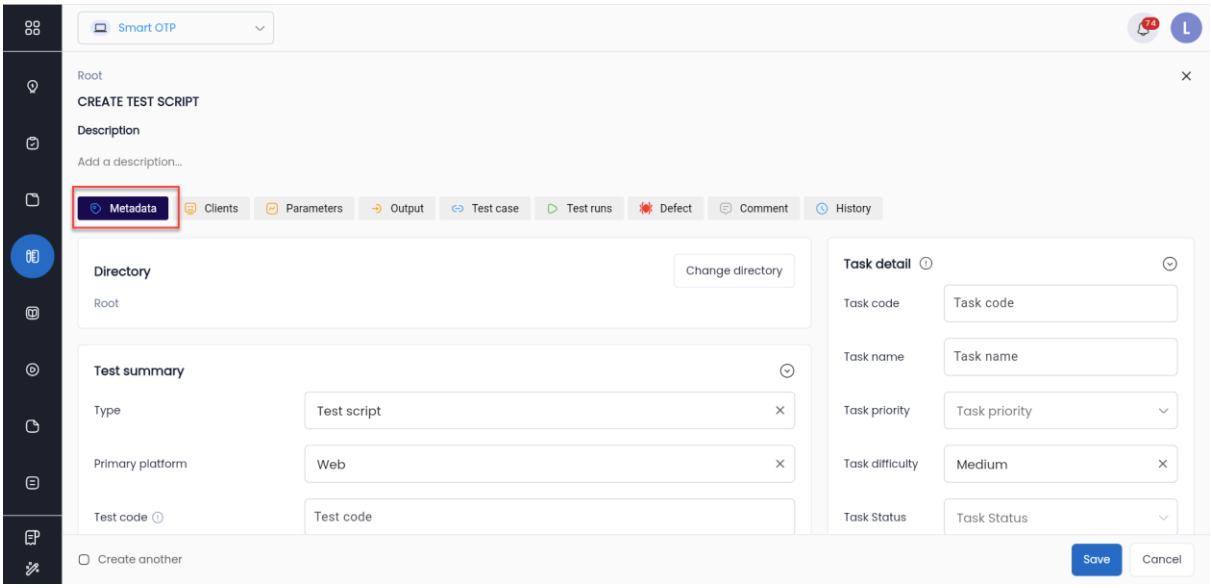
Bước 2: Chọn loại kịch bản kiểm thử mà người dùng muốn tạo mới



Bước 3: Nhấn "**New test**"



Bước 4: Nhập đúng đủ các thông tin vào form tạo mới (tab **Metadata)**



Mô tả các trường thông tin

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Thông tin cơ bản				
Code	-	Textbox	Nếu nhập, người dùng nhập code thì	Là mã dùng để định danh kịch bản kiểm thử trên hệ

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
			code phải là duy nhất, độ dài tối thiểu 2 ký tự, bắt đầu bằng chữ cái, chỉ chứa các ký tự từ a-z (viết thường, viết hoa), chữ số, "_" và "-"	thống Nếu code bỏ trống, hệ thống sẽ tự sinh theo cấu trúc T-stt, trong đó stt là số thứ tự của test trong số các kịch bản kiểm thử được đánh mã tự động của Project
Name	Có	Textbox		Cho phép người dùng nhập tên của test
Directory	-	Button	Hiển thị danh sách các directory thuộc loại test đang chọn	Chọn 1 directory người dùng muốn lưu test. Cách chọn 1 directory
Label	-	Button		Chọn 1 hoặc nhiều nhãn mà người dùng muốn gắn. Cách gắn nhãn
Description	-	Textarea		Cho phép người dùng nhập mô tả
Purpose	-	Textarea		Cho phép người dùng nhập mục đích

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Precondition	-	Textarea		Là điều kiện tiên quyết để có thể thực hiện kiểm thử
Test steps	-	Textarea		Cho phép người dùng nhập các bước thực hiện kiểm thử
Expected result	-	Textarea		Cho phép người dùng nhập kết quả mong đợi
Test data	-	Textarea		Cho phép người dùng nhập dữ liệu kiểm thử
Thông tin cài đặt				
Type	Có	Combobox		Hiển thị giá trị mặc định là loại test mà người dùng muốn tạo
Version	Có	Combobox	Hiển thị danh sách version ở trạng thái active có trong dự án	Cho phép người dùng chọn phiên bản của dự án. Test này thuộc phiên bản nào sẽ chọn phiên bản đó
Environment	Có	Combobox	Hiển thị danh sách environment ở trạng thái active có trong dự án	Cho phép người dùng chọn môi trường phát triển của dự án. Test thuộc môi trường

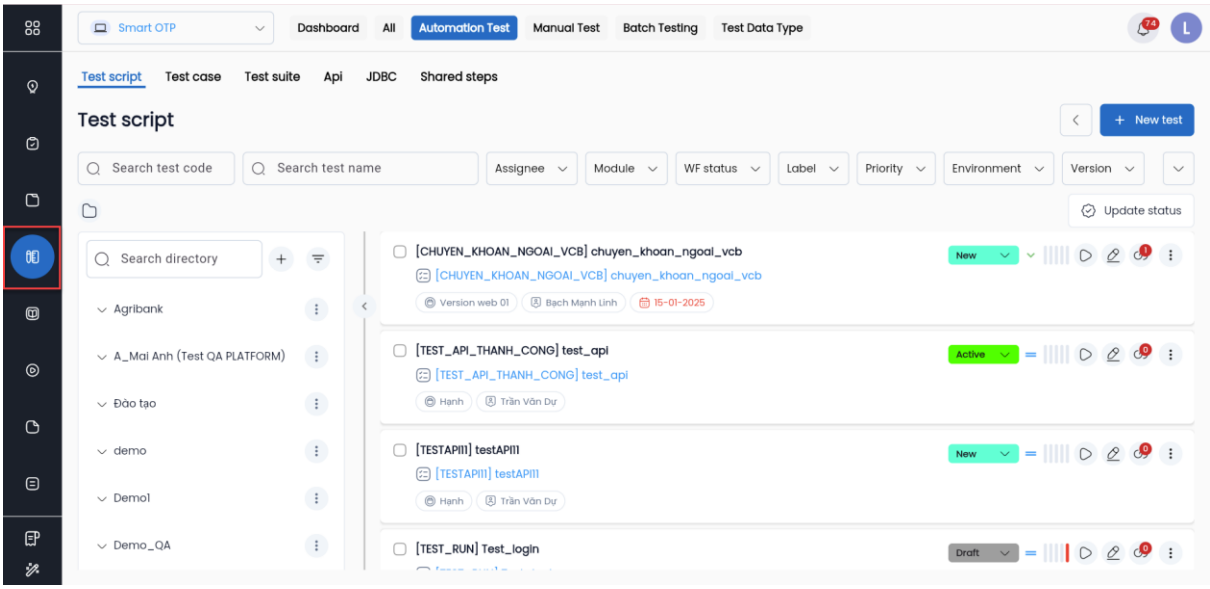
Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				nào sẽ chọn môi trường đó
Module	Có	Combobox	Hiển thị danh sách module ở trạng thái active có trong dự án	Cho phép người dùng chọn 1 module của dự án. Test thuộc module nào sẽ chọn module đó
Test Priority	Có	Combobox		Người dùng chọn 1 trong 5 trạng thái để xác định mức độ ưu tiên của kịch bản test: + Highest + High + Medium + Low + Lowest

Bước 5: Bấm nút "**Save**". Hệ thống hiển thị thông báo "**Create successful**". Test vừa tạo hiển thị trên lưới dữ liệu.

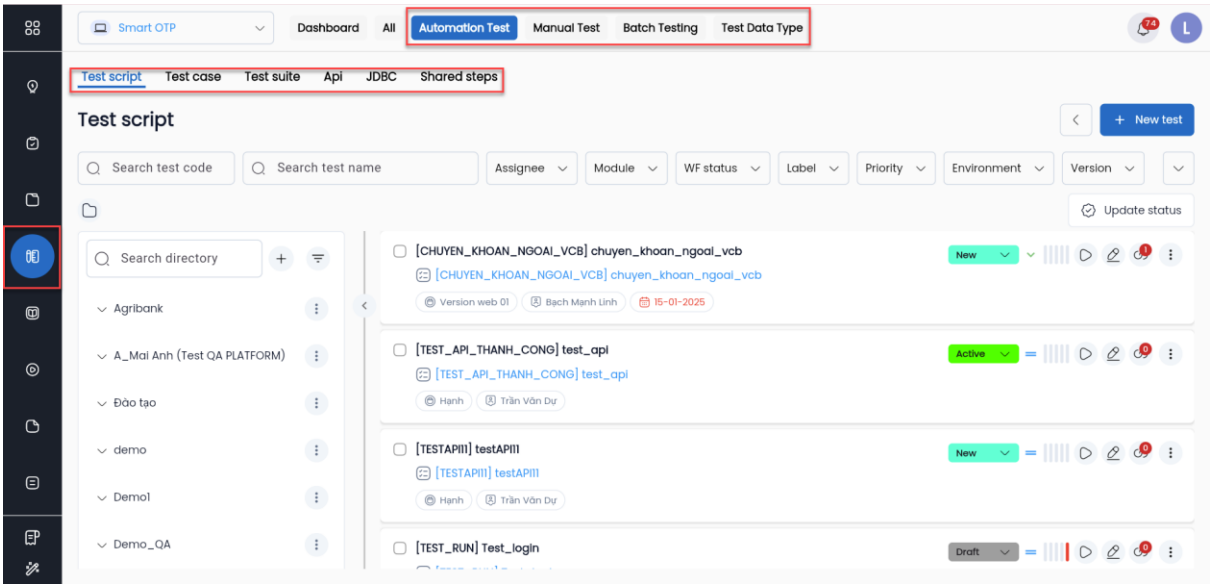
1.1.4.2 Tạo test tự động qua file excel

Để có thể tạo kịch bản kiểm thử 1 cách tự động qua file excel thì đầu tiên phải tải file mẫu có trên hệ thống. Cách làm như sau:

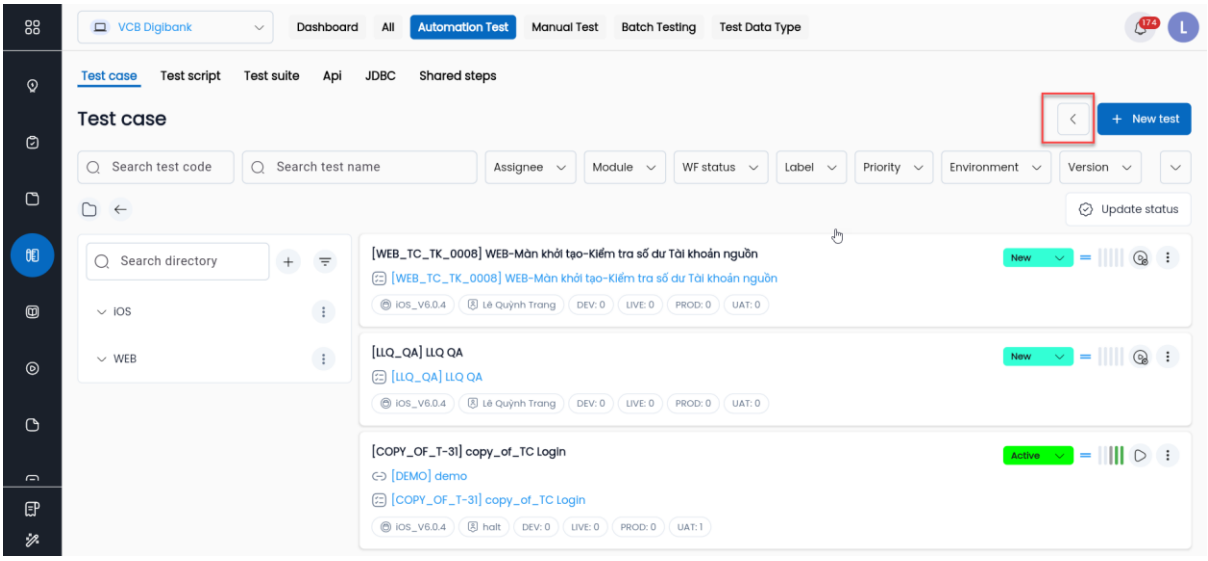
Bước 1: Chọn **Test Management**

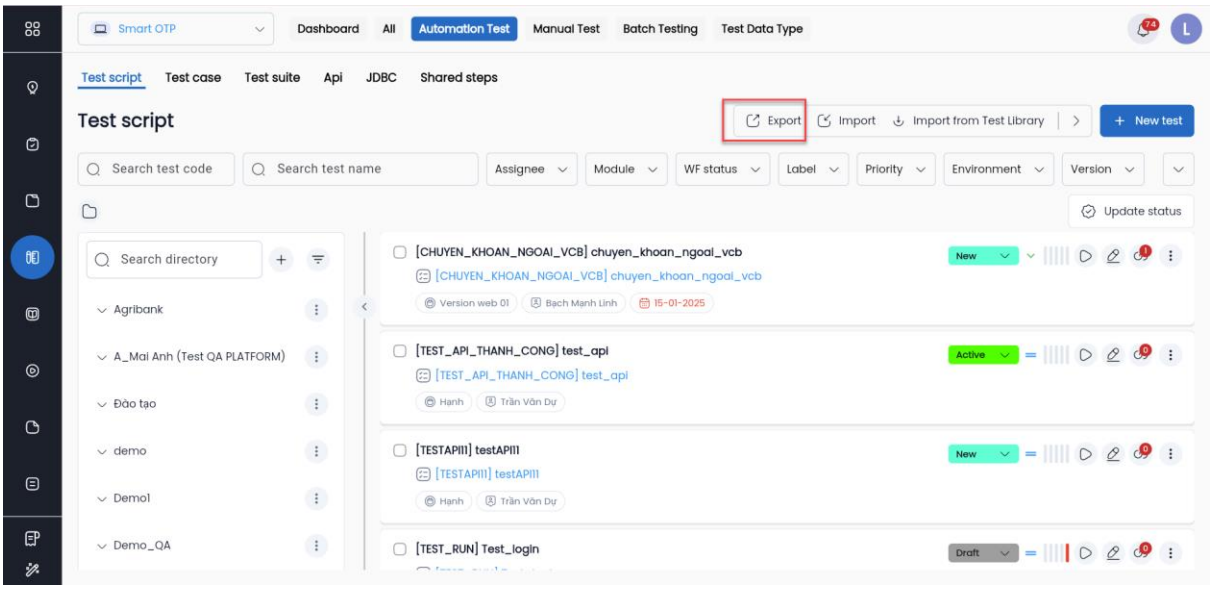


Bước 2: Chọn loại kịch bản kiểm thử người dùng muốn tạo



Nhấn **Show**, sau đó nhấn **Export**





File excel mẫu sẽ được tải về.

Bước 3: Người dùng nhập đầy đủ, chính xáccác thông tin vào file mẫu vừa tải về

- Mô tả thông tin sẽ nhập trong file excel

Tên trường	**Bắt buộc**	**Định dạng**	**Ràng buộc**	**Mô tả**
TYPE	Có	List option		Chọn 1 loại test: + Test case + Test script

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				+ Test manual + Test suite + Test api + Test jdbc + Shared steps
APPLICATION_MODULE	Có	Text	Code của Project module đã được khởi tạo trên hệ thống	Nhập code của cấu phần Project module muốn sử dụng trên hệ thống
CODE	Có	Text	Code là duy nhất	Người dùng nhập mã của test
NAME	Có	Text		Người dùng nhập tên của test
LABELS		Text		Gắn label nào thì nhập code của lable đó trên hệ thống, các label được cách nhau bởi dấu “,”
TEST_PRIORITY	Có	List option		Người dùng chọn 1 trong 5 trạng thái để xác định mức độ ưu tiên của test: + Highest + High + Medium + Low + Lowest
PURPOSE	Không	Text		Người dùng nhập mục đích

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
PRECONDITIONS	Không	Text		Người dùng nhập điều kiện tiên quyết
INPUTS	Không	Text		Người dùng nhập dữ liệu kiểm thử
STEPS	Không	Text		Người dùng nhập các bước thực hiện kiểm thử
EXPECTED_RESULTS	Không	Text		Người dùng nhập kết quả mong đợi
DIRECTORY_PATH	Có	Text		Nhập đường dẫn của thư mục. Nếu chưa có thư mục này trên hệ thống thì thư mục này sẽ được tự động tạo khi import file excel Cú pháp: **/directory/sub-directory/…**
SCRIPT_SUITE_CODE	-	Text	Nếu type là test case thì nhập cột này. Code của script hoặc suite được khởi tạo trên hệ thống trước test case này.	Người dùng nhập mã của test script hoặc test suite mà test case tham chiếu đến. Xem thêm test case

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
SCRIPT_SUITE_NAME	-	Text		Là tên test script/ suite tương ứng với script_suite_code đã nhập
APPLICATION_ENVIRONMENT	Có	Text	Code của Project environment đã được khởi tạo trên hệ thống	Nhập code của cấu phần Project environment muốn sử dụng trên hệ thống
APPLICATION_VERSION	Có	Text	Code của Project version đã được khởi tạo trên hệ thống	Nhập code của cấu phần Project version muốn sử dụng trên hệ thống
WF_STATUS	Có	List option		Người dùng chọn 1 trong 5 trạng thái để xác định tình trạng cũng như tiến độ của test: + Draft: bản nháp, chưa hoàn thiện + Active: đã sẵn sàng để đưa vào sử dụng + Review: đang được xem xét chỉnh sửa lại + Reject: kịch bản đã bị từ chối + Update: kịch bản đang được update

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
TASK_ASSIGNEE	Có	Text		Nhập email user của người được giao task
TASK_PRIORITY	Có	List option		Là mức độ ưu tiên thực hiện của task + Highest: Cao nhất + High: Cao + Medium: Trung bình + Low: Thấp + Lowest: Thấp nhất
TASK_DIFFICULTY	Có	List option		Là mức độ khó yêu cầu thực hiện của task + Very hard: Rất khó + Hard: Khó + Medium: Trung bình + Easy: Dễ + Very easy: Rất dễ
START_DATE	-			
DUE_DATE	-			
DESCRIPTION	-	Text		Người dùng nhập mô tả

Bước 4: Sau khi đã nhập đầy đủ thông tin, bấm nút **IMPORT** và chọn file excel danh sách test. Hệ thống sẽ tạo ra các test có trong danh sách

1.1.5 Test script

1.1.5.1 Test script là gì?

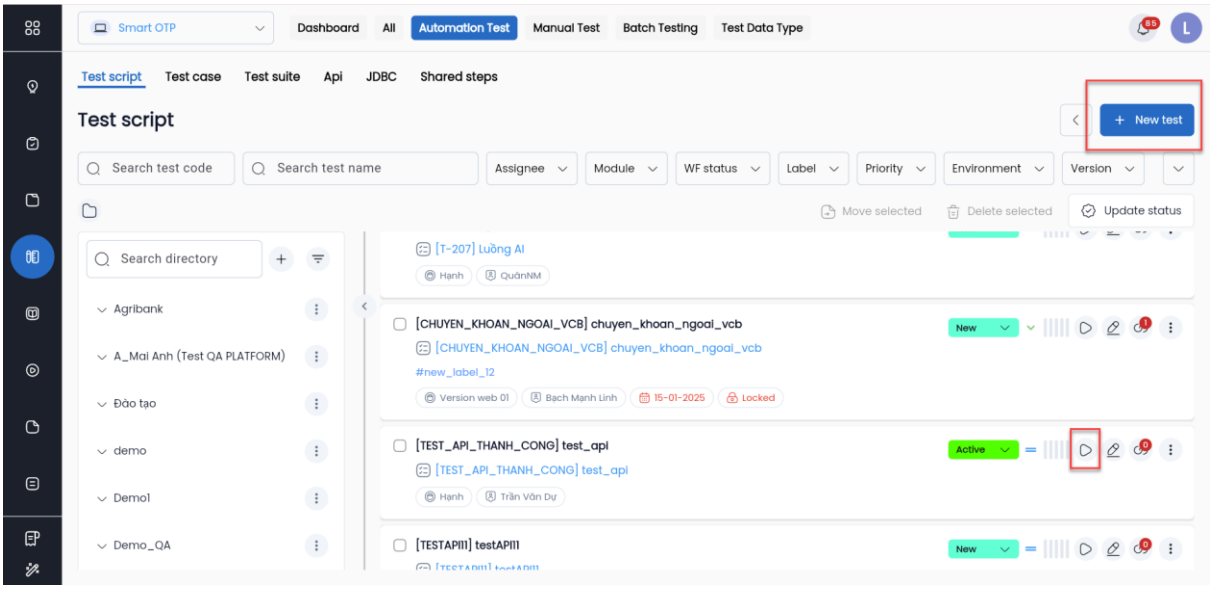
Test script là một dạng kịch bản kiểm thử bao gồm quy trình tiêu chuẩn thực hiện test một cách chi tiết.

Một test script có thể bao gồm rất nhiều test case.

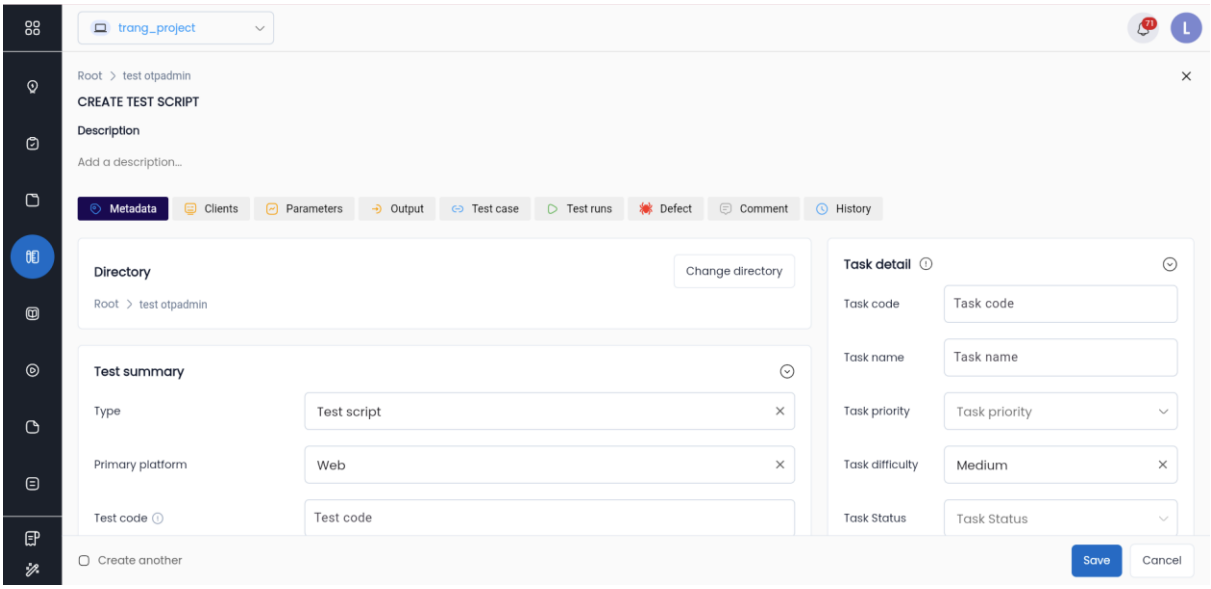
- Các chức năng trong test script

- Edit metadata: Chỉnh sửa thông tin metadata của test script
- Delete: Xoá test script
- Duplicate: Nhân bản test script
- Run: Thực thi test script
- Import: Dùng để thay thế hoặc tạo cấu hình mới của test script từ 1 file json
- Export: Tải về json file của test script
- Move to directory: Di chuyển test script tới một thư mục khác
- Find links: Tìm kiếm các test case tham chiếu đến script.
- Lock: Khóa test script

1.1.5.2 Tạo mới một test script



Bước 1: Trên màn hình tab Test script, Nhấn "**New test**", hệ thống hiển thị màn hình metadata



Màn hình Metadata của test script

Bước 2: Nhập đầy đủ và hợp lệ cho các trường thông tin, trong đó:

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
General information				
Directory	-	Button	Nếu Directory = Root thì test case thuộc directory gốc	Hiển thị danh sách các directory thuộc project ở tab Test Script cho phép người dùng chọn 1 directory cho test case
Type	Có	Combobox		Mặc định là test script
Primary platform	Có	Combobox		Hiển thị danh sách các primary platform có trong hệ thống, cho phép người dùng chọn 1 giá trị.
Test Code	Có	Text	Code phải là duy nhất và không thể chỉnh sửa,	Cho phép người dùng nhập mã của test case, là thông tin để phân biệt các test với nhau

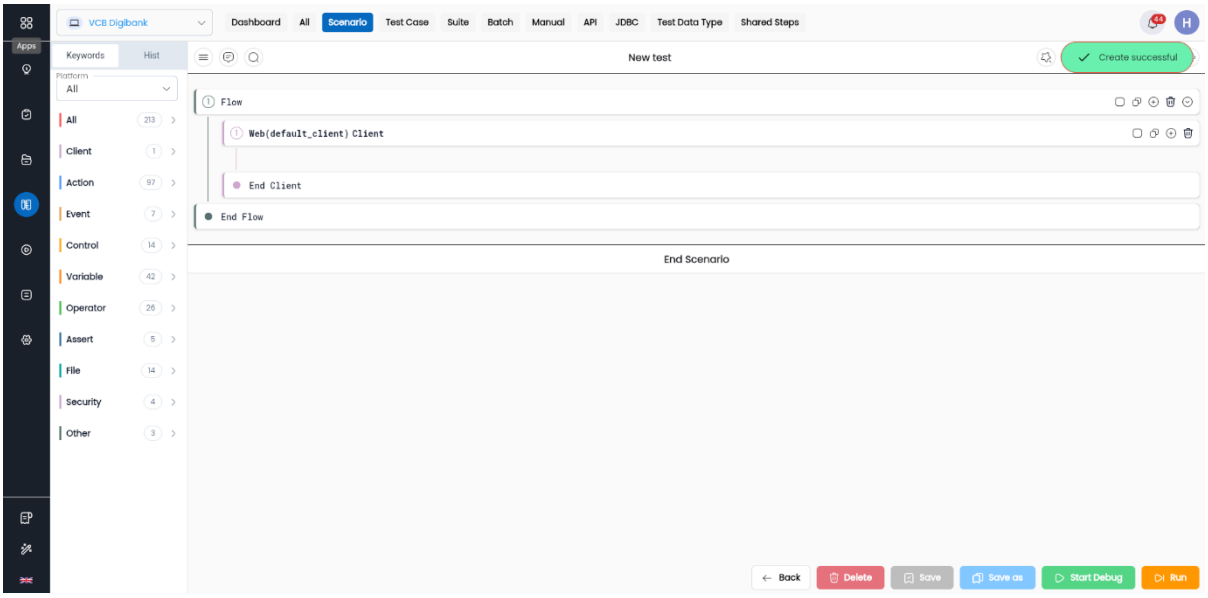
Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
			bắt đầu bằng chữ cái và không chứa kí tự đặc biệt ngoài "_"	
Test Name	Có	Text		Cho phép người dùng nhập tên của test script.
Test priority	Có	Combobox		Mức độ ưu tiên thực hiện của test + Highest: Cao nhất + High: Cao + Medium: Trung bình + Low: Thấp + Lowest: Thấp nhất
Environment	Có	Combobox	Danh sách environment ở trạng thái active	Hiển thị danh sách environment có trong hệ thống ở trạng thái active, cho phép người dùng chọn 01 giá trị
Module	Có	Combobox	Danh sách Module ở trạng thái active	Hiển thị danh sách module có trong dự án ở trạng thái active, cho phé người dùng chọn 01 giá trị
Version	Có	Combobox	Danh sách version ở trạng thái active	Hiển thị danh sách các version có trong hệ thống, cho phép người dùng chọn 1 giá trị. Chi tiết về version
Label	Có	Combobox	Danh sách label ở trạng thái active	Hiển thị danh sách các label có trong hệ thống, cho phép người dùng chọn 1 giá trị.

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Purpose	-	Text area		Cho phép người dùng nhập mục đích của test script
Precondition	-	Text area		Cho phép người dùng nhập điều kiện tiên quyết của test script
Test steps	-	Text area		Cho phép người dùng nhập các bước thực hiện trong test script
Expect result	-	Text area		Cho phép người dùng nhập kết quả mong muốn của test script
Test data	-	Text area		Cho phép người dùng nhập dữ liệu test của test script
Label	-			Click vào chức năng quản lý label, hiển thị màn hình danh sách các label đã có trong hệ thống. Click label muốn gán cho shared steps
Task Code	-	Text		Code của task tương ứng với test
Task Name	Có	Text		Là tên của task tương ứng với test được thực hiện
Task Priority	Có	Combobox		Là mức độ ưu tiên thực hiện task + Highest: Cao nhất + High: Cao + Medium: Trung bình + Low: Thấp + Lowest: Thấp nhất

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Task Status	-	-		Status mặc định dựa theo cấu hình test script mới tạo
Task difficulty	Có	Combobox		Mức độ khó trong yêu cầu thực hiện của test + Very Hard: Rất khó + Hard: Khó + Medium: Trung bình + Easy: Dễ + Very easy: Rất dễ
Assign to	-			Lựa chọn người thực hiện test Nếu không chọn, mặc định test được thực hiện bởi người tạo
Scheduled start date	-			Ngày bắt đầu của task tương ứng với test
Due Date	-			Ngày kết thúc của task tương ứng với test
Parameter	-			Cho phép người dùng truyền tham số cho test script
Create another				Giúp người dùng tạo cùng lúc nhiều test script khác nhau mà không phải thoát ra màn hình chính.
Clients	Có			Nhập nền tảng mà test script được xây dựng. Hệ thống mặc định tạo tự động clients cho nền tảng Web

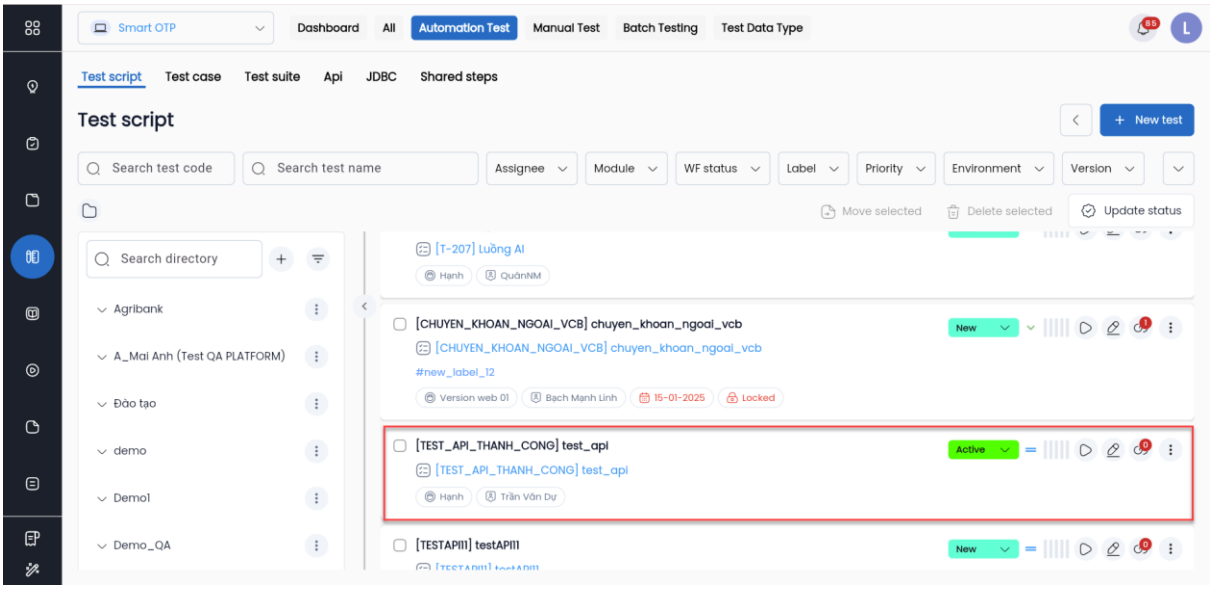
Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Output	-			Là tham số mà người dùng muốn hiển thị ở kết quả chạy đầu ra của test script
Comment			Cần phải tạo test script trước khi comment	Trong quá trình review kịch bản kiểm thử, người dùng có thể để lại ý kiến để chỉnh sửa cấu hình được kịp thời.
History				Cho phép người dùng xem toàn bộ nhật ký chỉnh sửa và cập nhật của test script, giúp quản lý và kiểm soát thay đổi trong suốt vòng đời kiểm thử.

Bước 3: Nhấn "**Save**", hệ thống hiển thị màn hình cấu hình, đồng thời thông báo "Create successful". Kịch bản kiểm thử được lưu trên hệ thống.

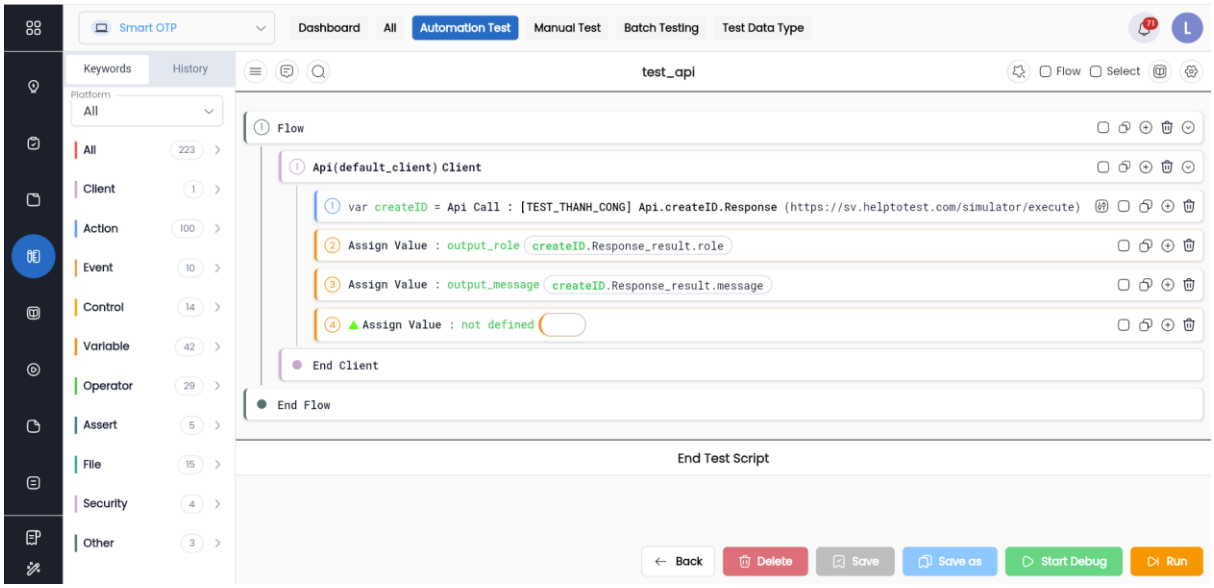


1.1.5.3 Cấu hình kịch bản kiểm thử

Nhấn chọn bản ghi kịch bản kiểm thử người dùng muốn thực hiện cấu hình

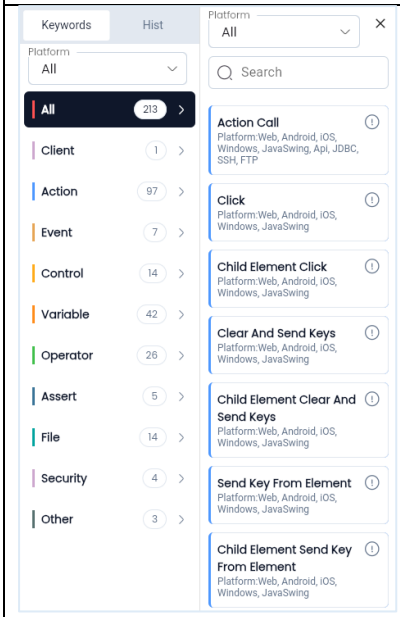


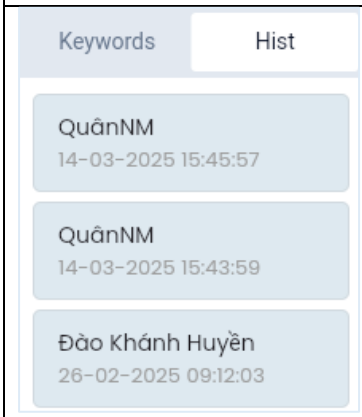
Hệ thống hiển thị màn hình *Low-code*

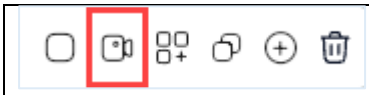
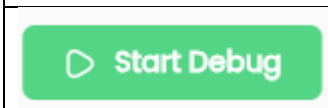
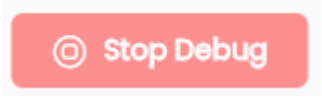
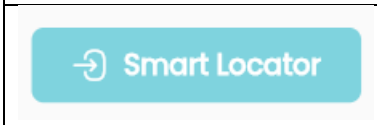
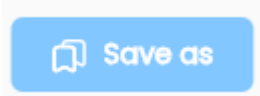


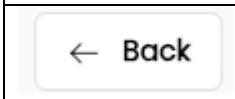
Mô tả trường thông tin

Chức năng	Mô tả
Demo1	Tên kịch bản kiểm thử
	Ẩn/ hiển thị danh sách keyword
	Comment: Chức năng cho phép người dùng (bao gồm toàn bộ user được phân quyền sử dụng kịch bản kiểm thử) bình luận công khai.

	Search: Chức năng cho phép người dùng tìm kiếm các keyword đã sử dụng trong kịch bản kiểm thử
	Play step by Step: Chức năng cho phép người dùng debug toàn bộ kịch bản kiểm thử hoặc một nhóm các keyword tuỳ ý, từ đó kiểm tra tính đúng đắn về logic, thuật toán của kịch bản kiểm thử trước khi thực thi
Flow	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 kịch bản kiểm thử (nếu có) mà không cần xem toàn bộ cấu hình
Select	Khi bật về chế độ On: có thể chọn 1 hoặc nhiều các keywords.
	Chức năng cho phép người dùng chuyển sang màn hình metadata của kịch bản kiểm thử từ màn hình *low-code*
	Undo – Redo: thực hiện quay lại phiên bản thao tác trước đó
	Keyword: các hành động, điều kiện kiểm soát cho phép người dùng xây dựng thuật toán thực hiện test steps của kịch bản kiểm thử, 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

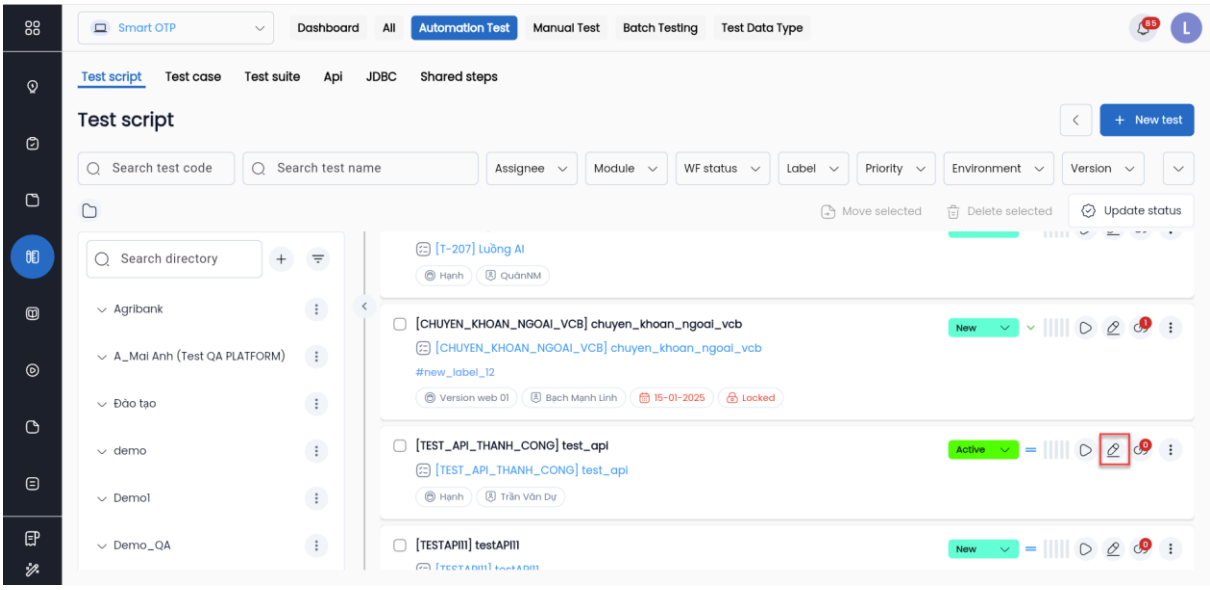
	File: Các hành động tương tác với File trên hệ thống kiểm thử Security: keyword về bảo mật Other: Loại khác, không ảnh hưởng đến steps của kịch bản kiểm thử
	Hist: Chức năng cho phép người dùng xem lại toàn bộ lịch sử chỉnh sửa cấu hình của test kịch bản kiểm thử, từ đó có thể tái sử dụng một số câu lệnh đã bị xoá/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 kịch bản kiểm thử, từ đó kiểm tra được tính đúng đắn của action Chỉ hiển thị khi đã start debug
	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?
	Copy: Chức năng cho phép người dùng sao chép 1 flow, group, control, action... trong cùng kịch bản kiểm thử hoặc kịch bản kiểm thử khác (thao tác trên cùng 01 tab của trình duyệt) từ một phiên bản history bất kỳ của kịch bản kiểm thử đó.
	Add same type: 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.
	Delete: Chức năng cho phép người dùng xoá 1 flow, group, control, action… của kịch bản kiểm thử

	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 kịch bản kiểm thử theo flow, client, group, control… Expand: Chức năng cho phép người dùng hiển thị toàn bộ action, control đã bị thu gọn trước đó của kịch bản kiểm thử
	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 các element xuất hiện trên màn hình của client.
	Save as: Chức năng cho phép người dùng lưu script hiện tại thành một script mới mà vẫn giữ nguyên kịch bản kiểm thử cũ (một hình thức duplicate)
	Save: Lưu những thay đổi của kịch bản kiểm thử bao gồm phần cấu hình và metadata Chỉ *enable* khi cấu hình có sự thay đổi
	Delete: Chức năng cho phép người dùng xoá 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
	Quay lại màn hình thư viện kịch bản kiểm thử.

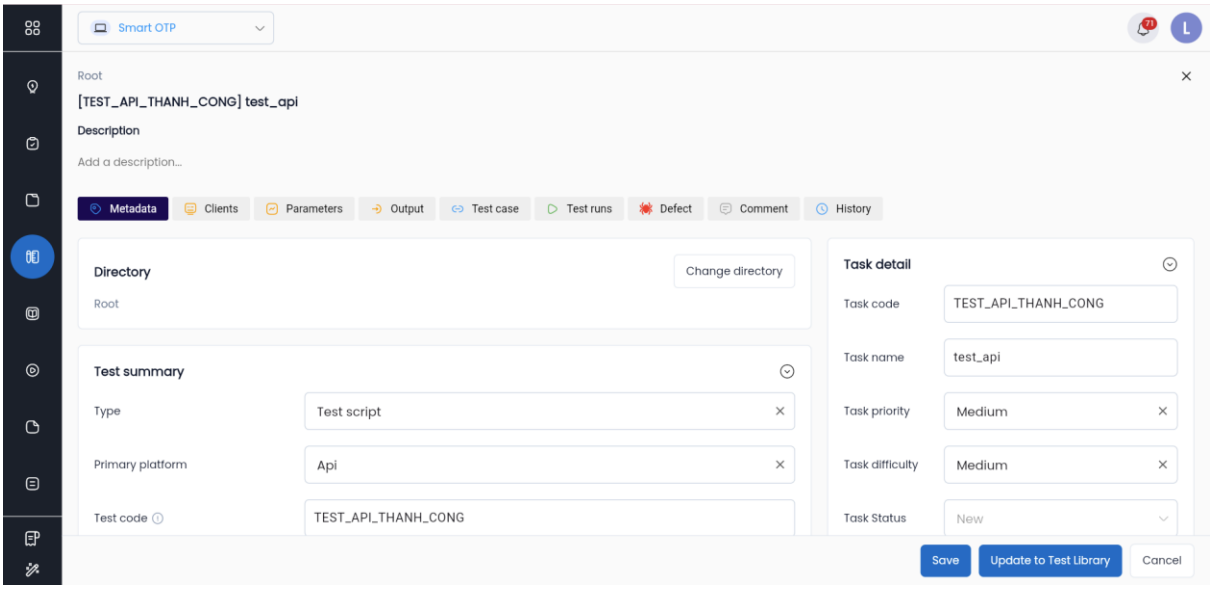
1.1.5.4 Chỉnh sửa Test Script

Bước 1: Tại màn hình thư viện kịch bản kiểm thử., nhấn “ ” (Edit metadata) của ở phía bên phải bản ghi kịch bản kiểm thử muốn chỉnh sửa thông tin.



Hệ thống điều hướng đến màn hình chỉnh sửa thông tin kịch bản kiểm thử. Các trường thông tin giống với màn hình thêm mới.

Lưu ý: Trường **Code** không cho phép chỉnh sửa.



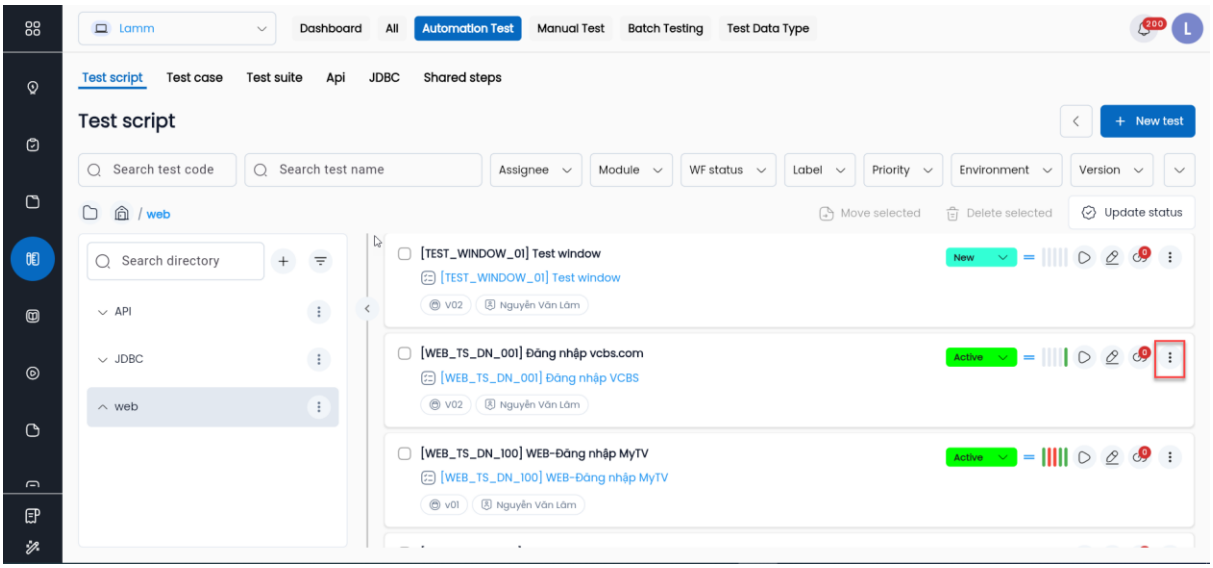
Bước 2: Tiến hành cập nhập thông tin metadata muốn thay đổi của kịch bản kiểm thử. Tạo mới một test script.

Bước 3: Nhấn **"Save.** Hệ thống hiển thị thông báo "Update successful".

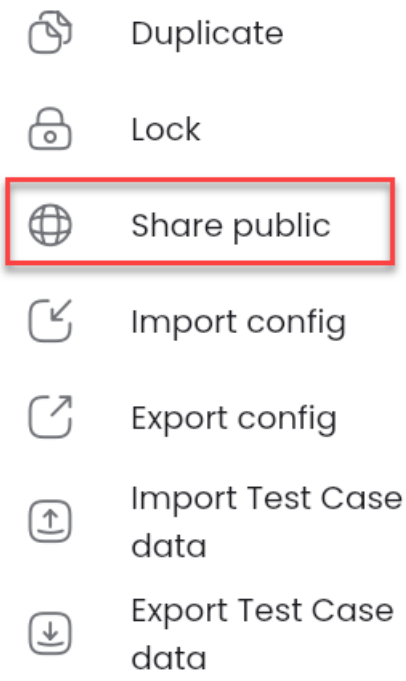
1.1.5.5 Share public

"Share pulic" là chức năng giúp người dùng thêm một Test Script vào Test Lib. Chỉ có thể áp dụng với Test Script có trạng thái là Active.

Bước 1: Nhấn Option ở phía bên cạnh bản ghi Test script



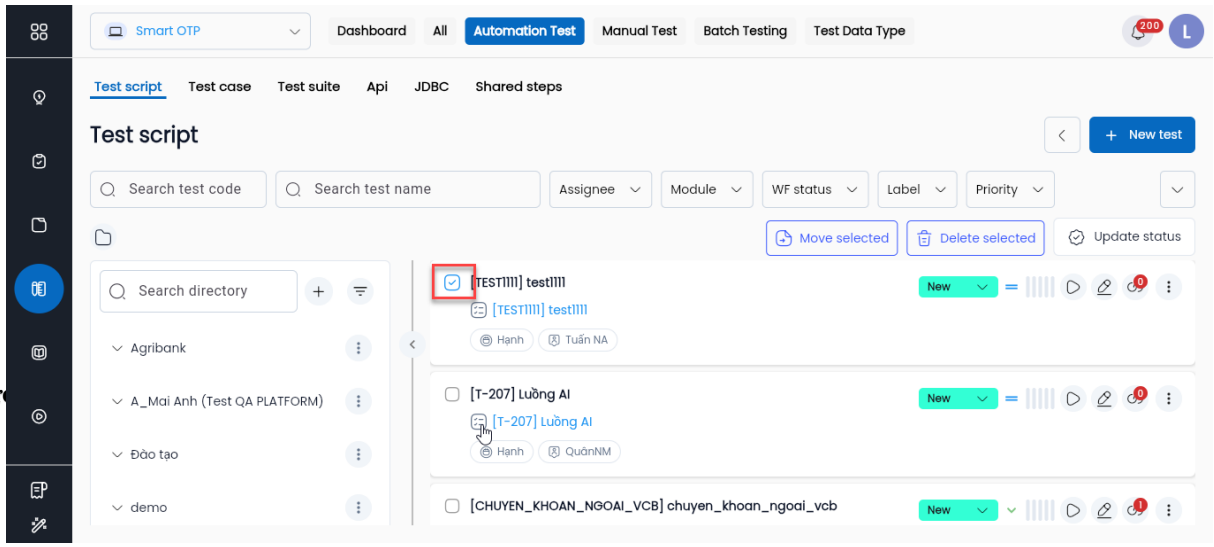
Bước 2: Chọn Share public



Hệ thống hiển thị thông báo thành công, Test script được thêm vào Shared Bussines Script. Thao tác tương tự để loại bỏ Test script khỏi Shared Bussines Script.

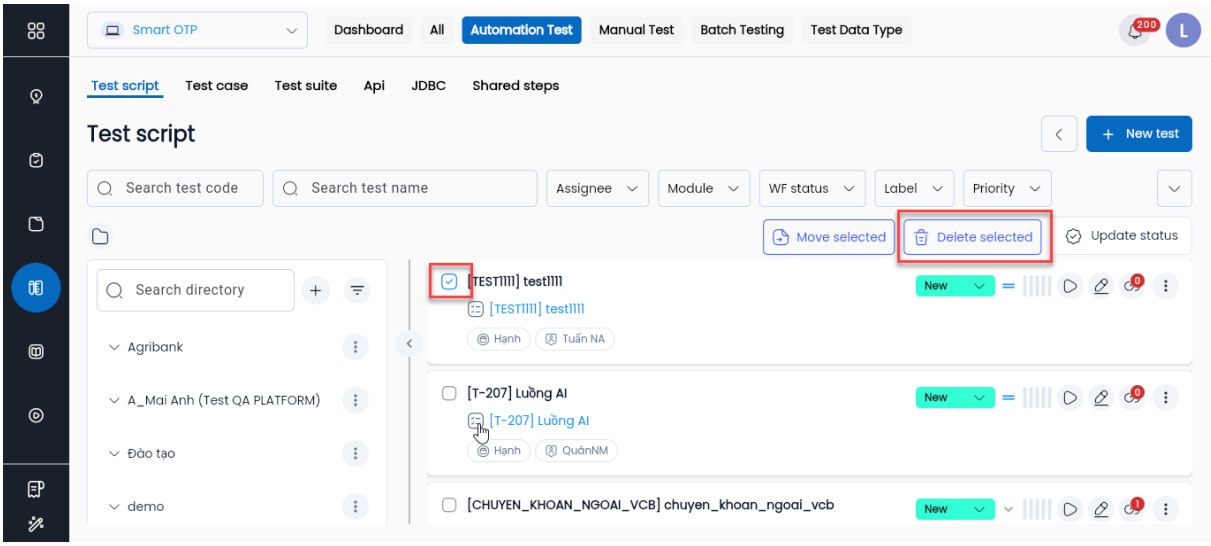
1.1.5.6 Xóa Test Script

Bước 1: Tại màn hình thư viện các kịch bản kiểm thử, chọn Test script bạn muốn xóa



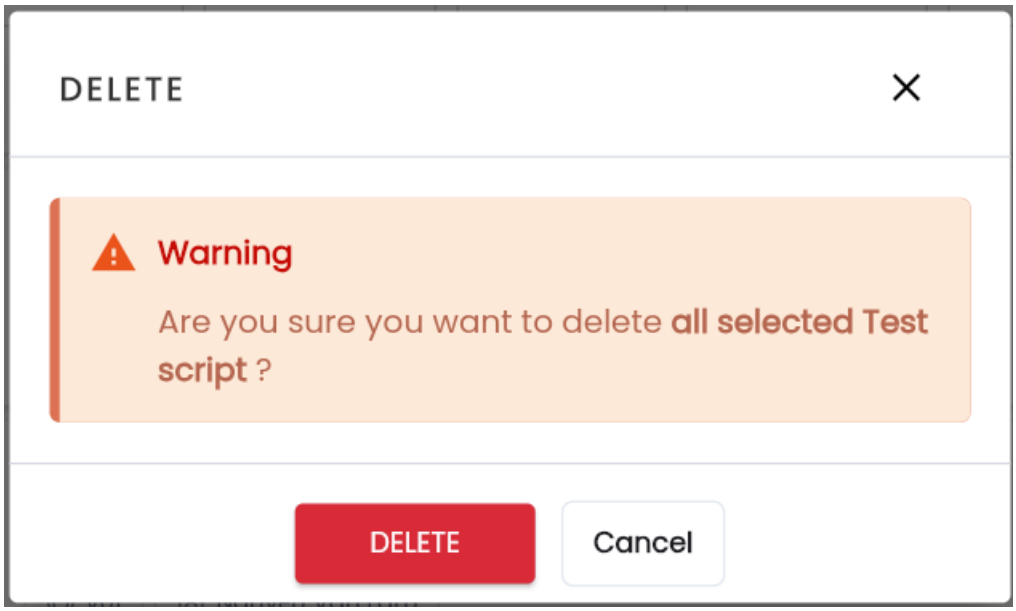
Bư

Nhấn **Delete selected**



Hệ thống hiển thị popup xác nhận xóa

Bước 3:

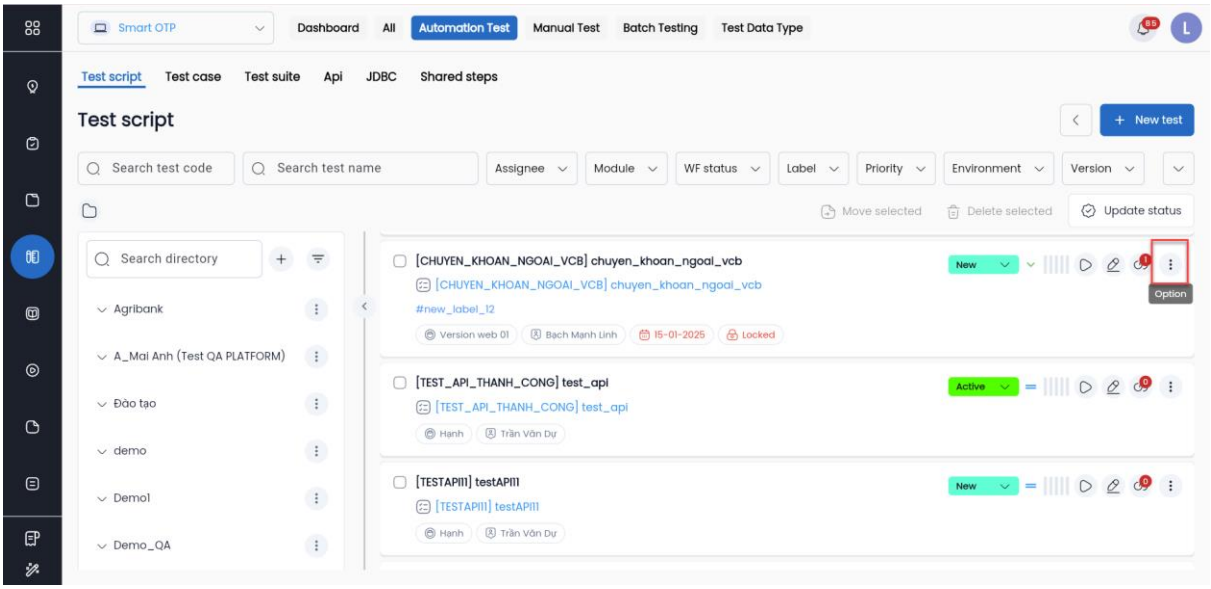


Nhấn "**DELETE**", hệ thống hiển thị thông báo xóa thành công khỏi hệ thống.

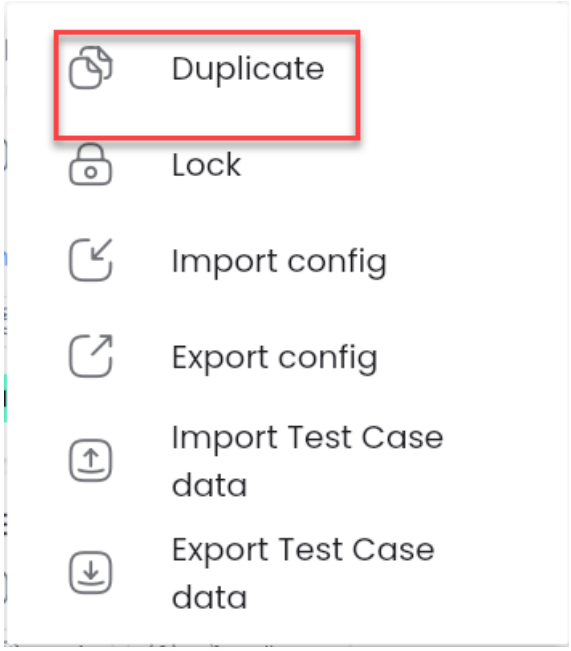
1.1.5.7 Nhân bản kịch bản kiểm thử

Sao chép kịch bản kiểm thử từ thư viện các kịch bản kiểm thử. Khi sao chép một kịch bản kiểm thử thì mọi thông tin cả về cấu hình lẫn mô tả đều được sao chép.

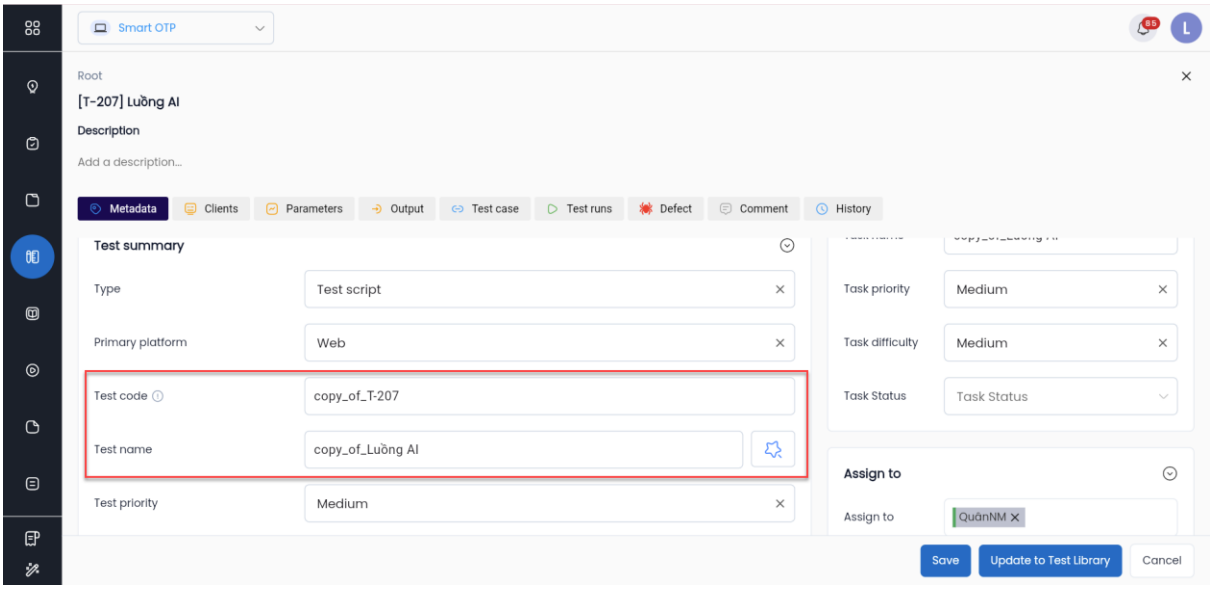
Bước 1: Tại màn hình thư viện các kịch bản kiểm thử. Nhấn "⋮" ở phía bên phải kịch bản kiểm thử muốn nhân bản,



Bước 2: Chọn "**Duplicate**"



Hệ thống hiển thị màn hình thể hiện các thông tin của kịch bản kiểm thử với mã và tên của kịch bản kiểm thử được thay đổi có định dạng mới:

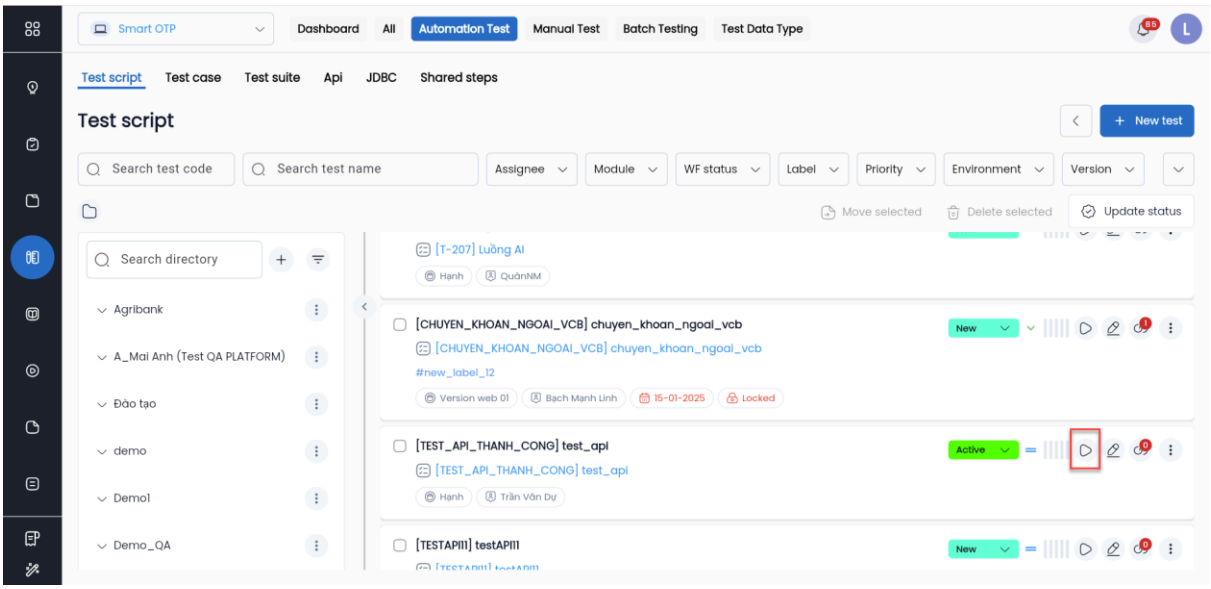


Bước 3: Nhấn "**Save",** hệ thống thông báo đã nhân bản thành công kịch bản kiểm thử.

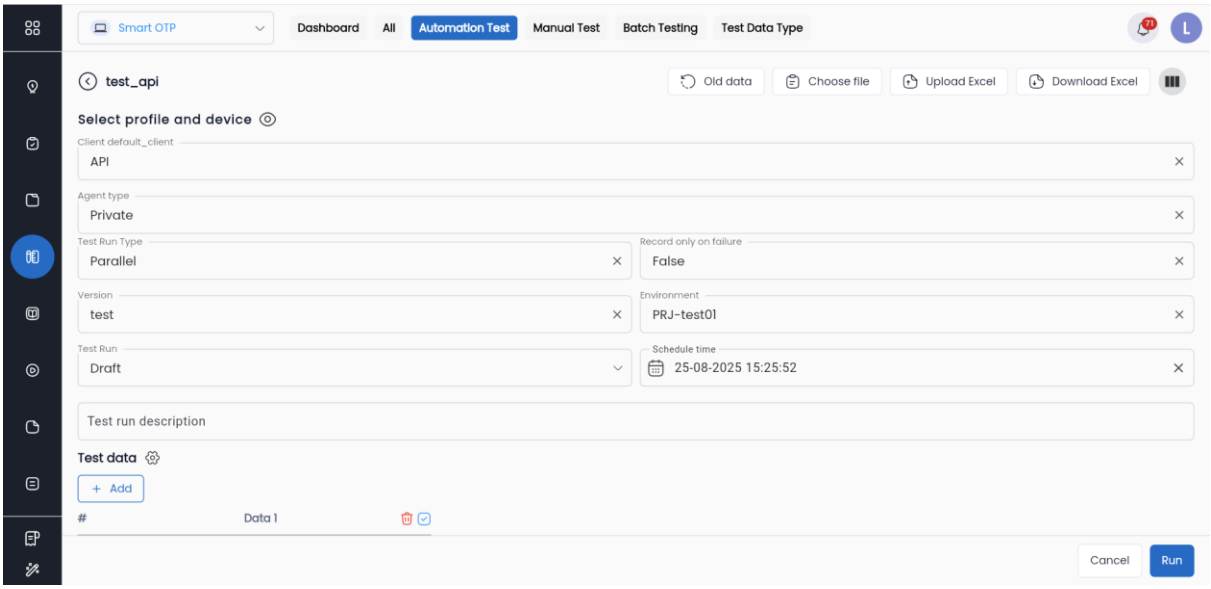
1.1.5.8 Thực thi kịch bản kiểm thử

Cách thực thi kịch bản kiểm thử từ thư viện các kịch bản kiểm thử:

Bước 1: Nhấn " " ở phía bên phải bản ghi kịch bản kiểm thử muốn thực thi



Bước 2: Hệ thống hiển thị màn hình thiết lập thông tin cho kịch bản kiểm thử



Mô tả các trường thông tin:

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Nút chức năng				
Old data	-	Button		Nút chức năng sẽ hiển thị lại danh sách lần Run trước để có thể dùng lại bộ dữ liệu từ lần Run trước.
Choose File	-	Button		Nút chức năng hiển thị màn hình của File Storage chứa các bộ dữ liệu đã được upload lên hệ thống.
Upload Excel	-	Button		Nút chức năng giúp người dùng tải lên bộ dữ liệu đã được chuẩn bị bằng file Excel.
Download Excel	-	Button		Nút chức năng cho phép người dùng tải về file excel mẫu có chứa

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				các trường thông tin cần dùng cho lần run test script.
	-	Button		Điều chỉnh kiểu hiển thị các trường thông tin Test data trong màn hình theo hàng ngang hoặc hàng dọc.
Select profile and device				Nút chức năng điều chỉnh ẩn/ hiện phần thông tin về nền tảng của lần Run test script
+ Add	-	Button		Nút chức năng giúp tạo thêm 1 dữ liệu test case trong lần chạy
	-	Button		Nút chức năng giúp xóa dữ liệu test case trong lần chạy
	-	Checkbox		Nút chức năng giúp chọn các dữ liệu test case mà người dùng muốn chạy
Thông tin trường thông tin				
Client default_client	Có	Combobox		Hiển thị danh sách các Profile đã được khai báo tại chức năng Projects Setting phù hợp với Client của test script.

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				Chỉ được chọn 01 giá trị
Test Run Type	Có	Combobox		Cho phép người dùng chọn 1 trong 2 giá trị: Parallel (Kiểm thử đồng thời). Sequence (Kiểm thử tuần tự).
Agent Type	Có	Combobox		Chọn type Agent sẽ thực thi test: Public: Hệ thống sẽ lựa chọn các Agent public đang active trên hệ thống có cùng thiết lập với profile đã chọn Private: Agent của user đang thực thi kịch bản kiểm thử AnyAgent: Hệ thống sẽ chọn bất kỳ Agent nào (bao gồm public và Agent private) có cùng profile đã chọn Người dùng được phép chọn 1.
Record only on failure		Combobox		Cho phép người dùng chọn một trong hai giá trị: + True + False

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Version	Có	Combobox		Hiển thị danh sách các version có trạng thái active của dự án. Lựa chọn phiên bản của hệ thống/ứng dụng cần test mà kịch bản kiểm thử sẽ được thực thi Cho phép lựa chọn 01 giá trị.
Environment	Có	Combobox		Hiển thị danh sách các environment có trạng thái active của dự án. Là môi trường phát triển sản phẩm mà kịch bản kiểm thử sẽ được thực thi. Cho phép lựa chọn 01 giá trị.
Test Run	Có	Combobox		Cho phép người dùng chọn 1 trong 2 giá trị cho thông tin Agent: + Draft + Complete Sau khi test script Run hoàn tất, dữ liệu Run theo "Draft" có thể chuyển thành "Complete" tùy theo mục đích người dùng nhưng ngược lại thì không được phép.

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Schedule time	Có	DateTime		Là thời gian sẽ thực thi kịch bản kiểm thử. Hệ thống mặc định là realtime, người dùng có thể lựa chọn một thời điểm khác trong tương lai để cài đặt lịch trình thực thi test.
Test run description	-	Text		Mô tả cho lần chạy test
Test case	-	Text		Nhập tên của test case
Test Run Expected Result	Có	Combobox		Cho phép người dùng chọn 1 trong 2 giá trị kết quả test case của lần Run Test Script: + Pass + Fail
Severity	Có	Combobox		Hiển thị 05 mức độ nghiêm trọng, cho phép người dùng chọn 1 giá trị: + Critical (Nghiêm trọng) + Major (Cao) + Average (Trung bình) + Minor (Thấp) + Trivial (Không hẳn là lỗi và cũng không ảnh hưởng đến trải

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				nghiệm của người dùng)
Priority	Có	Combobox		Cho phép người dùng chọn 1 trong 5 giá trị mức độ ưu tiên của kịch bản kiểm thử: + Highest + High + Medium + Low + Lowest

Bước 3: Nhấn "**Run**"

Sau khi Nhấn nút "Run" hệ thống hiển thị màn hình thực thi, đồng thời tự động ghi lại quá trình chạy các bước đã cấu hình trên hệ thống test.

Tham chiếu Kết quả thực thi.

1.1.5.9 Thực hiện chức năng multi-select

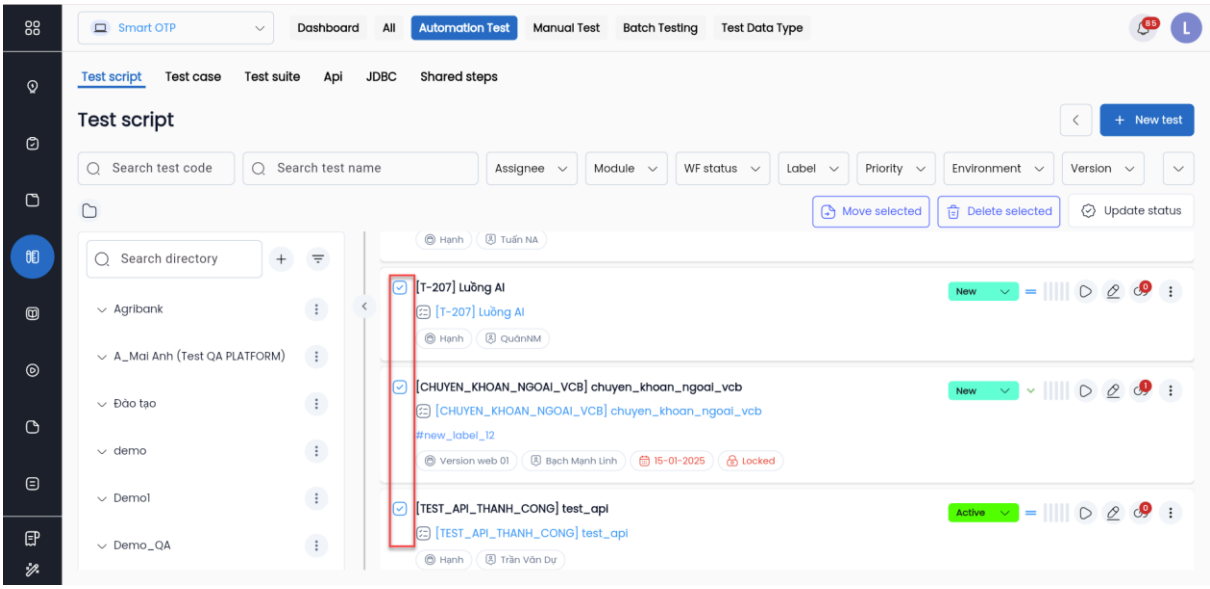
Chức năng **chọn nhiều (multi-select)** giúp người dùng thao tác cùng lúc với nhiều test script, tiết kiệm thời gian khi cần di chuyển, thay đổi trạng thái, hoặc xóa hàng loạt.

1.1.5.9.1 Thực hiện chức năng di chuyển hàng loạt test script vào bên trong thư mục.

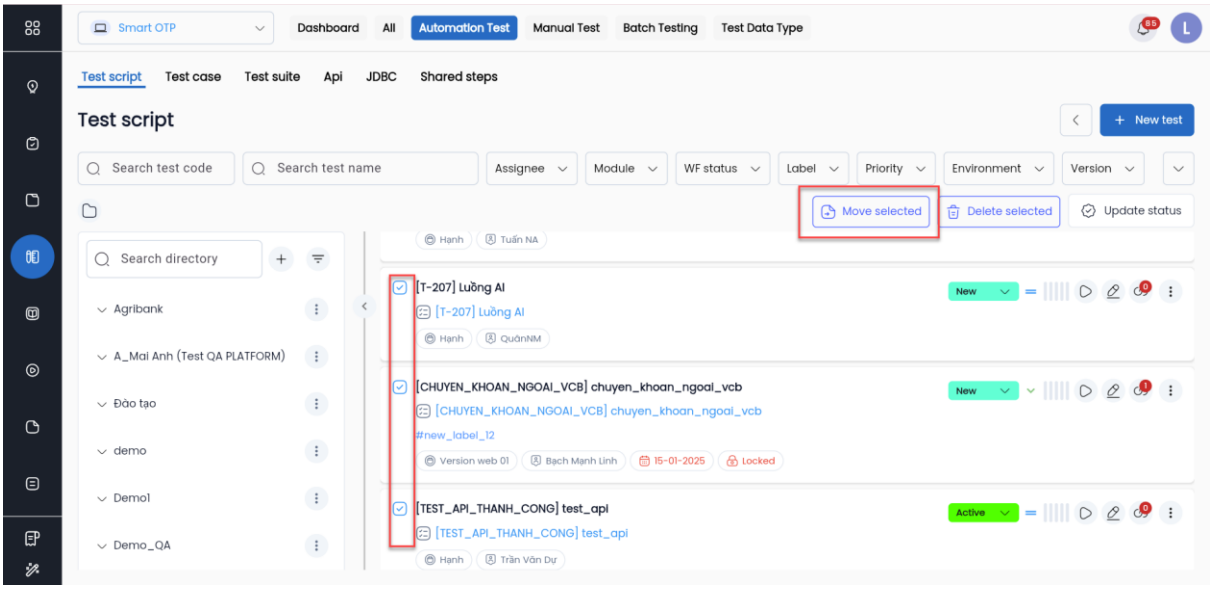
Khi chọn nhiều test script rồi bấm nút này, hệ thống sẽ cho phép bạn **chuyển toàn bộ test script đã chọn** sang một thư mục khác.

Các bước thực hiện:

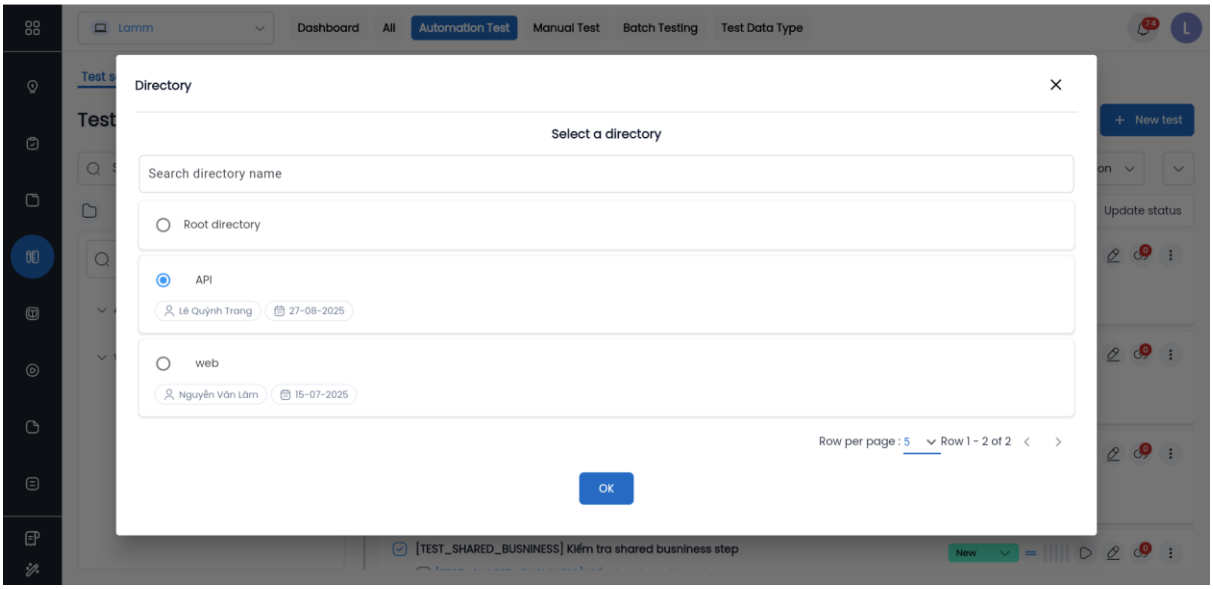
Bước 1: Thực hiện chọn các test script bạn muốn chuyển vào/ sang một thư mục khác.



Bước 2: Click vào “ ” để lựa chọn thư mục bạn muốn chuyển vào

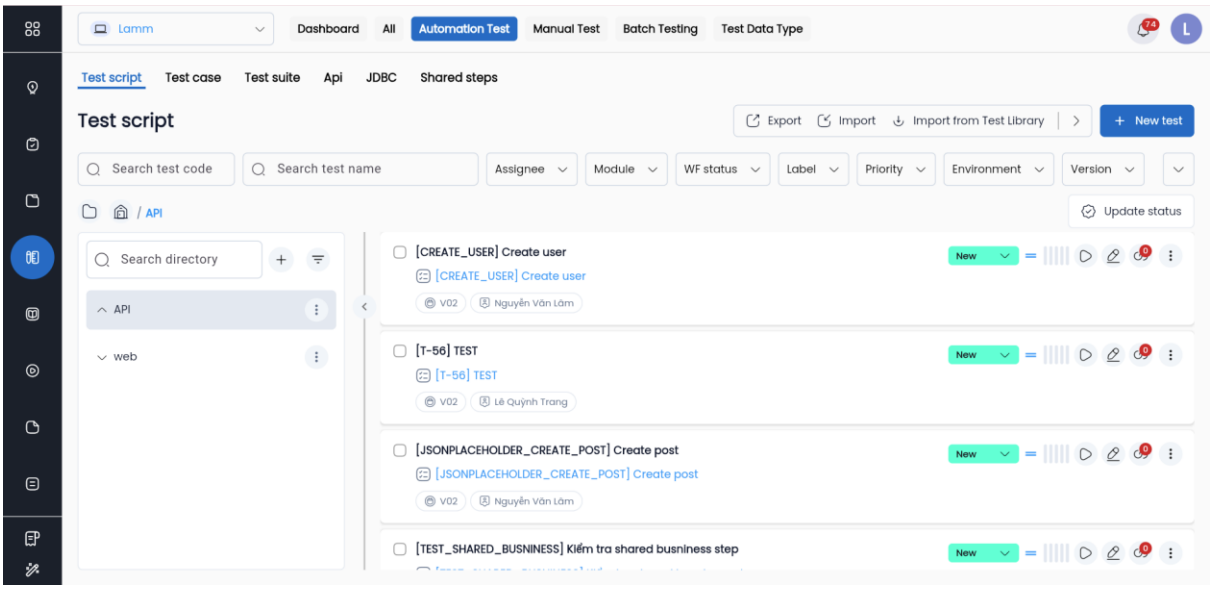


Màn hình hiển thị danh sách các thư mục đã được tạo, cho phép người dùng chọn thư mục đích để lưu các test script đã chọn.



Bước 3: Click "OK", màn hình hiển thị thông báo "Move success"

Các test script đã được chọn sẽ được chuyển vào thư mục mà bạn vừa chỉ định. Nếu thư mục đó đã có sẵn các test script khác, thì tất cả sẽ được gom lại trong cùng một thư mục.



1.1.5.9.2 Xóa hàng loạt test script

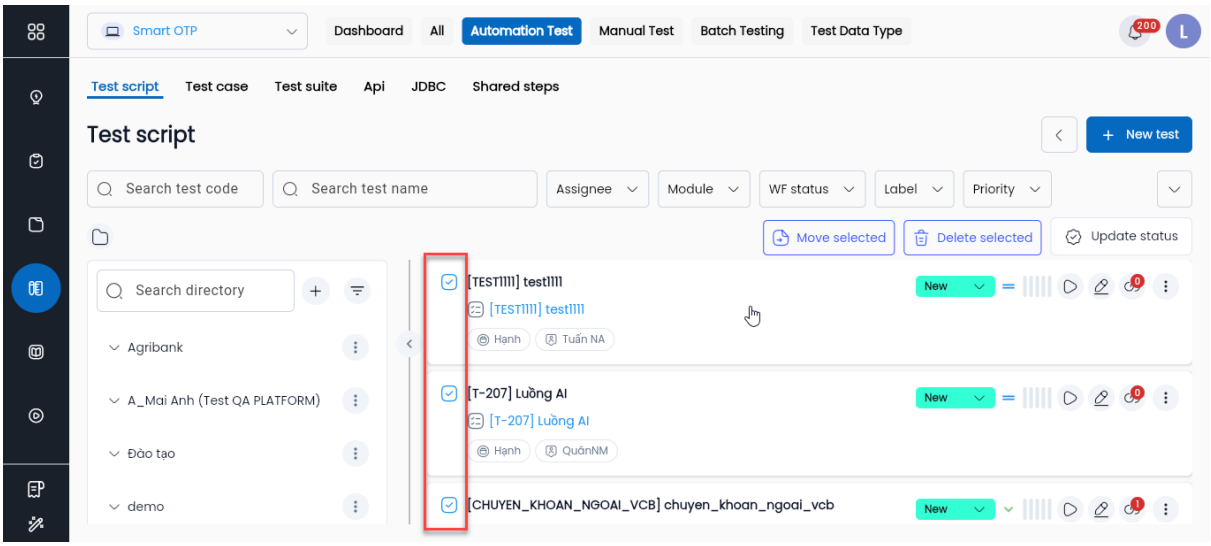
Chức năng này cho phép người dùng **xóa nhiều test script cùng lúc** thay vì phải thao tác từng cái, giúp tiết kiệm thời gian và quản lý dễ dàng hơn.

Chú ý:

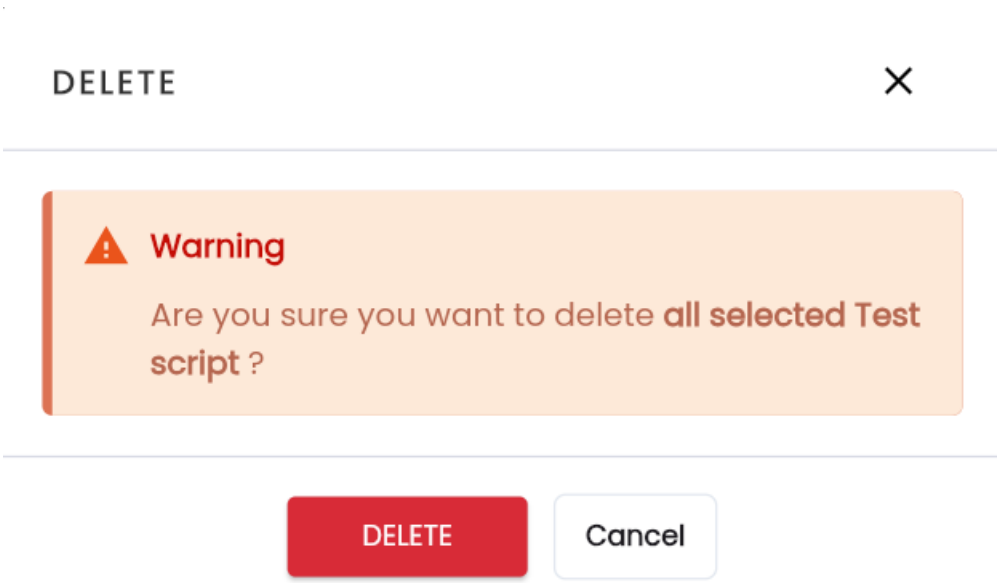
- Chỉ có thể xóa các **test script ở trạng thái Inactive**.
- Các test script phải **chưa từng được run**.
- Nếu test script đang ở trạng thái **Active** hoặc đã được run, hệ thống sẽ **không cho phép xóa**.

Các bước thực hiện:

Bước 1: Thực hiện chọn các test script bạn muốn xóa



Bước 2: Nhấn **Delete selected**, màn hình popup xác nhận xóa



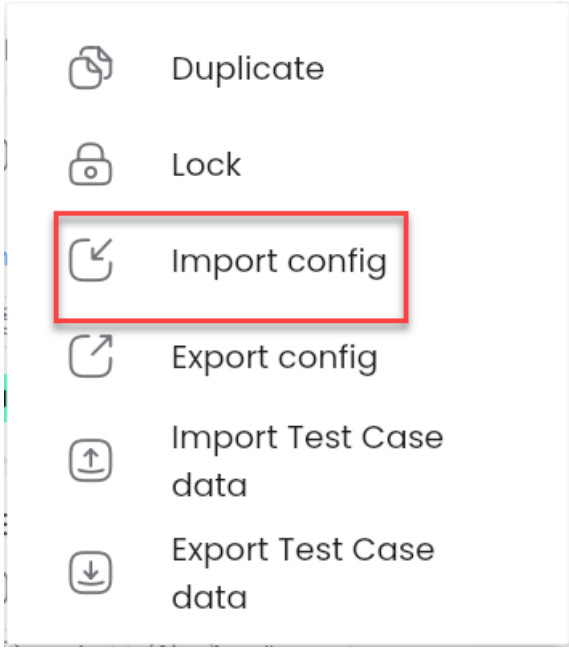
Bước 3: Nhấn "Delete", màn hình hiển thị thông báo "Delete success"

1.1.5.10 Import Json file

Json là một kiểu định dạng dữ liệu tuân theo một quy tắc nhất định mà hầu hết mọi ngôn ngữ lập trình hiện nay đều có thể đọc được, dùng để truyền tải dữ liệu.

Để thực hiện chỉnh sửa/sao chép cấu hình của kịch bản kiểm thử, ngoài duplicate, người dùng có thể sử dụng tính năng *import json file*.

Bước 1: Nhấn "⋮" ở phía bên phải bản ghi kịch bản kiểm thử muốn tải lên, chọn "**Import config**"



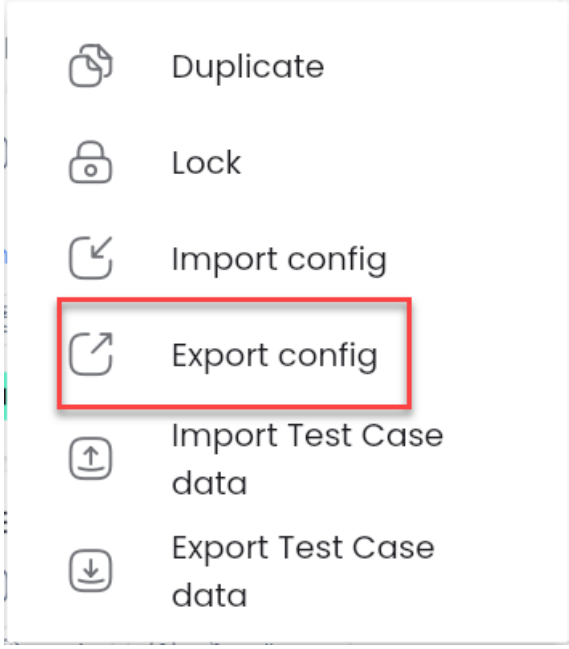
Bước 2: Thực hiện tải lên file json đã chuẩn bị.

Hiện nay: Người dùng có thể import file .json từ **template bất kỳ** (do chúng tôi cung cấp hoặc tự tạo). Hệ thống sẽ cho phép **mapping dữ liệu** từ template đó vào metadata của test case nội bộ.

Sau khi mapping, người dùng có thể **lưu lại thành một template đặc thù**, giúp tái sử dụng cho các lần import sau.

1.1.5.11 Export Json file

Nhấn "⋮" ở phía bên phải bản ghi kịch bản kiểm thử muốn tải về, chọn "**Export config**"



File json chứa dữ liệu kiểm thử được tải về và hiển thị trong thư mục trên máy người dùng

1.1.6 Shared Steps

Trong các kịch bản kiểm thử dạng quy trình/ flow của một dự án thường có các bước giống nhau xuất hiện. Ví dụ: Đăng nhập, Phê duyệt…

Để sử dụng lại một đoạn cấu hình đã có trong các test script/ suite khác, người dùng có thể sử dụng chức năng copy – drag hoặc select mode để sao chép một nhóm các keyword. Tuy nhiên, việc phải di chuyển qua lại giữa các test script/ test suite để thực hiện sao chép, kéo thả dễ dẫn đến nhầm lẫn, rất dễ phát sinh các lỗi không đáng có. Từ đó, QA Platform xây dựng một loại kịch bản kiểm thử khác là ***shared steps*** hỗ trợ người dùng tái sử dụng các đoạn thuật toán đã có.

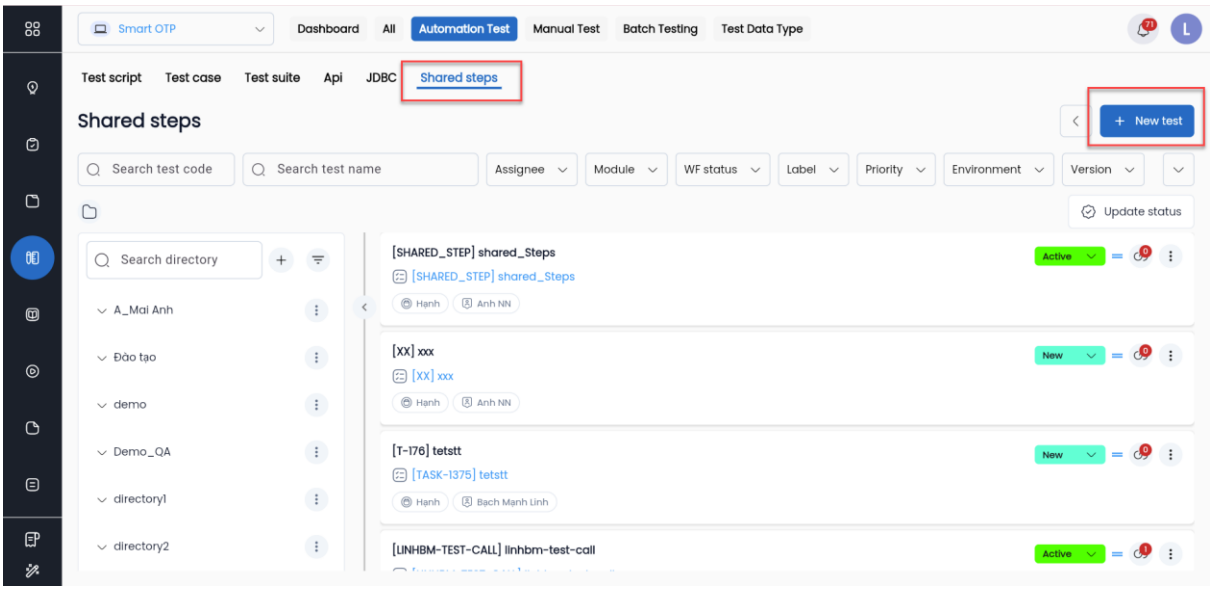
1.1.6.1 Shared Steps là gì?

Shared Steps là một tính năng hỗ trợ người dùng cấu hình kịch bản kiểm thử, cho phép người dùng tái sử dụng một bước/nhóm bước cấu hình có sẵn.

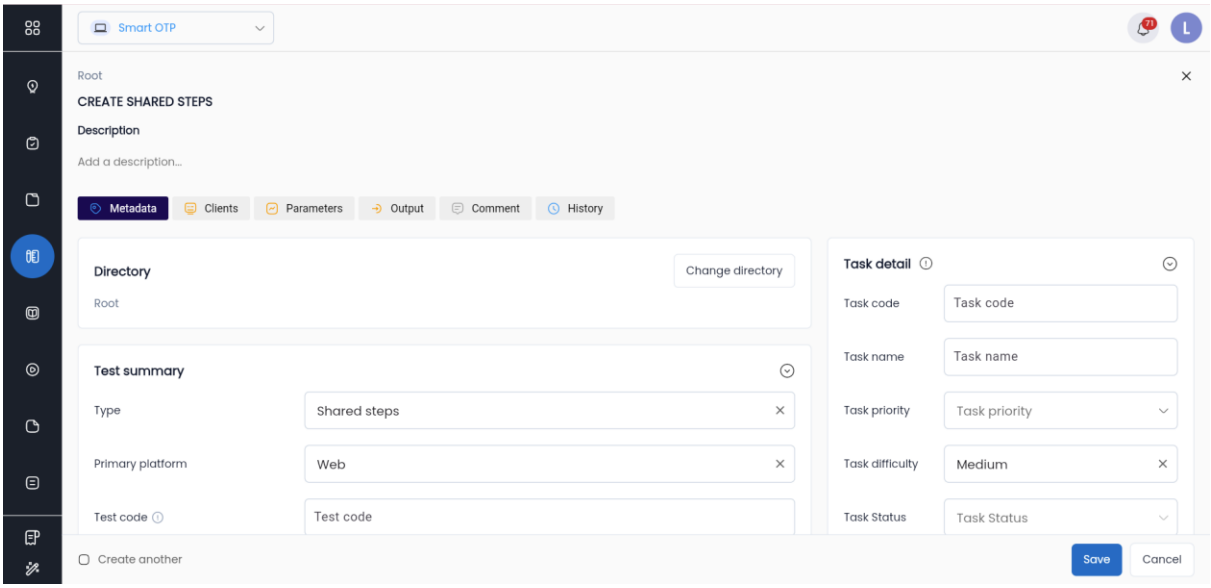
Là một kịch bản kiểm thử tương tự test script, có thể được sử dụng cho bất kịch bản kiểm thử nào thông qua keyword "*Action call*".

Sử dụng shared steps khi có một nhóm các hành động được xuất hiện trong nhiều kịch bản kiểm thử.

1.1.6.2 Tạo một shared steps



Bước 1: Nhấn **"New test",** hệ thống hiển thị màn hình Create Shared Steps



Bước 2: Nhập đầy đủ và hợp lệ các trường thông tin, trong đó:

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
General information				
Directory	-	Button	Nếu Directory =	Hiển thị danh sách các directory thuộc project ở tab

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
			Root thì test case thuộc directory gốc	Shared Steps cho phép người dùng chọn 1 directory cho test case
Type	Có	Combobox		Mặc định là Shared steps
Primary platform	Có	Combobox		Hiển thị danh sách các primary platform có trong hệ thống, cho phép người dùng chọn 1 giá trị.
Test Code	Có	Text	Code phải là duy nhất và không thể chỉnh sửa, bắt đầu bằng chữ cái và không chứa kí tự đặc biệt ngoài "_"	Cho phép người dùng nhập mã của test case, là thông tin để phân biệt các test với nhau
Test Name	Có	Text		Cho phép người dùng nhập tên của shared steps.
Test priority	Có	Combobox		Mức độ ưu tiên thực hiện của test + Highest: Cao nhất + High: Cao + Medium: Trung bình + Low: Thấp + Lowest: Thấp nhất
Environment	Có	Combobox	Danh sách environment ở trạng thái active	Hiển thị danh sách environment có trong hệ thống ở trạng thái active, cho phép người dùng chọn 01 giá trị

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Module	Có	Combobox	Danh sách Module ở trạng thái active	Hiển thị danh sách module có trong dự án ở trạng thái active, cho phé người dùng chọn 01 giá trị
Version	Có	Combobox	Danh sách version ở trạng thái active	Hiển thị danh sách các version có trong hệ thống, cho phép người dùng chọn 1 giá trị. Chi tiết về version
Label	Có	Combobox	Danh sách label ở trạng thái active	Hiển thị danh sách các label có trong hệ thống, cho phép người dùng chọn 1 giá trị.
Purpose	-	Text area		Cho phép người dùng nhập mục đích của shared steps
Precondition	-	Text area		Cho phép người dùng nhập điều kiện tiên quyết của shared steps
Test steps	-	Text area		Cho phép người dùng nhập các bước thực hiện trong shared steps
Expect result	-	Text area		Cho phép người dùng nhập kết quả mong muốn của shared steps
Test data	-	Text area		Cho phép người dùng nhập dữ liệu test của shared steps
Manage Label	-			Click vào chức năng quản lý label, hiển thị màn hình danh sách các label đã có trong hệ thống. Click label muốn gán cho shared steps

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Task Code	-	Text		Code của task tương ứng với test
Task Name	Có	Text		Là tên của task tương ứng với test được thực hiện
Task Priority	Có	Combobox		Là mức độ ưu tiên thực hiện task + Highest: Cao nhất + High: Cao + Medium: Trung bình + Low: Thấp + Lowest: Thấp nhất
Task Status	-	-		Status mặc định dựa theo cấu hình share steps mới tạo
Task difficulty	Có	Combobox		Mức độ khó trong yêu cầu thực hiện của test + Very Hard: Rất khó + Hard: Khó + Medium: Trung bình + Easy: Dễ + Very easy: Rất dễ
Assign to	-			Lựa chọn người thực hiện test Nếu không chọn, mặc định test được thực hiện bởi người tạo
Scheduled start date	-			Ngày bắt đầu của task tương ứng với test
Due Date	-			Ngày kết thúc của task tương ứng với test

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Parameter	-			Cho phép người dùng truyền tham số cho shared steps.
Clients	Có			Nhập nền tảng mà shared steps được xây dựng. Hệ thống mặc định tạo tự động clients cho nền tảng Web
Output	-			Là tham số mà người dùng muốn hiển thị ở kết quả chạy đầu ra của test shared steps
Comment			Cần phải tạo share steps trước khi comment	Trong quá trình review kịch bản kiểm thử, người dùng có thể để lại ý kiến để chỉnh sửa cấu hình được kịp thời.
History				Cho phép người dùng xem toàn bộ nhật ký chỉnh sửa và cập nhật của share steps, giúp quản lý và kiểm soát thay đổi trong suốt vòng đời kiểm thử.

Bước 1: Nhấn **"Save"** để tạo shared steps, hệ thống hiển thị màn hình cấu hình tương tự như các kịch bản kiểm thử khác.

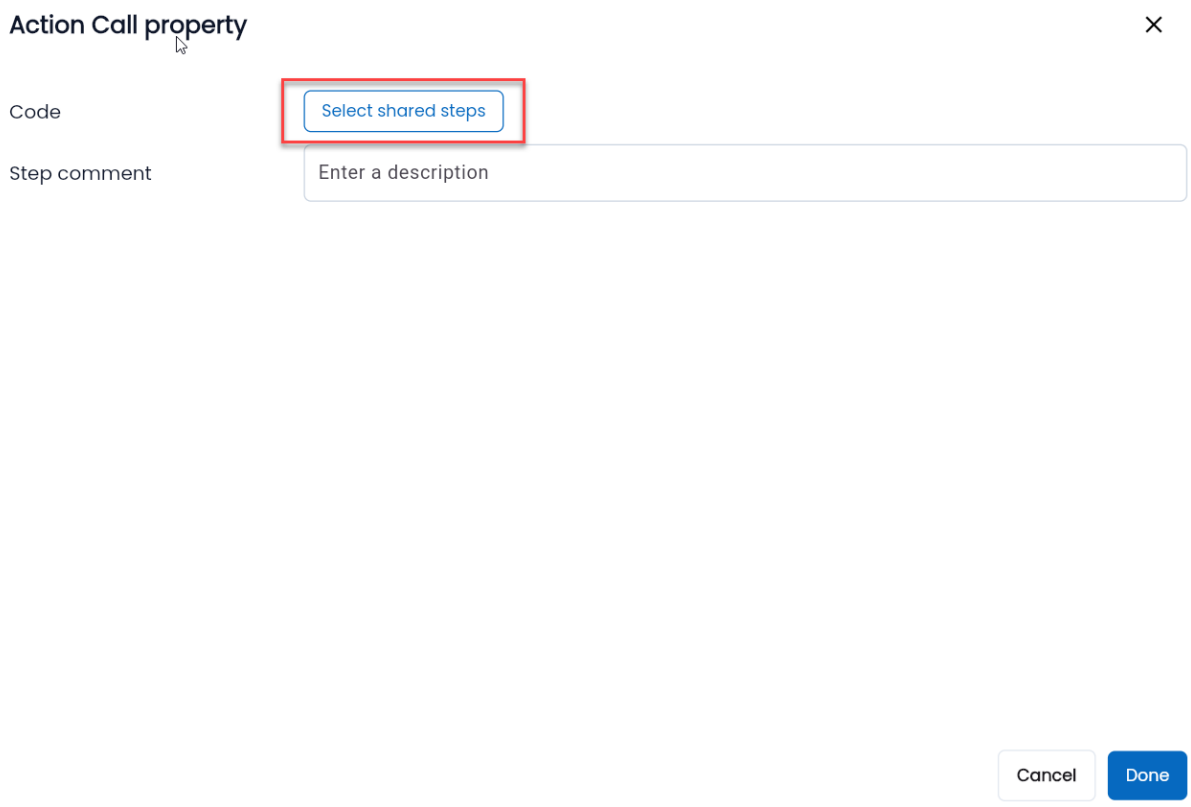
Lưu ý*: Shared steps** là một bước/nhóm các bước cấu hình, không phải là một luồng hoàn chỉnh nên không thể thực thi độc lập, do đó shared steps không có chức năng ***"Run". Ngoài ra shared steps có đầy đủ các chức năng tương tự các loại kịch bản kiểm thử khác.

1.1.6.3 Sử dụng shared steps

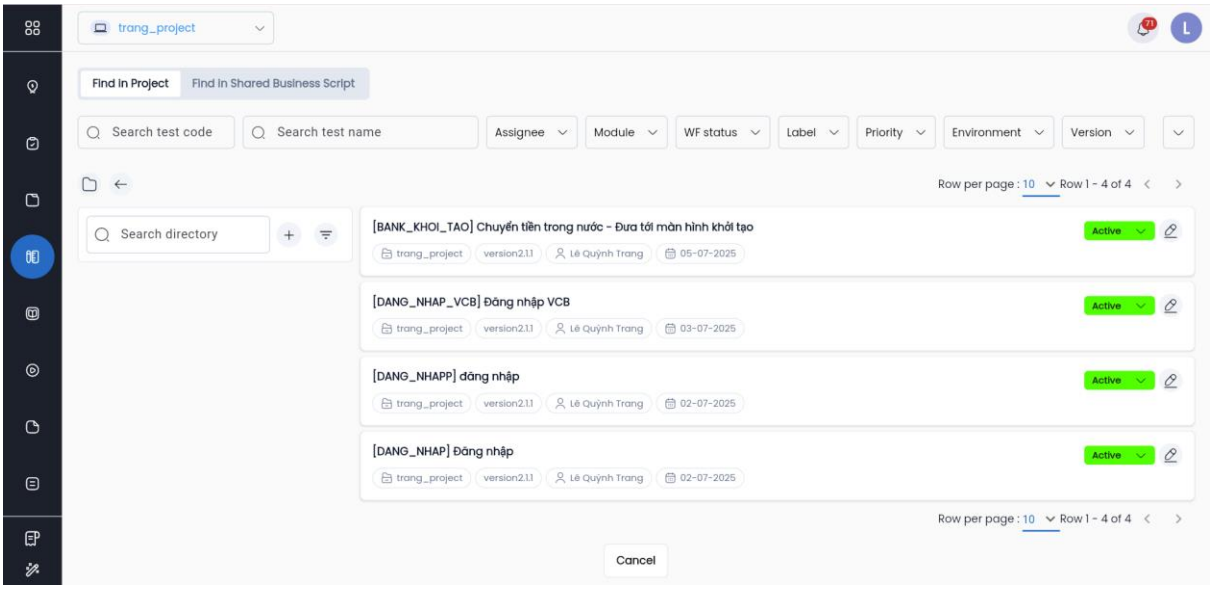
Để sử dụng một shared steps trong kịch bản kiểm thử khác, trạng thái của shared steps bắt buộc phải là ***active.***

Bước 1: Trên màn hình cấu hình kịch bản kiểm thử, thực hiện kéo thả keyword "Action call" vào vị trí cần sử dụng

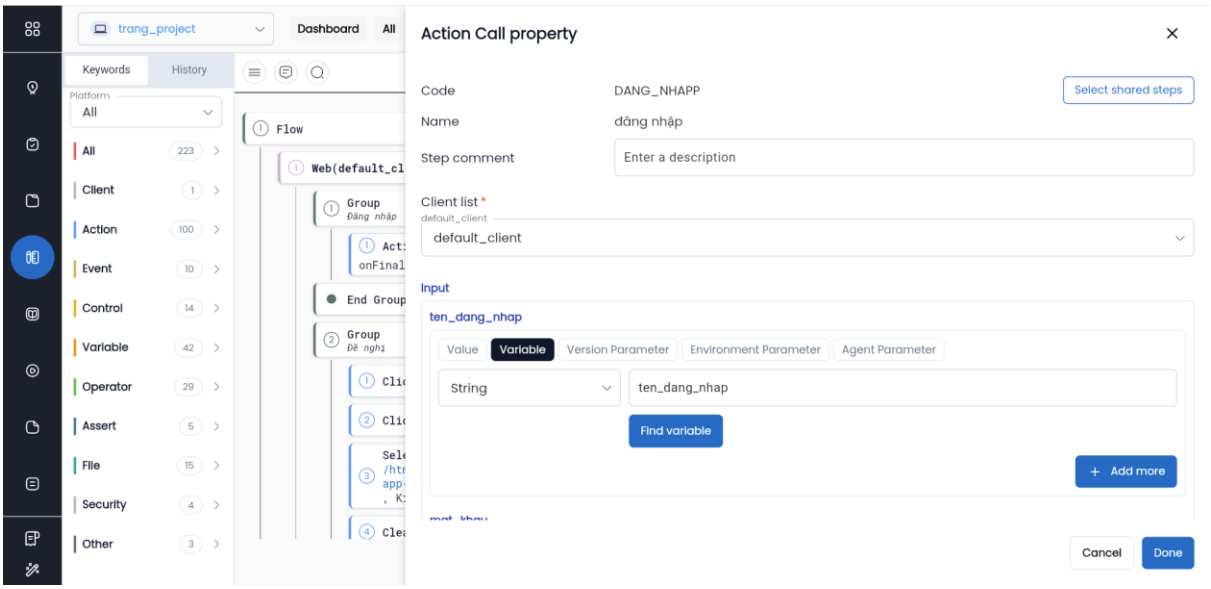
Bước 2: Nhấn vào Action call, hệ thống hiển thị màn hình *Action call property*,



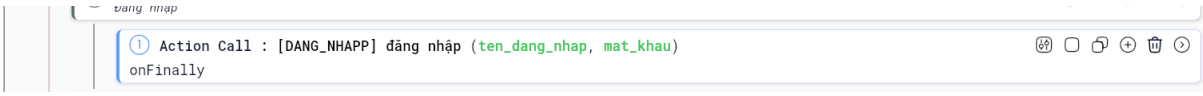
Bước 3: Nhấn "Select shared steps", hệ thống hiện thị danh sách các shared steps



Bước 4: Nhấn chọn bản ghi shared steps người dùng muốn sử dụng, hệ thống hiển thị màn hình thiết lập thông tin cho shared steps khi sử dụng trong kịch bản kiểm thử. Người dùng thực hiện nhập các thông tin cần thiết. Nhấn **Done** để hoàn tất.



Hệ thống hiển thị shared steps đã được sử dụng trong kịch bản kiểm thử.



1.1.6.4 Thay đổi cấu hình shared steps

1.1.6.4.1 Thay đổi cấu hình từ thư viện shared steps

Bước 1: Chọn tab Shared steps, chọn kịch bản shared steps người dùng muốn thay đổi cấu hình.

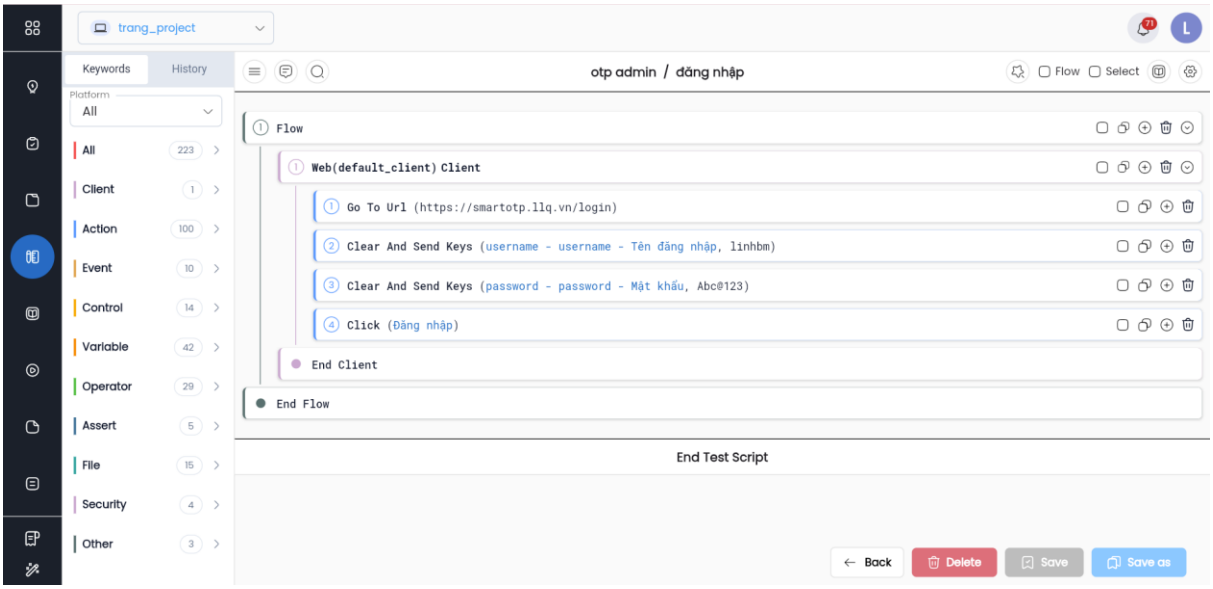
Bước 2: Thực hiện chỉnh sửa, cập nhật cấu hình shared steps tương tự như các kịch bản kiểm thử khác

Bước 3: Nhấn "**Save**".

1.1.6.4.2 Thay đổi cấu hình shared steps tại kịch bản kiểm thử

Bước 1: Trên màn hình cấu hình kịch bản kiểm thử, nhấn "View config" ở phía bên phải shared steps cần chỉnh sửa

Hệ thống hiển thị màn hình cấu hình shared steps



Bước 2: Thực hiện chỉnh sửa, cập nhật cấu hình của shared steps.

Bước 3: Nhấn "**Save**" để lưu cấu hình đã chỉnh sửa.

1.1.6.5 Thực hiện chức năng multi-select

Tham chiếu mục [Thực hiện chức năng multi-select của test script]

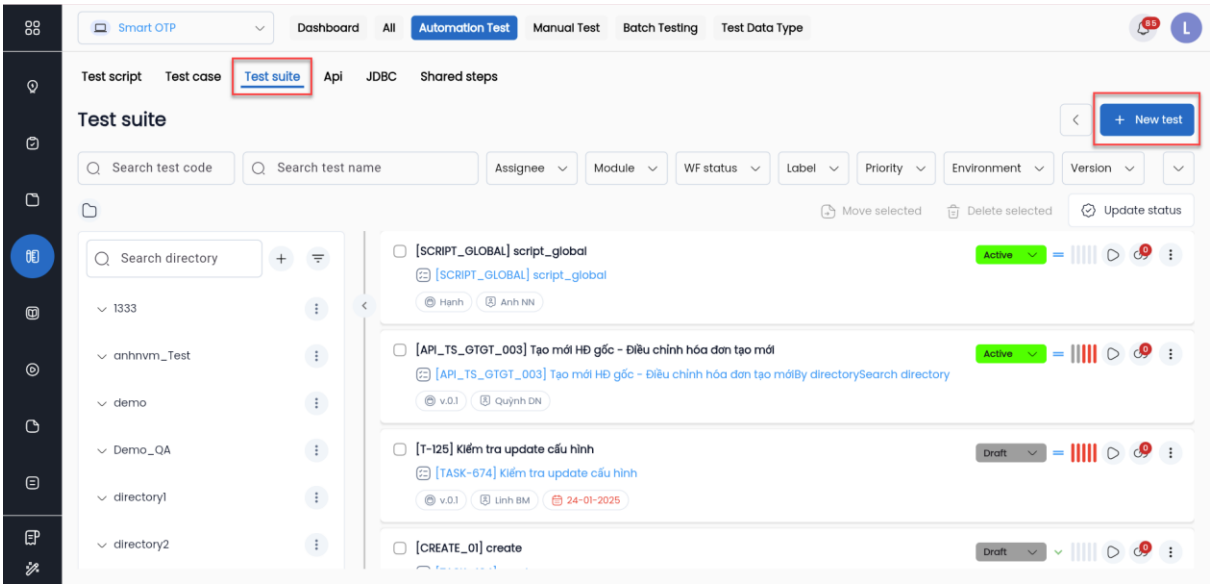
1.1.7 Test suite

1.1.7.1 Test suite là gì?

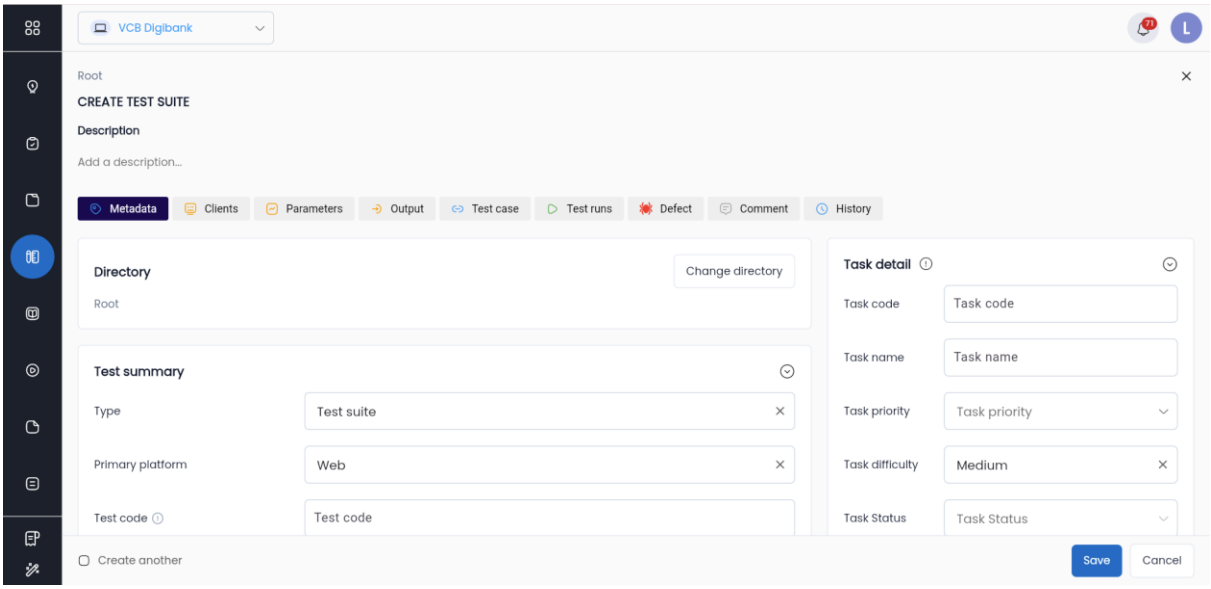
Test suite là kịch bản kiểm thử tương tự như test script, tuy nhiên có thể gọi đến nhiều test script đã có trong dự án và quản lý trình tự chạy của chúng để thực hiện một luồng lớn hơn hoặc test tích hợp các chức năng.

Test Suite tạo sự linh hoạt để quản lý thứ tự giữa các test script, nhóm test script theo bộ để có thể tạo các nhóm test case khác nhau. Điều này giúp người sử dụng dễ dàng chọn các test script nào sẽ được đưa vào sử dụng cũng như xác định thứ tự mà chúng sẽ chạy

1.1.7.2 Tạo mới một test suite



Bước 1: Trên màn hình của tab Suite, nhấn "**New test**", hệ thống hiển thị màn hình metadata



Bước 2: Nhập đầy đủ và hợp lệ cho các trường thông tin. [Tham chiếu Tạo mới Test script]

Bước 3: Nhấn "**Save**" để lưu.

1.1.7.3 Cấu hình test suite

Tham chiếu [Low-code test script]

1.1.7.4 Chỉnh sửa test suite

Tham chiếu tới mục [Edit test script]

1.1.7.5 Xóa test suite

Tham chiếu tới mục [Xóa test script]

1.1.7.6 Thực thi test suite

Tham chiếu tới mục [Run test script]

1.1.7.7 Nhân bản test suite

Tham chiếu tới mục [Nhân bản test script]

1.1.7.8 Import Json file

Tham chiếu tới mục [Import test script]

1.1.7.9 Thực hiện chức năng multi-select

Tham chiếu mục [Thực hiện chức năng multi-select của test script]

1.1.7.10 Export Json file

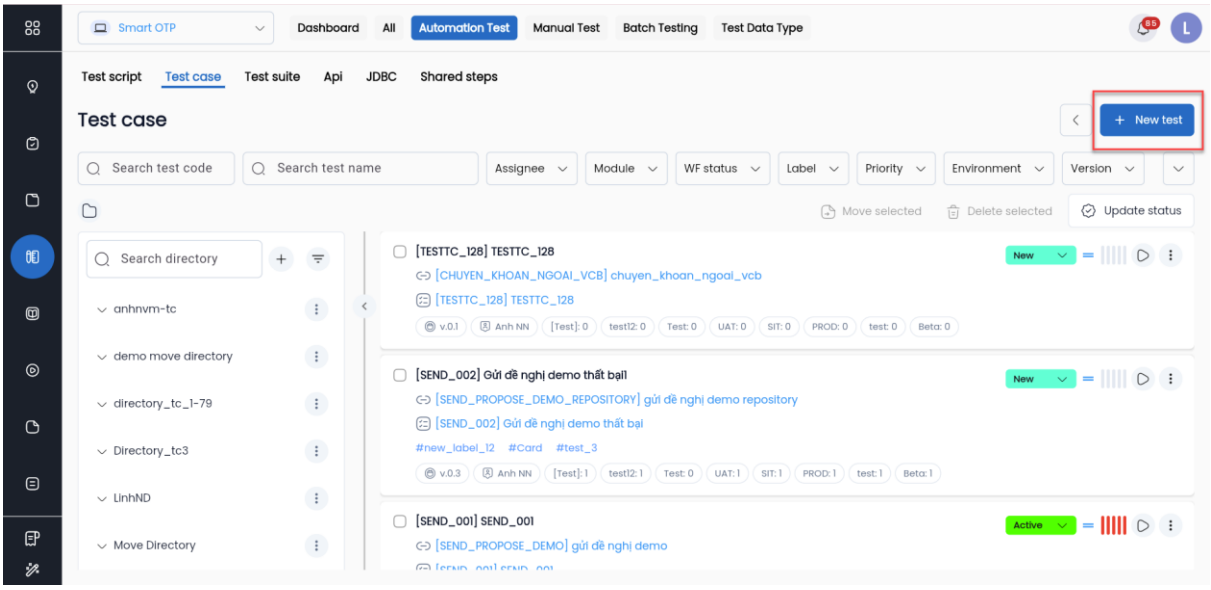
Tham chiếu tới mục [Export test script]

1.1.8 Test Case

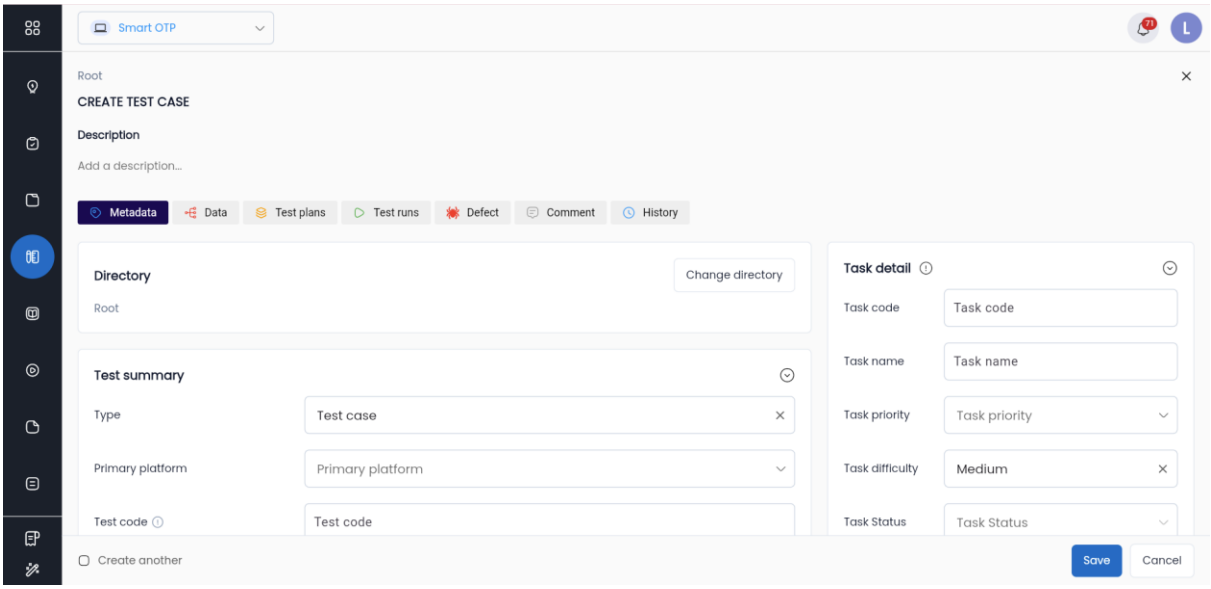
1.1.8.1 Test case là gì?

Test case là kịch bản kiểm thử bắt nguồn từ test script, trong đó thể hiện chi tiết về một trường hợp dữ liệu cụ thể của script, kết quả mong đợi, mức độ nghiêm trọng, mức độ ưu tiên của test case.

1.1.8.2 Tạo một test case

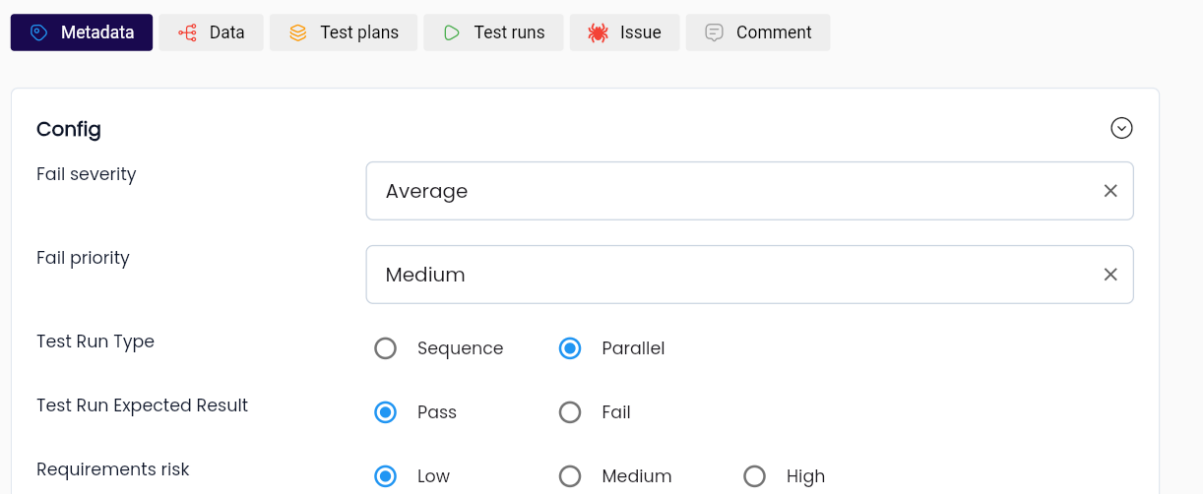


Bước 1: Trên màn hình thư viện test case, nhấn "**New test**", hệ thống hiển thị màn hình metadata



Bước 2: Nhập đầy đủ và hợp lệ các trường thông tin bắt buộc của test case, tương tự như test script

- Tuy nhiên có Test case yêu cầu một số thông tin khác:



Mô tả các trường thông tin

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
General information				

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Phần thông tin chung của 1 test case. Tương tự Test Script, tham chiếu tới mục Tạo mới một test script				
Section Config				
Test Run Type	Có	Radio button		Cho phép người dùng chọn 1 trong 2 giá trị: Sequence: Kiểm thử tuần tự Parallel: Kiểm thử đồng thời
Fail severity	-	Combobox		Hiển thị 05 mức độ nghiêm trọng, cho phép người dùng chọn 1 giá trị: Critical: Nghiêm trọng, ngăn cản người dùng thực hiện đúng chức năng đang kiểm thử Major: Cao, tuy nhiên không hoàn toàn ngăn chặn người dùng thực hiện chức năng đang test Average: Trung bình Minor: Thấp, không ảnh hưởng đến trải

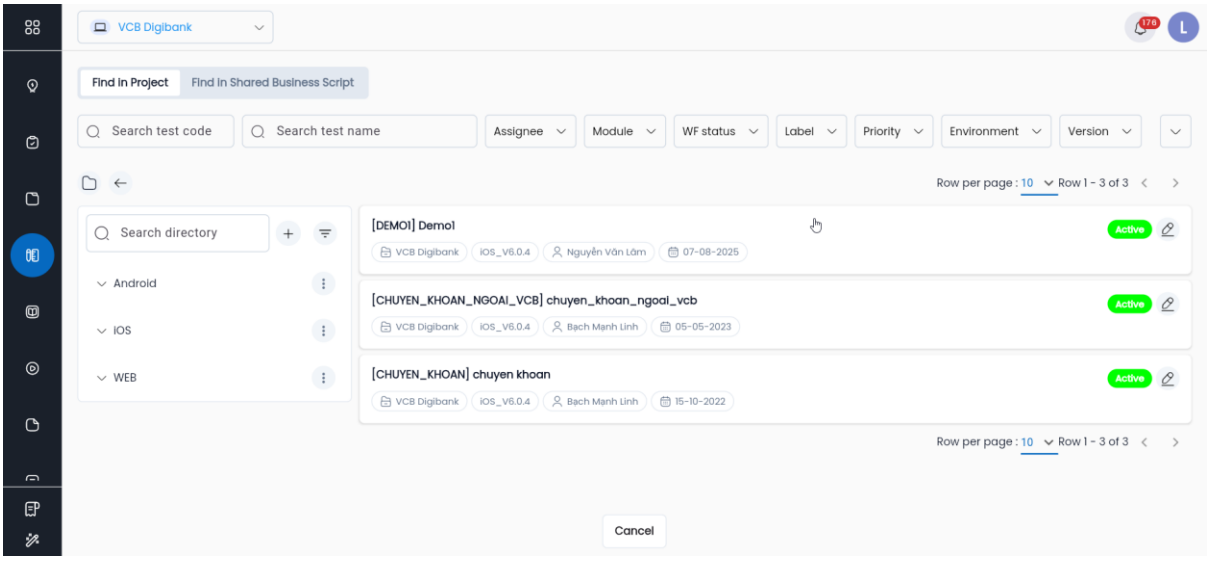
Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				nghiệm của người dùng Trivial: Không hẳn là lỗi và cũng không ảnh hưởng đến trải nghiệm của người dùng, tuy nhiên nên cân nhắc fix lỗi
Fail priority	Có	Combobox		Mức độ ưu tiên Dựa trên chức năng kiểm thử của test case mà quyết định mức độ ưu tiên: Highest High Medium Low Lowest
Test run expected result	Có	Combobox		Kết quả mong muốn của test case Pass Fail
Requirements risk	Có	Radio button		Khuyến nghị rủi ro Low: mức độ rủi ro thấp (ít khi xảy ra) Medium: mức độ rủi ro trung bình High: mức độ rủi ro cao

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Test run expected error message	-	Textarea		Thông báo lỗi mong muốn khi fail
Tab Data				
Select Test scenario	-	Button	Test case được xây dựng dựa trên test script	Chọn một test script để truyền tham số kiểm thử.
Select Test suite	-	Button	Test case được xây dựng dựa trên test suite	Chọn một test suite để truyền tham số kiểm thử

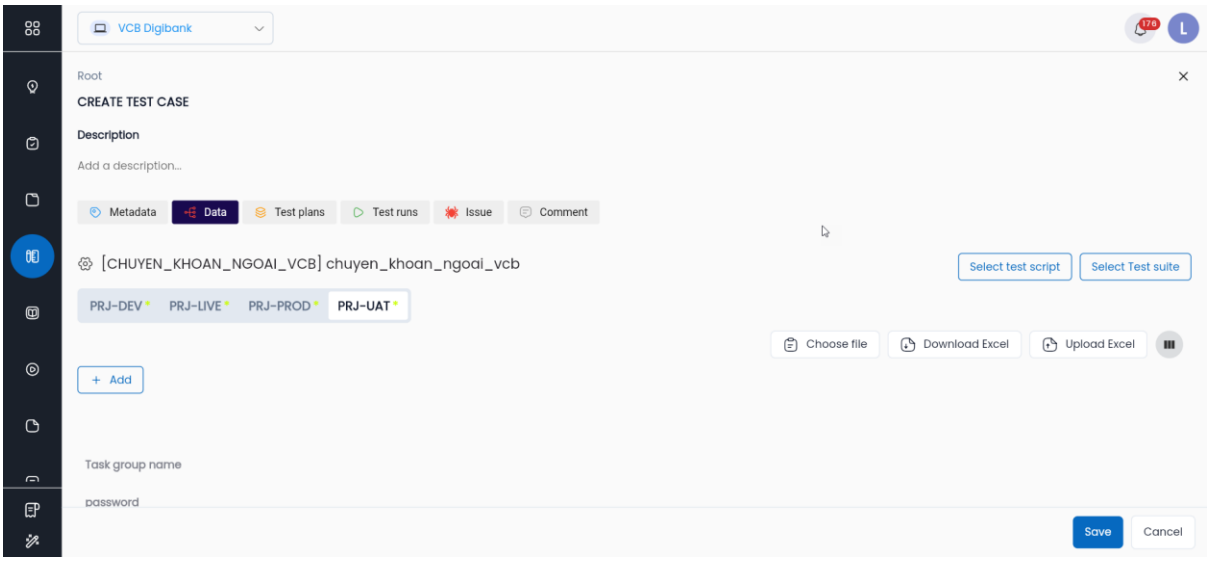
Bước 3: Tại tab Data, chọn test script hoặc test suite

Các bước chọn test script hoặc test suite để truyền tham số kiểm thử cho test case là tương tự nhau.

Khi nhấn “Select test script” tại tab Data, hệ thống hiển thị màn hình lựa chọn test script.



Bước 4: Người dùng chọn một test script trong danh sách cho test case, màn hình hệ thống hiển thị các parameter của script đã chọn



Trong đó:

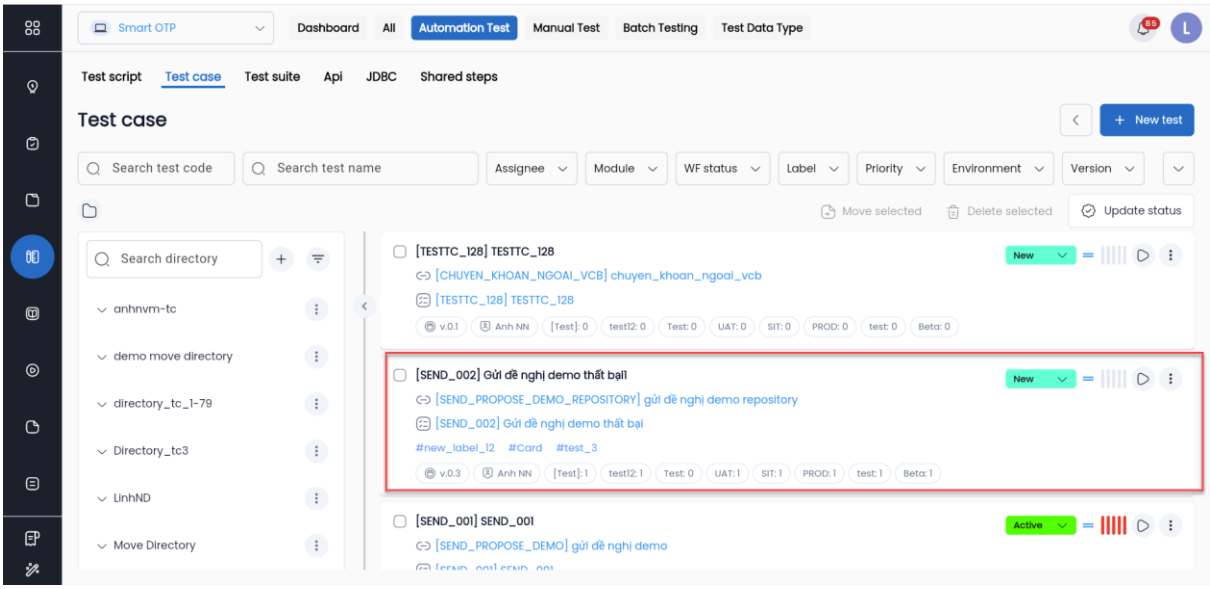
Các trường thông tin	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Test name	-	Label		Tên test script gọi đến

Parameters	-	Text		Nhập thông tin các Parameter đã khai báo tại test script hoặc test suite
	-	Icon	Xuất hiện khi có ít nhất 01 item	Click icon Delete để xoá dòng dữ liệu
+ Add	-	Icon		Click icon Add để thêm dòng dữ liệu thủ công cho test case Có thể click nhiều lần
Download Exel	-	Button		Click Download Excel để tải về form parameter của test script
Import Excel	-	Button		Click Import Excel và lựa chọn file để tải lên file excel chứa dữ liệu của test case (theo form vừa tải về khi click "Download Excel") một cách tự động
Choose File	-	Button		Click Choose File và lựa chọn file dữ liệu đã active trên project (trong chức năng File)
	-	Icon		Click icon này để layout các parameter của 01 item theo dòng hoặc cột

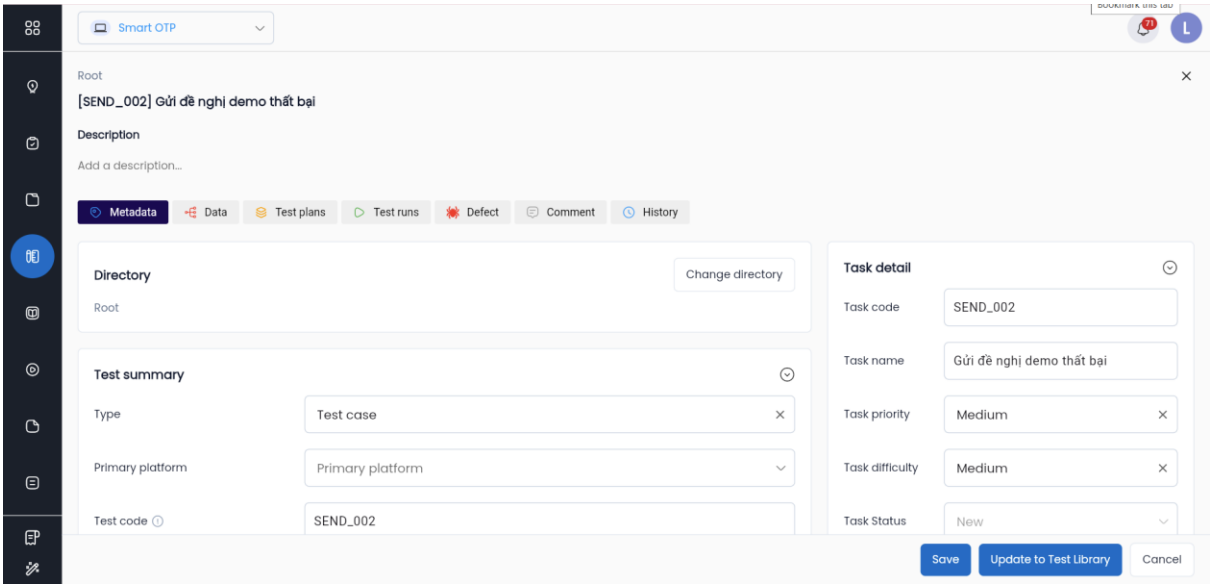
Bước 5: Nhấn "**Save**".

1.1.8.3 Edit test case

Bước 1: Tại màn hình thư viện test case, nhấn vào bản ghi test case muốn chỉnh sửa thông tin



Bước 2: Hệ thống điều hướng đến màn hình chỉnh sửa thông tin test case. Các trường thông tin giống với màn hình thêm mới. Trường **Code** và **Task Code** không được phép chỉnh sửa



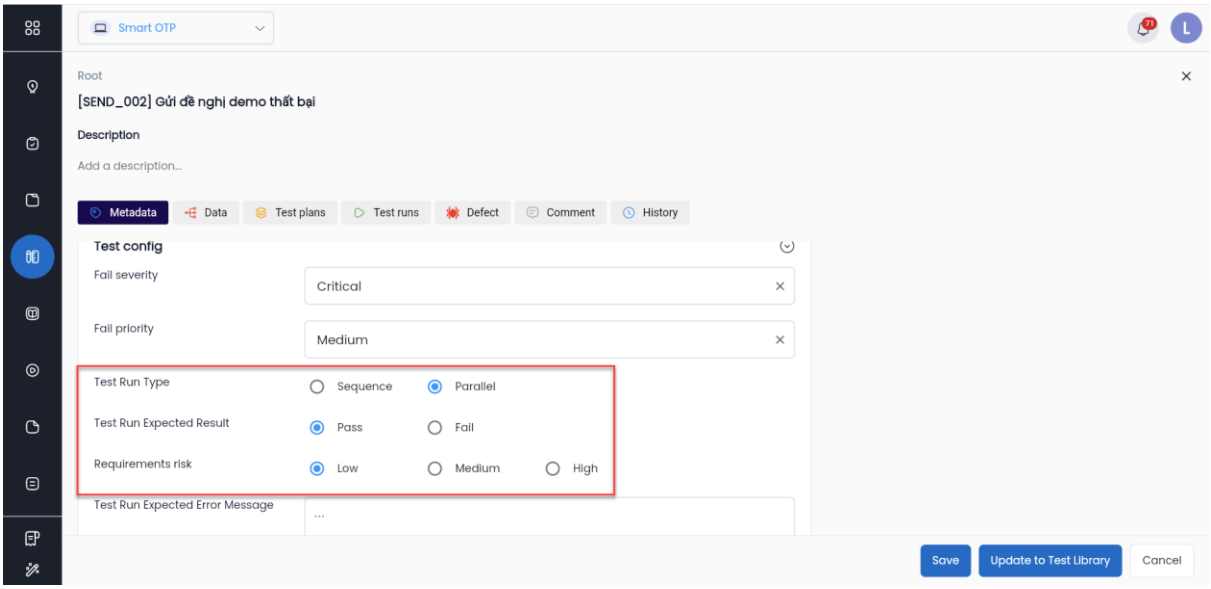
Bước 3: Nhập thông tin muốn chỉnh sửa của test case.

Bước 4: Nhấn **"Save".** Thông tin chỉnh sửa cho test case được cập nhật. Hệ thống hiển thị thông báo **"Updated successful".**

*1.1.8.4 Edit **Config** test case*

Bước 1: Trong Tab Metadata, chuyển tới phần Config

Để truy cập vào màn hình phần config của test case, người dùng có thể thực hiện click vào test case cần chỉnh sửa, hệ thống hiển thị màn hình Edit Test case, nhấn vào tab “Metadata”, cuộn màn hình xuống phần Config



Bước 2: Nhập thông tin cần chỉnh sửa tương tự như khi tạo mới test case

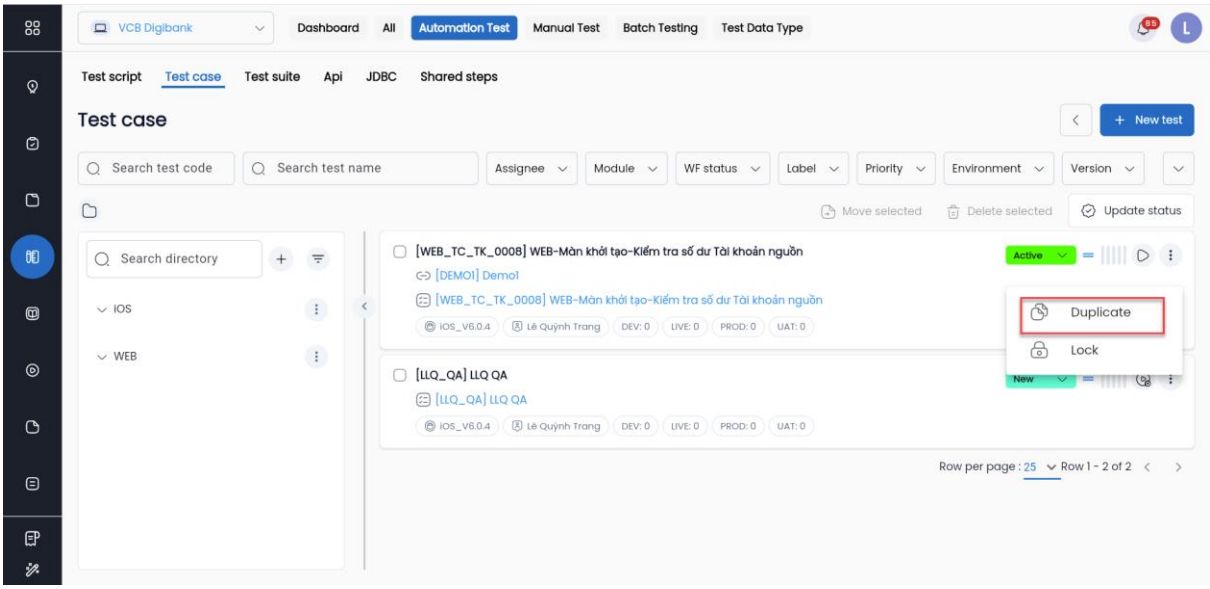
Bước 3: Nhấn “**Save**” để lưu thông tin chỉnh sửa

1.1.8.5 Xóa Test case

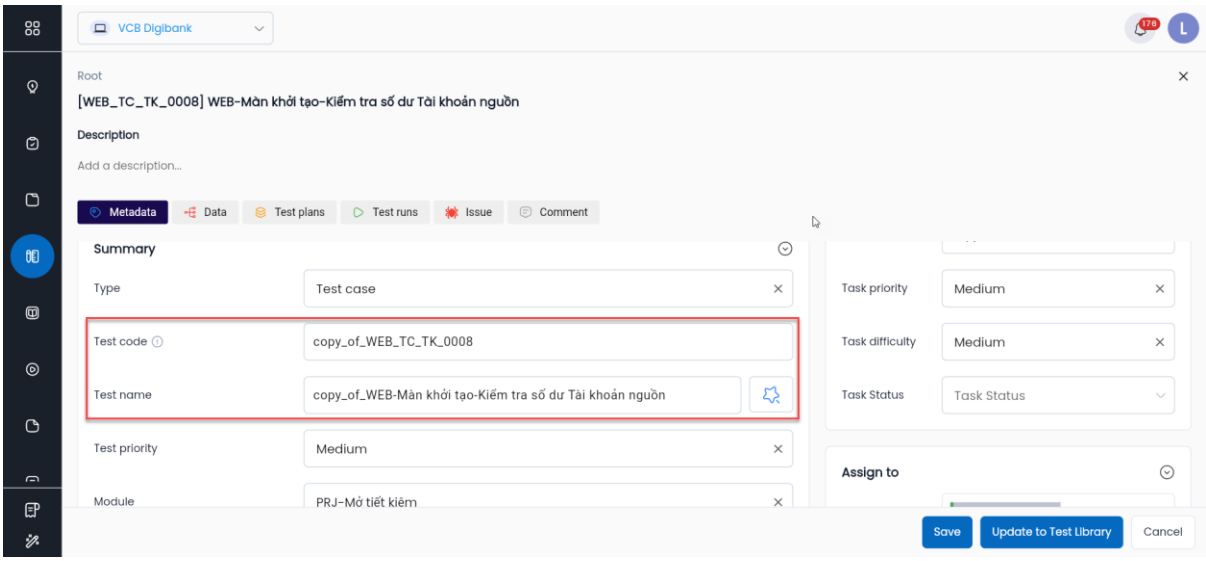
Tham chiếu đến mục [Xóa kịch bản kiểm thử]

1.1.8.6 Nhân bản test case

Bước 1: Nhấn button “ ” ở phía bên phải test case muốn nhân bản, chọn “**Duplicate**”



Bước 2: Các trường thông tin tại màn hình "Duplicate test case" đều được sao chép lại, tham chiếu [Tạo mới một test case](#).



Ngoài ra

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Tab General information				
Thông tin cơ bản				
Code	Có	Text	Code phải là duy nhất và không thể chỉnh sửa, độ dài tối thiểu 2 ký tự, bắt đầu bằng chữ cái và không chứa kí tự đặc biệt ngoài “_”,“-”	Code của test case duplicate sẽ mặc định được hệ thống chỉnh sửa thành “copy_of_” + code của test case được dùng để nhân bản. Người dùng có thể chỉnh sửa
Name	Có	Text		Name của test case duplicate sẽ mặc định được hệ thống chỉnh sửa thành “copy_of_” + name của test case được dùng để nhân bản. Người dùng có thể chỉnh sửa
Thông tin cấu hình metadata				
Project	Có	Combobox		Hiển thị danh sách các dự án trên hệ thống mà người dùng tham gia.

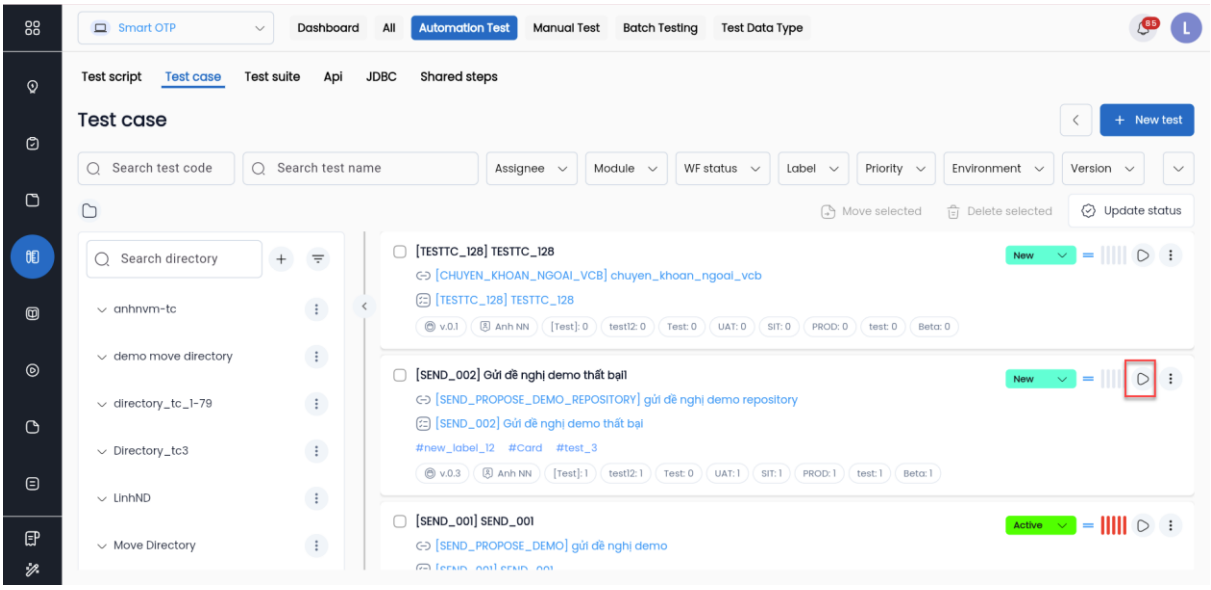
Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				Có thể duplicate test case từ các dự án khác nhau. Chỉ được chọn 01 giá trị
Workflow Status	Có	Combobox		Cho phép người dùng chọn 1 trong 4 giá trị: Draft/ Reject/ Active/ Review

Bước 3: Chọn test script/ test suite tương tự như khi tạo mới test case

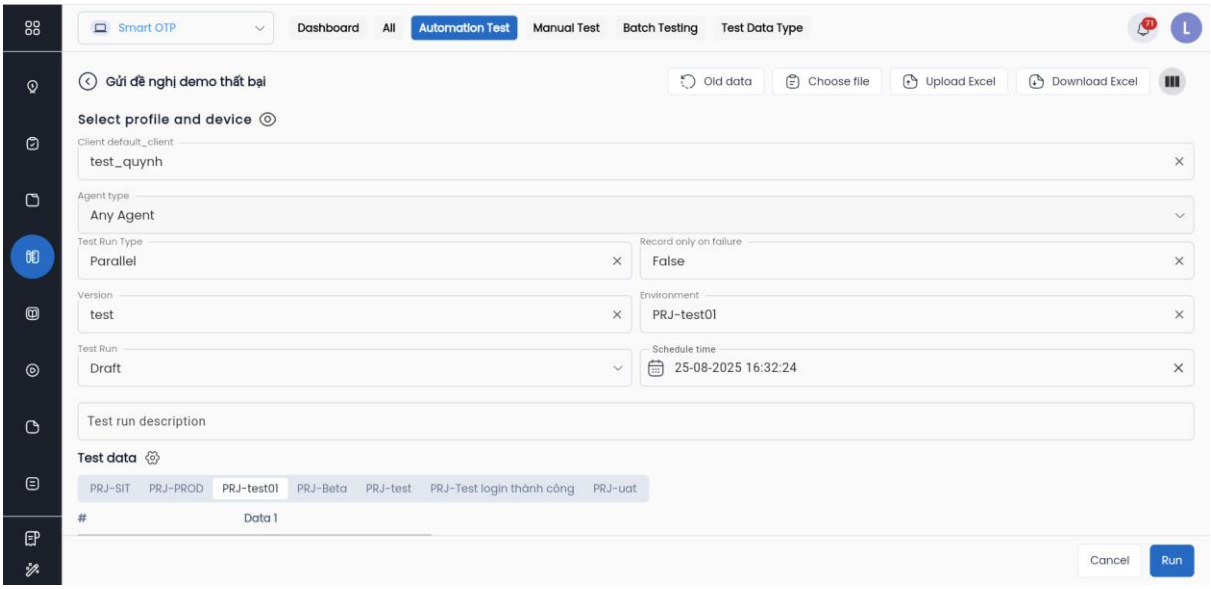
Bước 4: Nhấn "**Save**", hệ thống hiển thị màn hình thư viện test case của project đang thao tác và thông báo "Duplicate successful"

1.1.8.7 Thực thi test case

Bước 1: Trên màn hình thư viện test case, nhấn vào button " " ở phía bên phải test case muốn thực thi



Bước 2: Hệ thống hiển thị màn hình thực thi test case



Trong đó:

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Nút chức năng				
Old data	-	Button		Nút chức năng sẽ hiển thị lại danh sách lần Run trước để có thể dùng lại bộ dữ liệu từ lần Run trước.
Choose File	-	Button		Nút chức năng hiển thị màn hình của File Storage chứa các bộ dữ liệu đã được upload lên hệ thống.

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Upload Excel	-	Button		Nút chức năng giúp người dùng tải lên bộ dữ liệu đã được chuẩn bị bằng file Excel.
Download Excel	-	Button		Nút chức năng cho phép người dùng tải về file excel mẫu có chứa các trường thông tin cần dùng cho lần run test script.
	-	Button		Điều chỉnh kiểu hiển thị các parameters trong màn hình theo hàng ngang hoặc hàng dọc.
Select profile and device				Nút chức năng điều chỉnh ẩn/ hiện phần thông tin về nền tảng của lần Run test case
+ Add	-	Button		Nút chức năng giúp tạo thêm 1 dữ liệu test trong lần chạy

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
	-	Button		Nút chức năng giúp xóa dữ liệu test trong lần chạy
Thông tin trường thông tin				
Client default_client	Có	Combobox		Hiển thị danh sách các Profile đã được khai báo tại chức năng Projects Setting phù hợp với Client của test script. Chỉ được chọn 01 giá trị
Test Run Type	Có	Combobox		Cho phép người dùng chọn 1 trong 2 giá trị: Parallel: Kiểm thử đồng thời Sequence: Kiểm thử tuần tự
Agent Type	Có	Combobox		Chọn type Agent sẽ thực thi test: Public: Hệ thống sẽ lựa chọn các

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				Agent public đang active trên hệ thống có cùng thiết lập với profile đã chọn Private: Agent của user đang thực thi kịch bản kiểm thử AnyAgent: Hệ thống sẽ chọn bất kỳ Agent nào (bao gồm public và Agent private) có cùng profile đã chọn Người dùng được phép chọn 1.
Record only on failure		Combobox		Người dùng được phép lựa chọn trong 2 giá trị:

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				True Fasle (hệ thống mặc định là False)
Version	Có	Combobox	Hiển thị danh sách các version có trạng thái active của dự án	Lựa chọn phiên bản của hệ thống/ ứng dụng cần test mà kịch bản kiểm thử sẽ được thực thi. Cho phép lựa chọn 01 giá trị
Environment	Có	Combobox	Hiển thị danh sách các environment có trạng thái active của dự án	Là môi trường phát triển sản phẩm mà kịch bản kiểm thử sẽ được thực thi. Cho phép lựa chọn 01 giá trị
Test Run	Có	Combobox		Cho phép người dùng chọn 1 trong 2 giá trị cho thông tin Agent: Draft Complete

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				Sau khi test script Run hoàn tất, dữ liệu Run theo "Draft" có thể chuyển thành "Complete" tùy theo mục đích người dùng nhưng ngược lại thì không được phép.
Schedule time	Có	Date		Là thời gian sẽ thực thi kịch bản kiểm thử. Hệ thống mặc định là realtime, người dùng có thể lựa chọn một thời điểm khác trong tương lai để cài đặt lịch trình thực thi test.
Test run description	-	Text		Mô tả cho lần chạy test
Test data	-	Button		Button "Config of test data" cho phép người dùng di chuyển đến tab Data để thực

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				hiện các thao tác thêm, sửa, xóa tại tab Data và các tab Metadata, Config

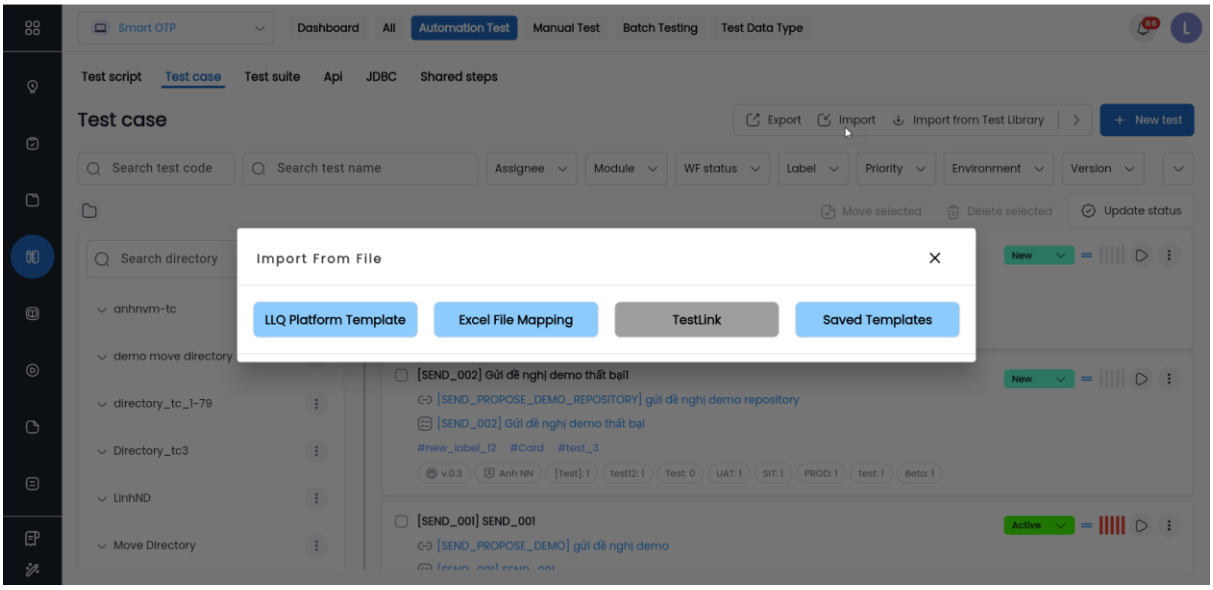
Bước 3: Nhấn "**Run**"

Sau khi Nhấn "Run" hệ thống sẽ chuyển hướng đến màn hình Run test case, đồng thời hệ thống tự ghi lại quá trình chạy các bước đã cấu hình trên hệ thống test.

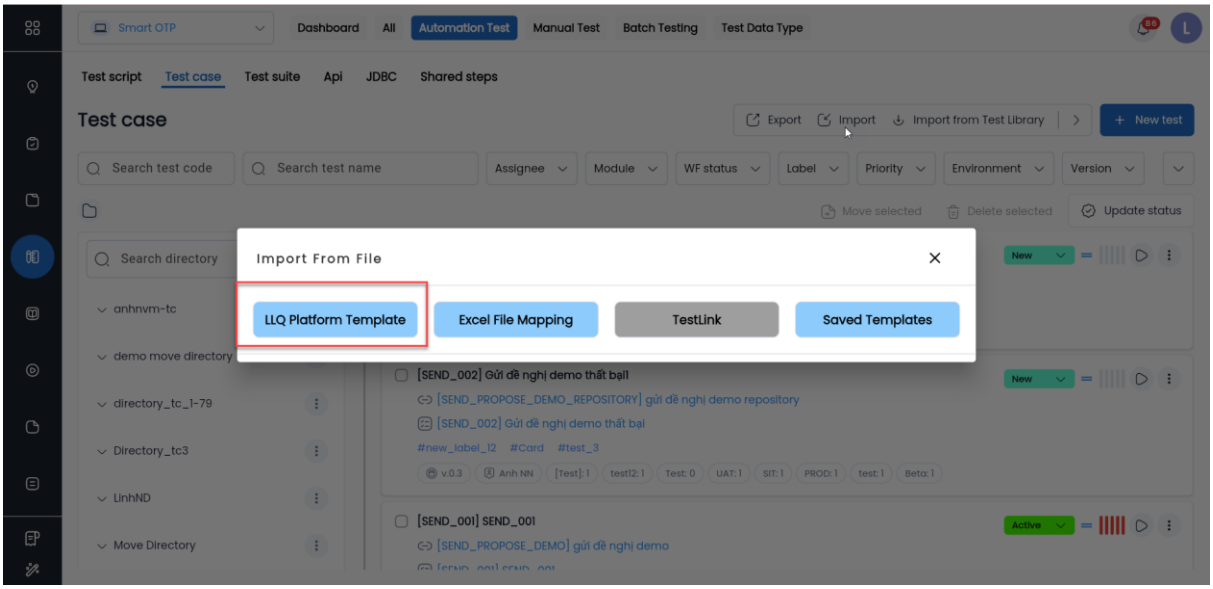
1.1.8.8 Thực hiện chức năng multi-select

Tham chiếu mục [Thực hiện chức năng multi-select của test script]

1.1.8.9 Import from file



❖ **LLQ Platform Template:**



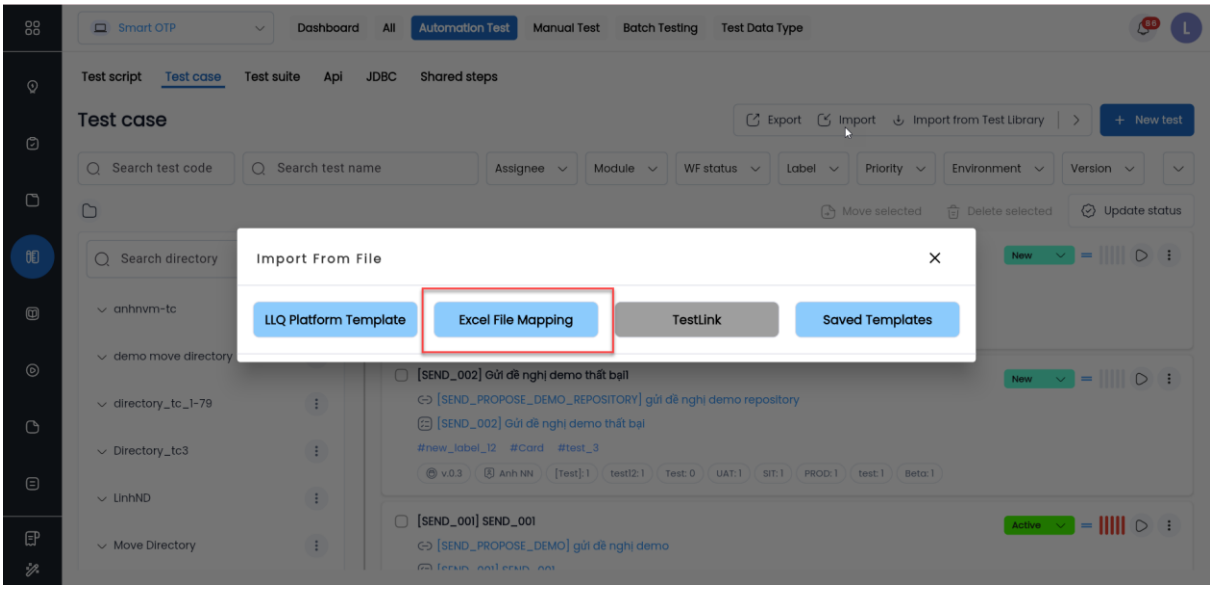
Với lựa chọn **LLQ Platform Template**, người dùng có thể **sử dụng trực tiếp file template chuẩn do hệ thống cung cấp** để import dữ liệu lên nền tảng.

- **Chuẩn hóa dữ liệu:** Template đã được định nghĩa sẵn đầy đủ các cột/trường cần thiết.
- **Nhanh chóng:** Không cần tự tạo file thủ công, chỉ cần điền thông tin vào template và import lên hệ thống.

Cách dùng

Bước 1: Tải **file LLQ Platform Template** chuẩn từ hệ thống.
Bước 2: Điền dữ liệu vào các cột/trường có sẵn.
Bước 3: Trở lại màn hình import, chọn **LLQ Platform Template**.
Bước 4: Upload file vừa điền → hệ thống sẽ nhận diện và import trực tiếp.

❖ Excel File Mapping:



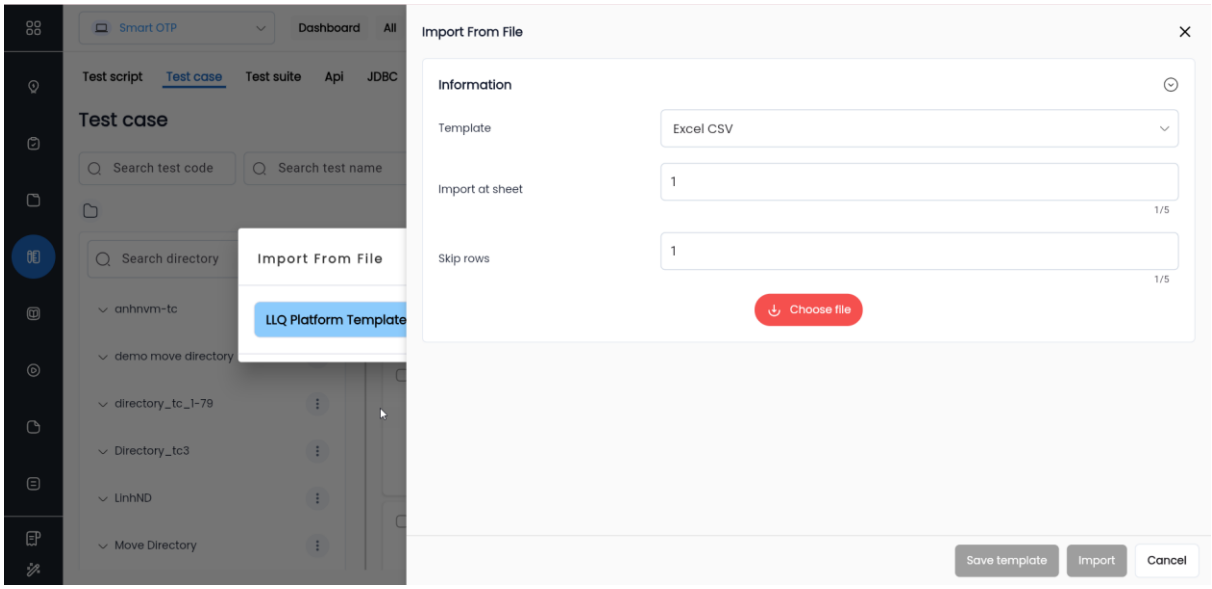
Với lựa chọn **Excel File Mapping**, người dùng có thể import file .json từ **template bất kỳ** (do chúng tôi cung cấp hoặc tự tạo). Hệ thống sẽ cho phép **mapping dữ liệu** từ template đó vào metadata của test case nội bộ.

- **Linh hoạt:** Cho phép dùng bất kỳ file Excel có sẵn, không cần theo template chuẩn.
- **Tùy chỉnh mapping:** Trong quá trình import, bạn sẽ cần **mapping các cột trong file Excel** với **các trường dữ liệu trên giao diện hệ thống**.
- **Kiểm soát:** Người dùng toàn quyền xác định cột nào trong Excel tương ứng với trường nào trên hệ thống

Cách dùng

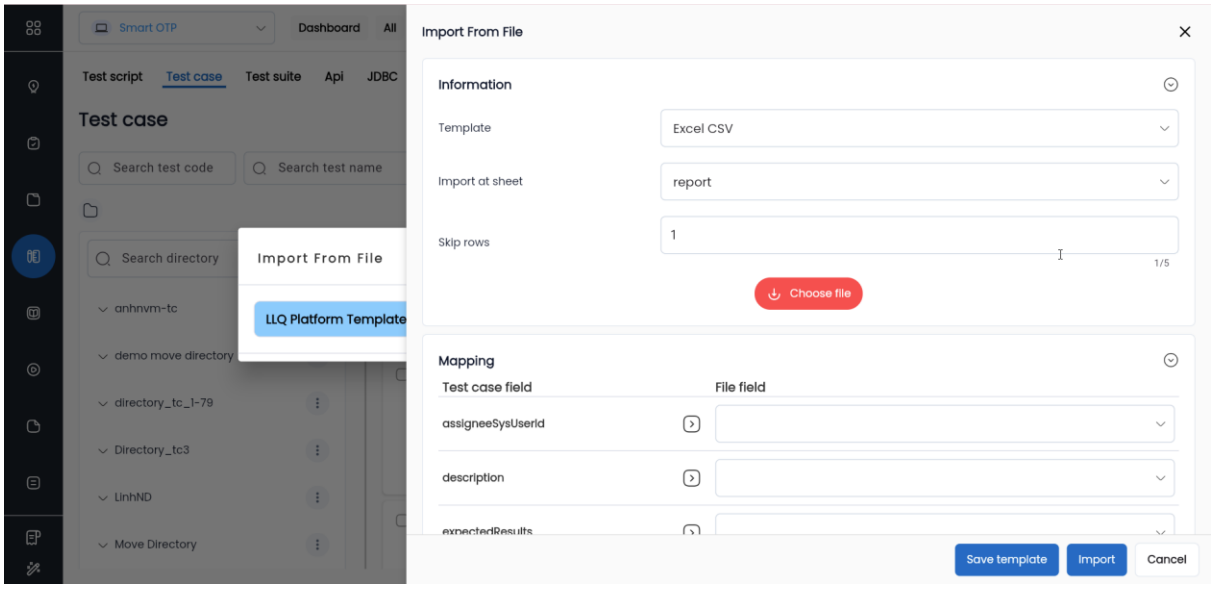
Bước 1: Chuẩn bị file Excel của bạn.
Bước 2: Vào màn hình import → chọn **Excel File Mapping**.

Hệ thống hiển thị màn hình Import from file



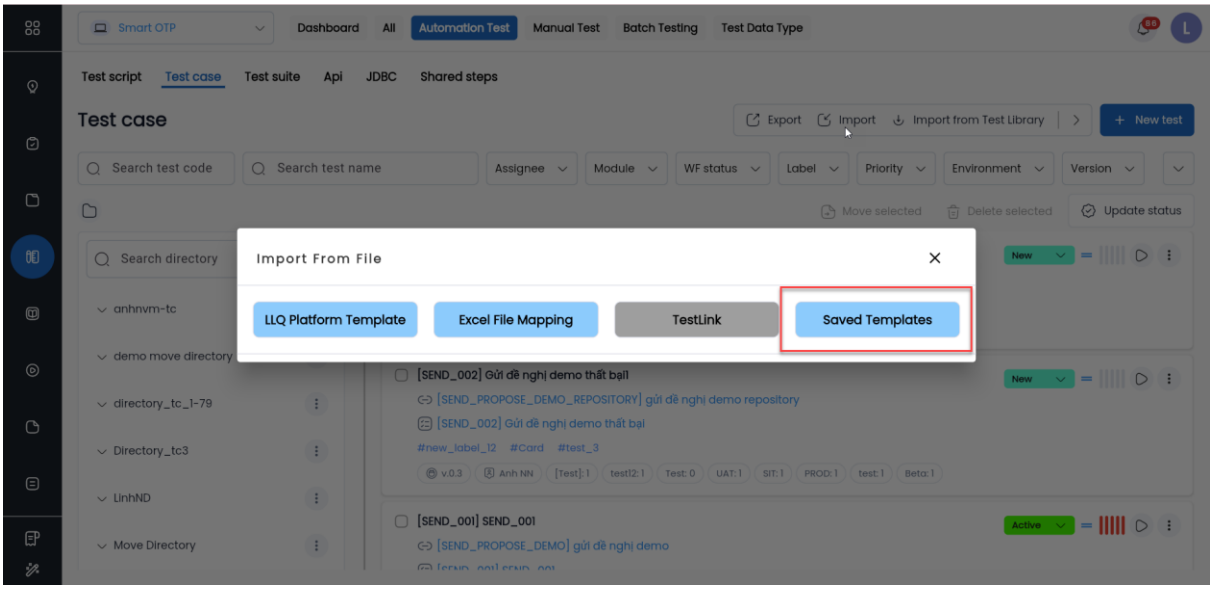
Bước 3: Upload file Excel.

Bước 4: Thực hiện **mapping**: Chọn trường tương ứng trên giao diện cho từng cột trong file.

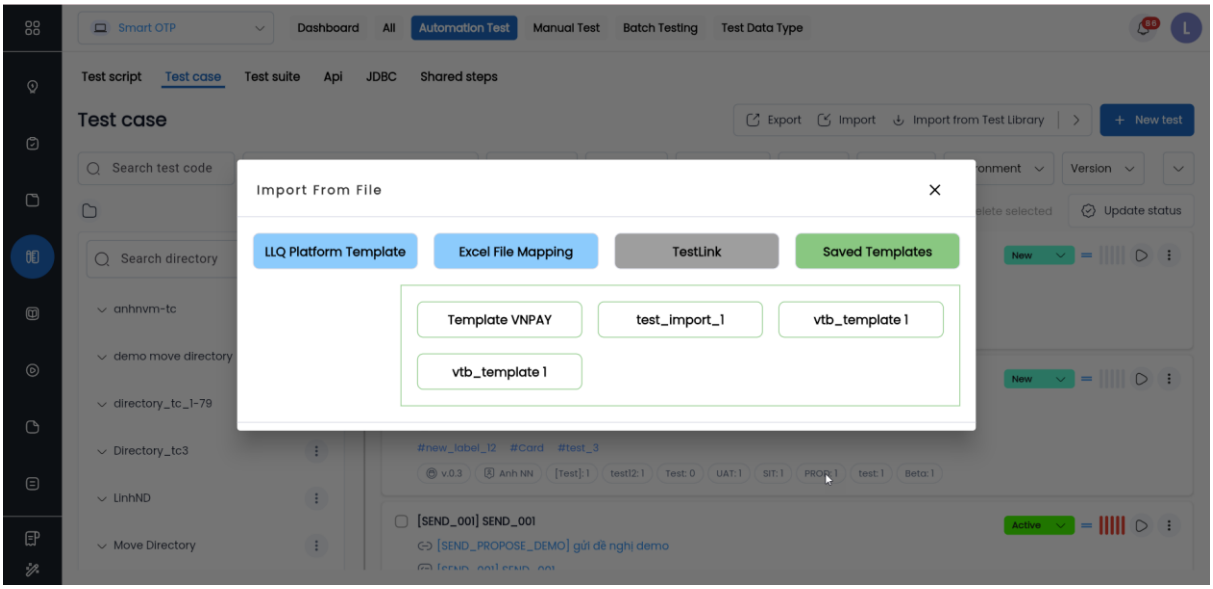


Bước 5: Xác nhận và tiến hành import. Trong trường hợp người dùng muốn sử dụng lại template, Click **"Save template".**

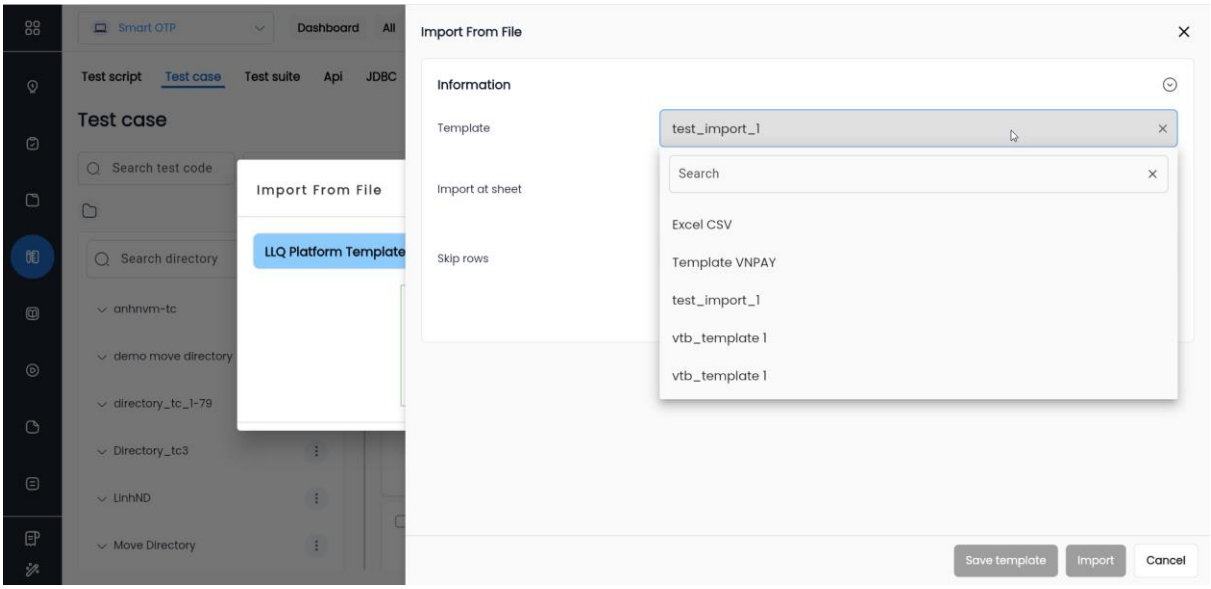
❖ Save Templates



- **Hiển thị danh sách các templates đã từng import trước đó** → người dùng có thể chọn lại để tái sử dụng, không cần cấu hình/mapping lại từ đầu.



- **Cho phép tạo mới template ngay tại đây** → trường hợp người dùng muốn import một file Excel khác, hoặc muốn định nghĩa lại cấu trúc/mapping mới.



1.1.9 Batch Testing

Lưu ý: Nếu hệ thống không hiển thị Batch Testing vui lòng chuyển xuống mục 5.6.2 Test Plan

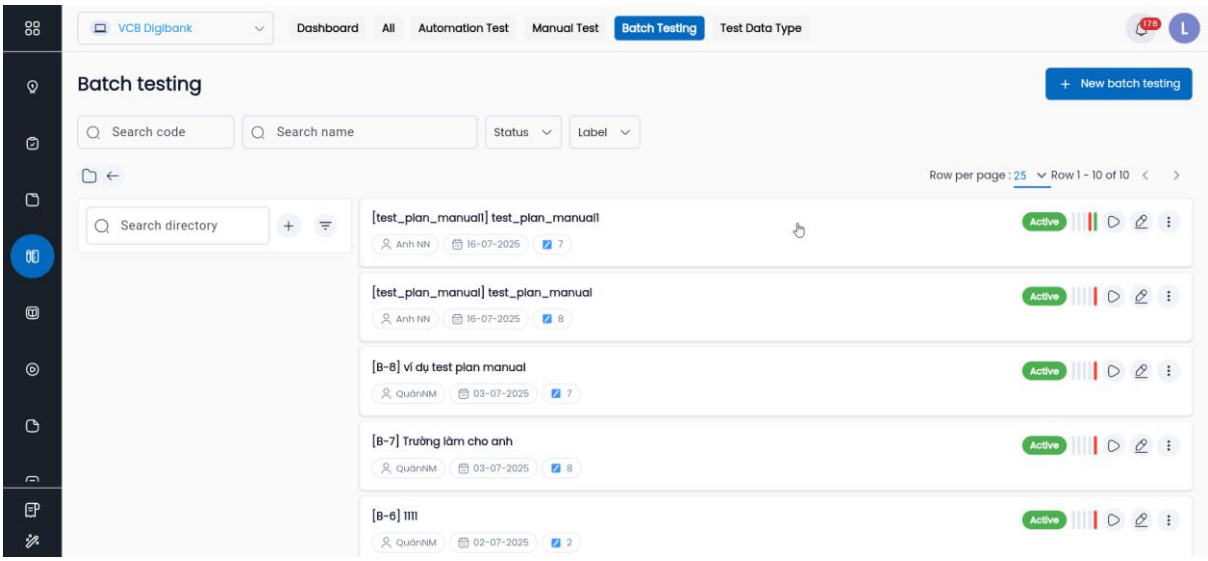
Trong quá trình phát triển sản phẩm sẽ có rất nhiều version được cập nhật để đáp ứng nhu cầu của người dùng cũng như cải tiến để thực hiện bài toán của thị trường. Khi đó việc thực hiện kiểm thử hồi quy là cần thiết để đảm bảo các chức năng cũ vẫn hoạt động đúng ở phiên bản hiện tại.

Tuy nhiên một hệ thống lớn sẽ có rất nhiều chức năng nhỏ, tương ứng với nhiều kịch bản kiểm thử. Việc thực hiện riêng lẻ từng kịch bản kiểm thử sẽ mất rất nhiều thời gian cho việc rà soát test case cần thực thi. Để thực hiện kiểm thử hồi quy nhanh chóng, công cụ kiểm thử phải đáp ứng được bài toán kiểm thử hàng loạt.

1.1.9.1 Batch Testing là gì?

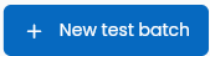
Batch Testing là chức năng thực hiện kiểm thử hàng loạt, phục vụ kiểm thử hồi quy, smoke test, sanity testing.

1.1.9.2 Thư viện Batch Testing



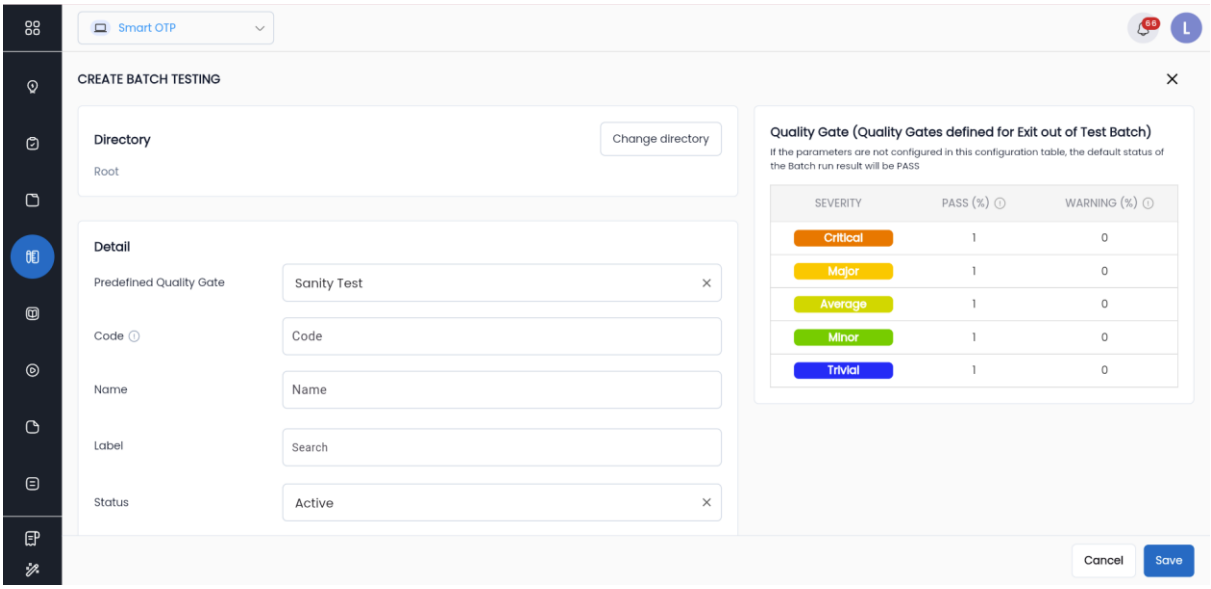
Mô tả trường thông tin

Trường thông tin	**Bắt buộc**	**Định dạng**	**Ràng buộc**	**Mô tả**
Search name	-	Text		Nhập tên test batch để thực hiện tìm kiếm theo tên
Search code	-	Combobox		Chọn trạng thái của batch để tìm kiếm: - Active - Inactive
Label	-	Combobox		Chọn label của batch để tìm kiếm
Name	-	-	-	Tên của test batch
Batch Steps	-	-	-	Số lượng Batch Steps trong Test Batch
Last run	-	-	-	Thống kê số lượng và trạng thái các batch steps trong lần chạy cuối cùng, trong đó dựa vào màu sắc để thống kê được kết quả: Green: Run pass

				Red: Run fail Orange: Running Blue: New Yellow: Warning
Owner	-	-	-	Người tạo test batch
Create date	-	-	-	Ngày tạo
	-	Icon		Click vào icon , hệ thống xổ ra list các chức năng mà người dùng thao tác với batch: - Move to directory: cho phép người dùng di chuyển từ thư viện/ directory này sang directory khác. - Delete: Xoá batch
	-	Button		Click vào button “New test batch”, hệ thống hiển thị màn hình create test batch

1.1.9.3 Tạo mới Test Batch

Bước 1: Nhấn **New Batch testing**, hệ thống hiển thị màn hình “CREATE BATCH TESTING”



Bước 2: Nhập đầy đủ và hợp lệ các trường thông tin bắt buộc, trong đó:

Trường thông tin	**Bắt buộc**	**Định dạng**	**Ràng buộc**	**Mô tả**
Predefined Quality Gate	Có	Combobox	Danh sách lựa chọn là Predefined quality gate đã được cấu hình trên hệ thống và có trạng thái active	□ **Mục đích:** Chuẩn hóa việc đánh giá chất lượng test, tránh phải cấu hình lại nhiều lần. □ **Cách dùng:** Chọn 1 mẫu có sẵn (Sanity Test, Regression Test, Smoke Test, UAT Smoke Test, PPE Smoke Test, PPE Regression Test). □ **Kết quả:** Batch test sẽ tự động kiểm soát và báo trạng thái dựa trên ngưỡng đã chọn.
Code	Có	Text	Nếu người dùng nhập	Là mã để định danh test batch trên dự án kiểm thử

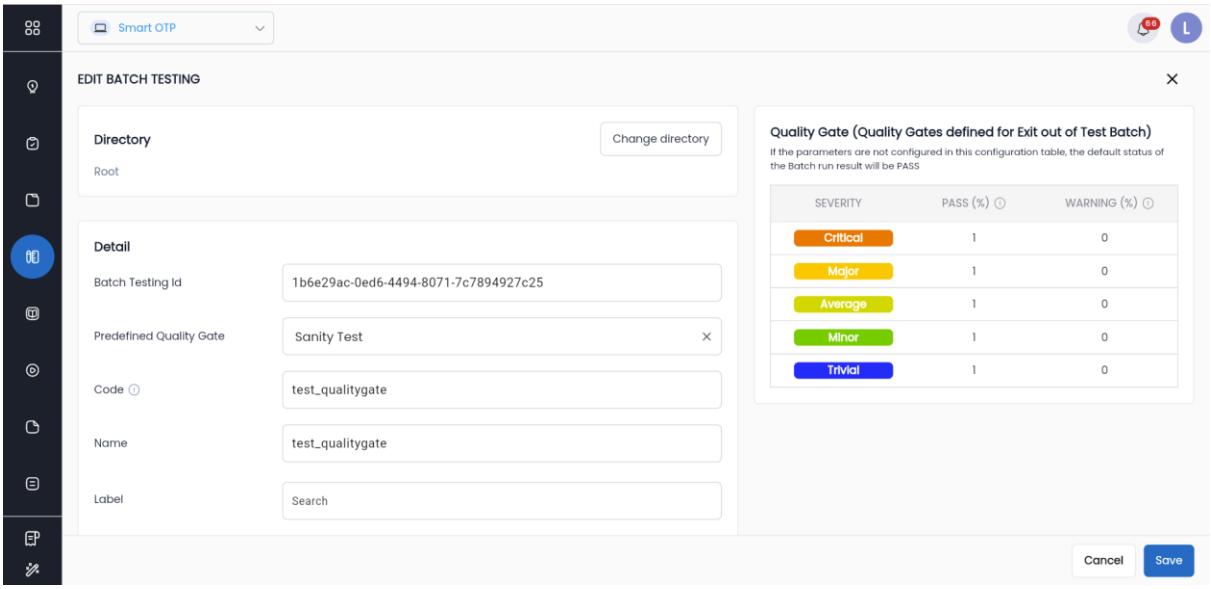
			code thì code phải là duy nhất, bắt đầu bằng chữ cái và không chứa kí tự đặc biệt ngoài "_","-"	Nếu không nhập, hệ thống tự sinh mã theo cấu trúc B-stt, trong đó: - B là ký hiệu test batch - stt được sinh tăng dần từ 1 và không trùng lặp
Name	Có	Text		Nhập tên của batch testing. Người dùng nên đặt tên test batch theo module, chức năng… để dễ dàng quản lý, theo dõi hơn
Status	Có	Combobox		Chọn trạng thái của batch: + Active + Inactive.
Description	-	Textarea		Nhập mô tả của test batch
Save		Button		Người dùng click "Save" để lưu batch
Cancel		Button		Người dùng click "Cancel" để huỷ

Bước 3: Nhấn "**Save**", hệ thống thông báo "**Create successful**"

1.1.9.4 Chỉnh sửa Test Batch

Bước 1: Trên màn hình Batch Testing, nhấn vào " " ở phía bên phải bản ghi batch cần chỉnh sửa.

Hệ thống hiển thị màn hình EDIT BATCH TESTING



Bước 2: Chỉnh sửa các thông tin cần thiết, trong đó "Id" và "Code" là hai trường thông tin không thể chỉnh sửa

Bước 3: Nhấn **"Save",** hệ thống thông báo "**Update successful**"

1.1.9.5 Xoá Test Batch

Chỉ có thể xoá các batch không chứa batch step

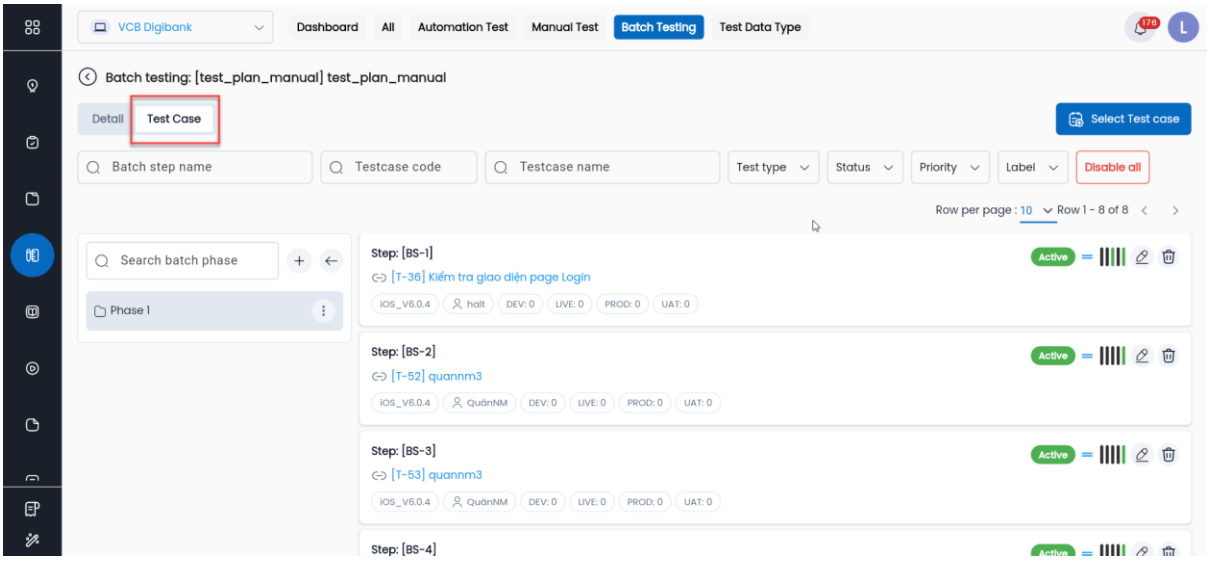
Tham chiếu đến mục [Xóa kịch bản kiểm thử]

Nếu Batch lựa chọn xoá đã được thực thi hoặc có chứa Batch step, hệ thống hiển thị thông báo không thể xóa.

1.1.9.6 Batch Step là gì?

Batch Step là các bước thực hiện trong Batch Testing, hiện tại hệ thống hỗ trợ xây dựng batch từ test case.

Trên màn hình thư viện test batch, nhấn vào batch bất kỳ, hệ thống hiển thị màn hình tab Detail của batch. Chọn tab "Test case" để xem các steps của batch.

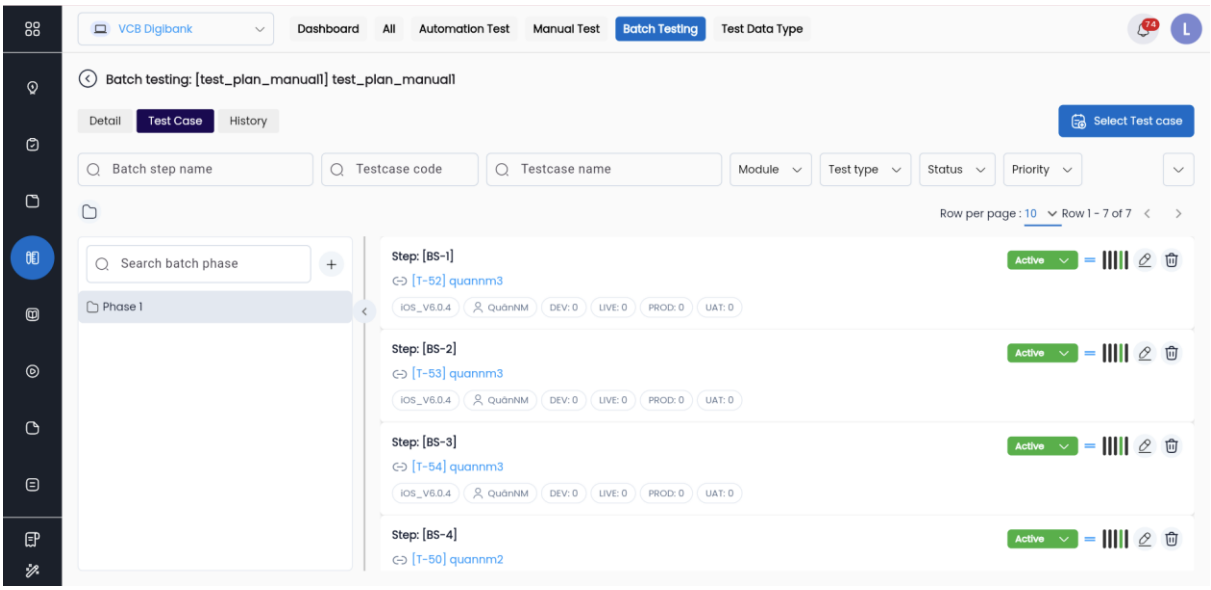


Trong đó:

Trường thông tin	**Bắt buộc**	**Định dạng**	**Ràng buộc**	**Mô tả**
Batch Step name		Text		Nhập tên batch step để thực hiện tìm kiếm theo tên
Test case code		Text		Tìm kiếm theo mã của test case
Test case name		Text		Tìm kiếm theo tên của test case
Test type		Combobox		Tìm kiếm theo loại của batch - Automation - Manual
Status		Combobox		Tìm kiếm trạng thái loại của batch batch:

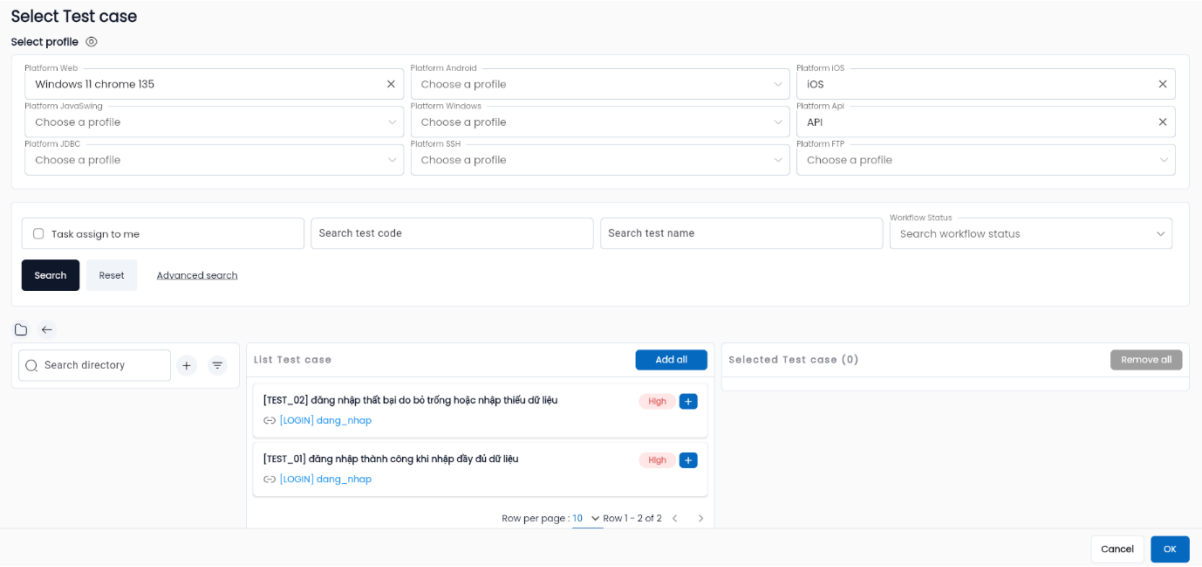
				- Active - Inactive
Label		Combobox		Tìm kiếm theo nhãn của batch
Priority		Combobox		Tìm kiếm theo mức độ ưu tiên - Highest - High - Medium - Low - Lowest
Disable all		Button		Click button "Disable all" cho phép vô hiệu hóa tất cả các batch steps. Hệ thống sẽ hiển thị Popup cảnh báo xác nhận vô hiệu hóa
Name	-	-	-	Tên của test batch step
Test Name	-	-	-	Tên của kịch bản test đã chọn (test case, test script, test suite)
Owner	-	-	-	Người tạo batch step
	-	Icon		Nhấn vào icon " ", hệ thống hiển thị màn hình cho phép người dùng người dùng thao tác chỉnh sửa với batch step
Select Test case	-	Button		Nhấn vào button "Select Test case" hệ thống hiển thị màn hình Select Test case

1.1.9.7 Select Test case



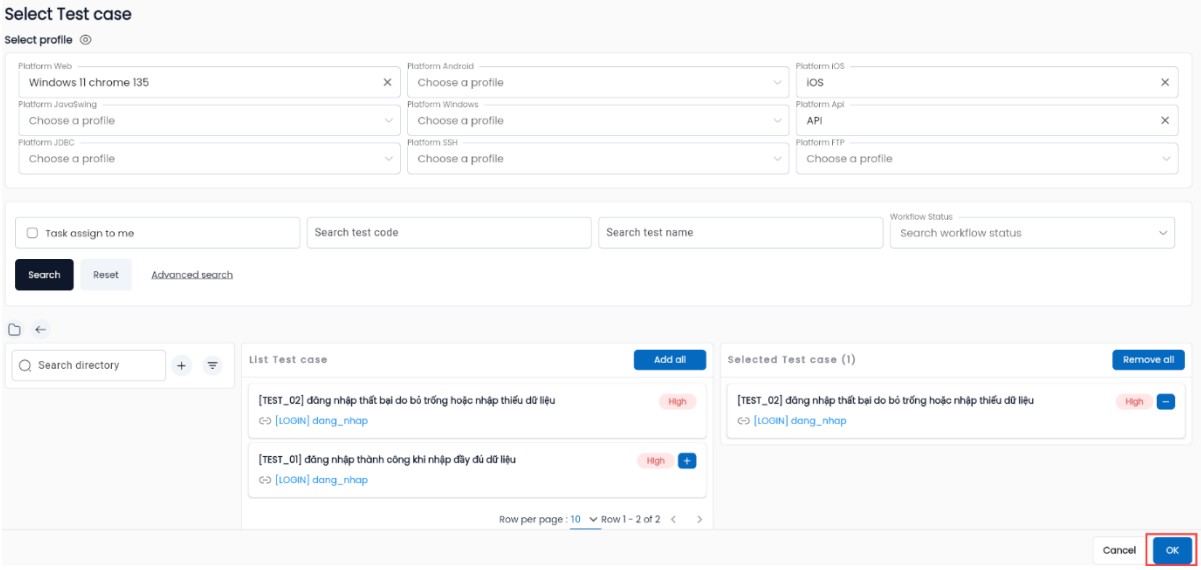
Lựa chọn test case cho Batch step

Bước 1: Trên màn hình danh sách Batch Step, nhấn "Select Test case", hệ thống hiển thị màn hình Select Test case và danh sách test case đã active của dự án, các thông tin dữ liệu và tìm kiếm.



Người dùng nhấn button cho việc chọn test case sẽ được sử dụng cho batch steps.

Sau khi chọn test case, nhấn "**OK**"



Hệ thống quay lại màn hình Batch Step đưa ra thông báo “Create successful” cùng với tên của test case đã chọn. Nhấn vào test case vừa thêm để hiển thị các thông tin của test case đó.

1.1.9.8 Edit Batch Step

Bước 1: Trên màn hình tab Batch Step, nhấn vào button “ ” ở phía bên phải batch step cần chỉnh sửa, hệ thống hiển thị màn hình Edit Batch Step

Cập nhật thông tin cần chỉnh sửa tương tự như tạo mới batch step

Bước 2: Nhấn “**OK**”, hệ thống thông báo “**Update successful**”

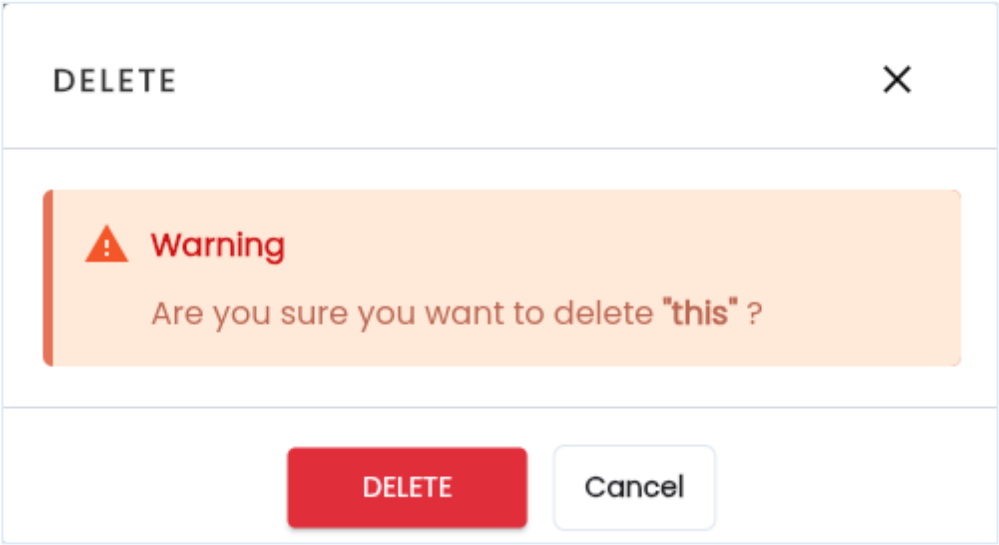
1.1.9.9 Xóa Batch Step

Chỉ có thể xoá batch step chưa được thực thi từ trong batch.

Trên màn hình tab Batch Step, nhấn vào button “ ” ở phía bên phải bản ghi step cần xoá

Hệ thống hiển thị popup xác nhận xoá

Bước 1:



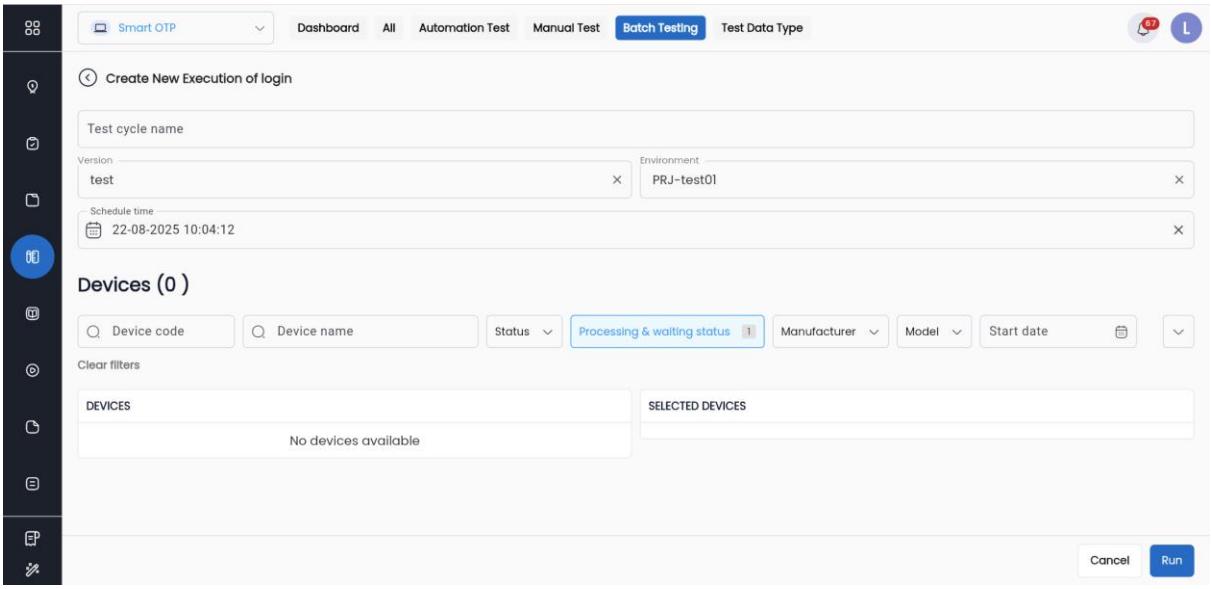
Bước 2: Nhấn **“DELETE”,** hệ thống thông báo “**Delete successful**”

Nếu batch đã được thực thi trước đó, hệ thống hiển thị thông báo không thể xóa

1.1.9.10 Thực thi test batch

Bước 1: Trên màn hình thư viện test batch, nhấn button “ ” ở phóa bên phải bản ghi batch cần thực thi

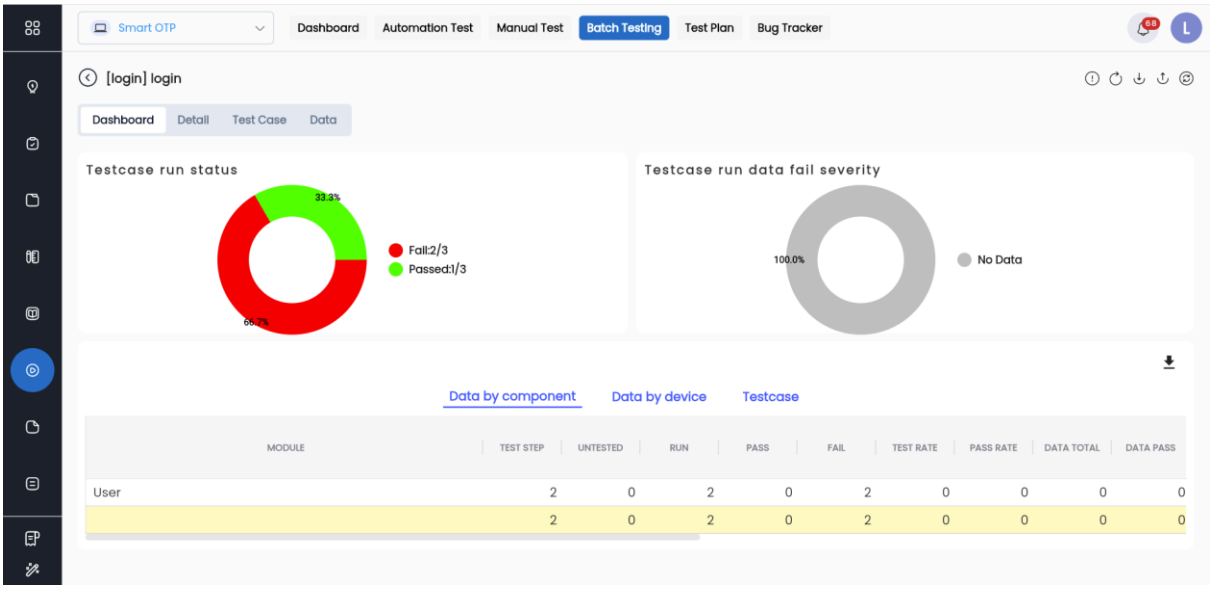
Hệ thống hiển thị màn hình:



Bước 2: Thiết lập lịch trình thực thi test batch, trong đó:

- Schedule date: Lịch trình thực thi, hệ thống mặc định là realtime, tuy nhiên người dùng có thể thay đổi vào một thời điểm khác trong tương lai.

Bước 3: Sau khi thiết lập lịch trình thực thi test batch, nhấn "Run", hệ thống điều hướng đến màn hình thực thi Batch



Trong quá trình run, trạng thái của test batch sẽ chuyển sang processing rồi đến kết quả thực thi (fail/ pass).

1.1.10 Manual Test

Manual test là kiểm thử thủ công, thực hiện hoàn toàn bằng thao tác của nhân viên kiểm thử. Với những kịch bản kiểm thử không thể thực hiện hoặc không cần thiết sử dụng công cụ tự động, **Centralize Automation Testing Suite** cho phép người dùng quản lý trong tab Manual Test.

Các kịch bản kiểm thử thuộc về Non-functional testing, người dùng cũng có thể quản lý tại Manual.

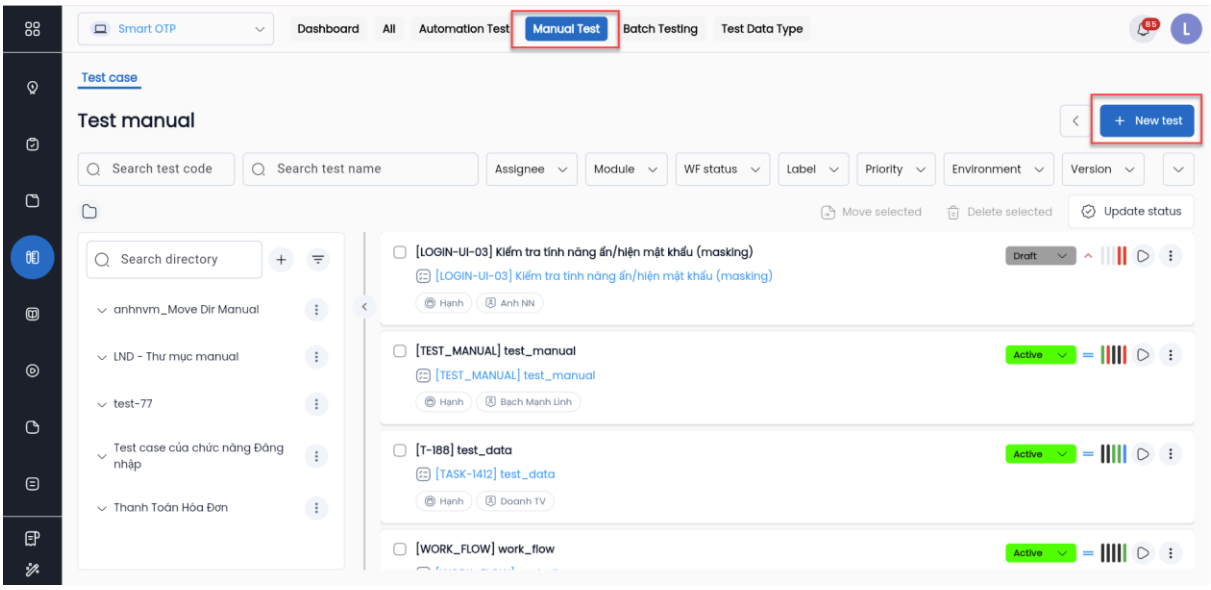
1.1.10.1 Các loại kiểm thử quản lý bởi Manual

Với các luồng kiểm thử chức năng nhỏ, tần suất lặp lại thấp, bộ dữ liệu đầu vào cho kịch bản kiểm thử ít, người dùng có thể lựa chọn kiểm thử thủ công thay vì kiểm thử tự động.

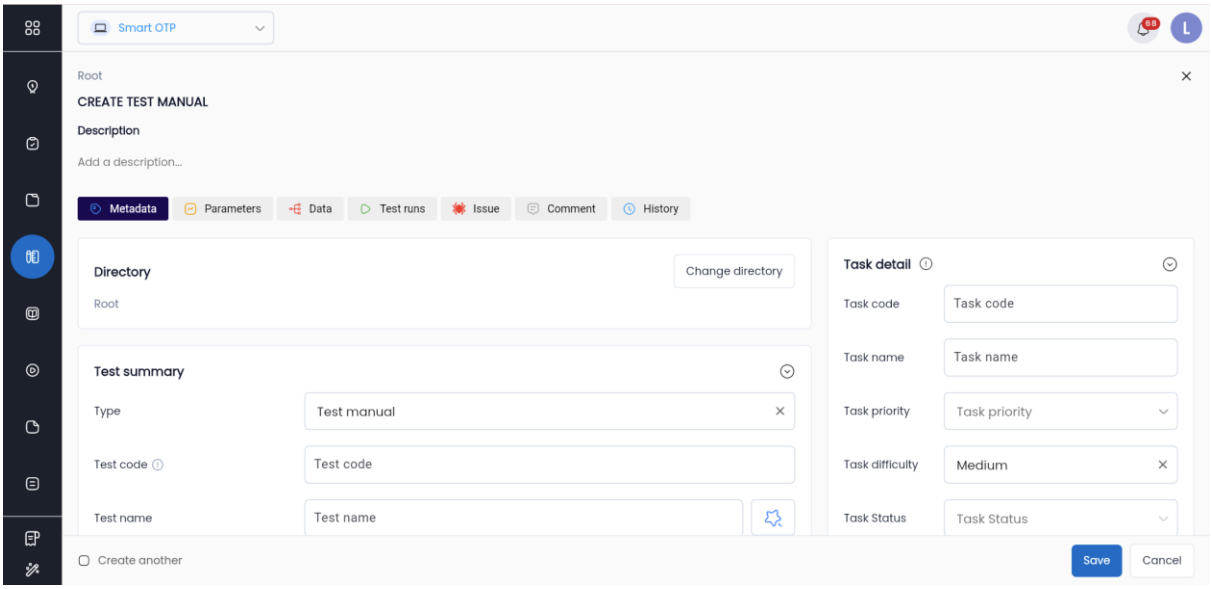
Manual Test cho phép người dùng cập nhật kết quả và quản lý kịch bản kiểm thử.

1.1.10.2 Khởi tạo Manual Test

Bước 1: Trên màn hình thư viện Manual, nhấn "**New test**"



Hệ thống hiển thị màn hình "**CREATE TEST MANUAL**"



Bước 2: Nhập đầy đủ và hợp lệ các trường thông tin bắt buộc, trong đó:

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Tab General information Phần thông tin chung của 1 Manual test. Tương tự Test script, tham chiếu tới mục Tạo mới một test script				
Tab Parameters Lưu trữ các Parameters của Manual Test				
Show description	-	Button		Cho phép người dùng ẩn/hiện trường dữ liệu "Description" của một Parameter trong test manual.
New parameter	-	Button		Cho phép người dùng tạo một Parameter mới cho manual.

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Clear parameter	-	Button		Cho phép người dùng xóa tất cả Parameter đã khai báo cho test manual.
Data Type	Có	Combobox		Kiểu dữ liệu của 1 parameter. Hệ thống khi khởi tạo parameter mới mặc định để là "String".
Name	Có	Text	Tên phải bắt đầu bằng chữ cái, chứa các ký tự chữ cái từ a-z (viết thường, viết hoa), chữ số và các ký tự "_","-"	Tên của 1 parameter.

Bước 3: Nhấn **"Save",** hệ thống thông báo "**Create successful**"

1.1.10.3 Edit Manual Test

Tham chiếu Edit Test script

1.1.10.4 Delete Manual

Tham chiếu Xóa Test script

1.1.10.5 Nhân bản Test Manual

Tham chiếu Duplicate Test script

1.1.10.6 Thực thi test Manual

Tham chiếu Run Test script

1.1.10.7 Export Json file

Tham chiếu Export Json file

1.1.10.8 Import Json file

Tham chiếu Import Json file

1.1.10.9 Move to directory

Tham chiếu Move to directory

1.1.11 Test API

1.1.11.1 Kiểm thử API

1.1.11.1.1 API Testing là gì?

API (Application Programming Interface) là một bộ giao diện và tập hợp các quy tắc mà một ứng dụng, dịch vụ hoặc thư viện cung cấp để cho phép các ứng dụng khác tương tác và sử dụng chức năng của nó.

API testing là một loại kiểm thử phần mềm liên quan đến việc kiểm thử các giao diện lập trình ứng dụng (APIs) một cách trực tiếp và là một phần của kiểm thử tích hợp để xác định xem hệ thống có đáp ứng các yêu cầu về tính năng, độ tin cậy, hiệu suất và bảo mật.

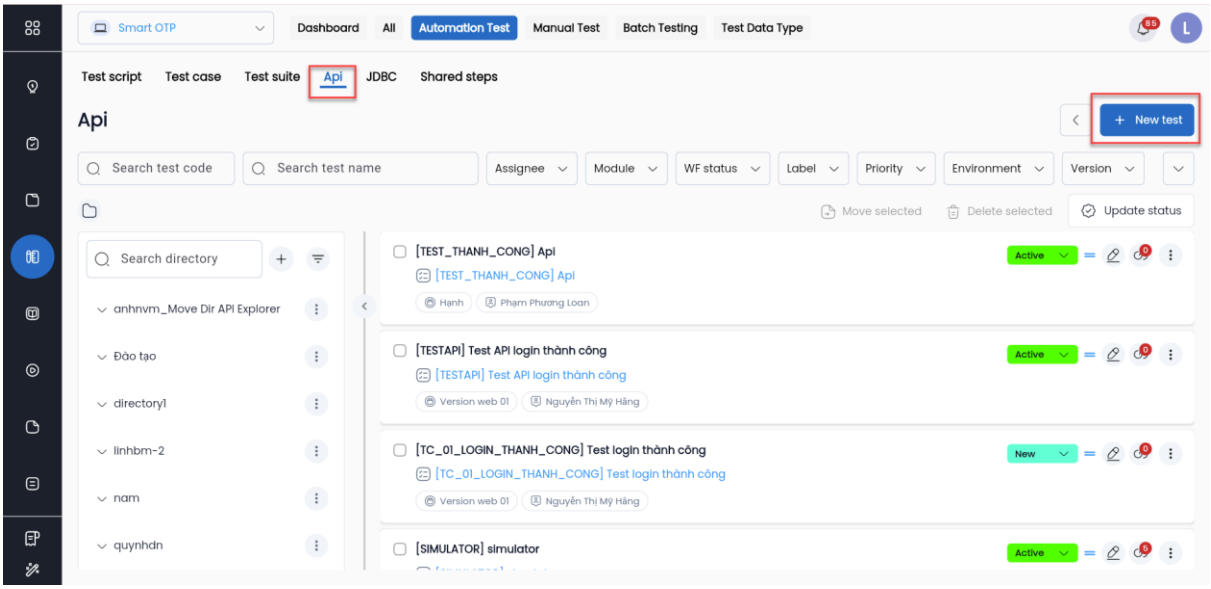
1.1.11.1.2 Mục tiêu của kiểm thử API

- ***Đảm bảo tính chính xác của dữ liệu***: đầu ra từ API phải khớp với kỳ vọng khi cung cấp đầu vào cụ thể
- ***Xác minh chức năng***: đảm bảo API thực hiện đúng các chức năng được định nghĩa.
- ***Kiểm tra hiệu suất***: đảm bảo API hoạt động nhanh chóng và hiệu quả, kể cả khi có lưu lượng lớn
- ***Bảo mật***: kiểm tra khả năng xử lý các vấn đề liên quan đến quyền truy cập, xác thực và bảo vệ dữ liệu.
- ***Xử lý lỗi***: đảm bảo API trả về thông báo lỗi đúng và hợp lý khi gặp vấn đề.

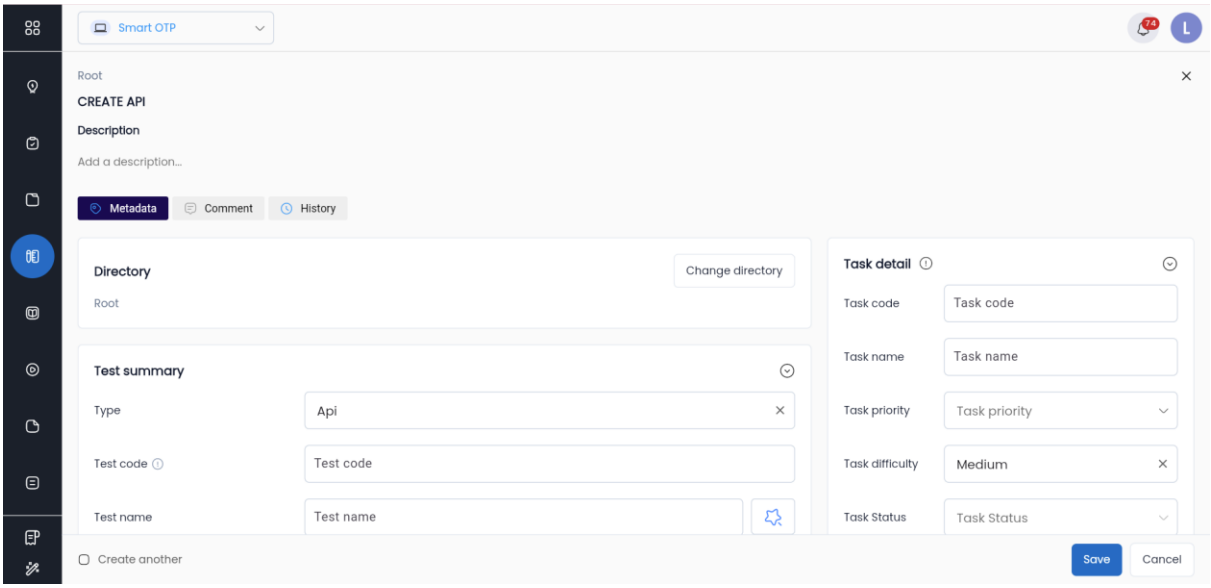
1.1.11.1.3 Kiểm thử API trên LLQ QA Platform

1.1.11.1.3.1 Tạo API

Bước 1: Trên màn hình thư viện API test, click "**New test**"



Hệ thống hiển thị màn hình “CREATE API”



Bước 2: Nhập đầy đủ và hợp lệ các trường thông tin bắt buộc, trong đó:

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
General information				
Directory	-	Button	Nếu Directory = Root thì test	Hiển thị danh sách các directory thuộc project

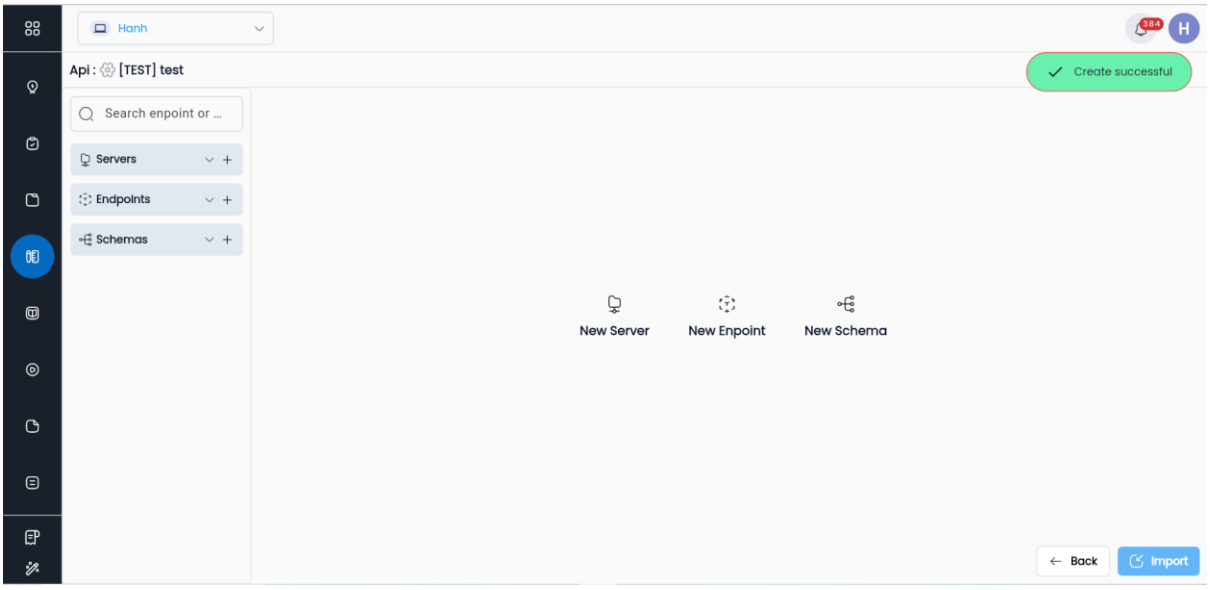
Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
			case thuộc directory gốc	ở tab API, cho phép người dùng chọn 1 directory cho API
Test code	-	Text	Nếu người dùng nhập code thì code phải là duy nhất, độ dài tối thiểu 2 ký tự, bắt đầu bằng chữ cái, chỉ chứa các ký tự từ a-z (viết thường, viết hoa), chữ số, "_" và "-"	Là mã dùng để định danh kịch bản kiểm thử trên hệ thống Nếu code bỏ trống, hệ thống sẽ tự sinh theo cấu trúc T-stt, trong đó stt là số thứ tự của test trong số các kịch bản kiểm thử được đánh mã tự động của Project
Test name	Có	Textbox		Là tên của kịch bản kiểm thử Người dùng nên đặt tên có nghĩa, liên quan đến chức năng/ trường hợp kiểm thử của test được khởi tạo
Module	Có	Combobox	Hiển thị danh sách module active của dự án kiểm thử	Cho phép người dùng chọn 1 module của dự án kiểm thử. Test thuộc module nào sẽ chọn module đó
Version	Có	Combobox	Hiển thị danh sách version	Cho phép người dùng chọn phiên bản của dự

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
			active của dự án kiểm thử	án kiểm thử. Test này thuộc phiên bản nào sẽ chọn phiên bản đó
Label	-	Button		Chọn 1 hoặc nhiều nhãn mà người dùng muốn gắn.
Purpose	-	Textarea		Cho phép người dùng nhập mục đích của test api
Precondition	-	Textarea		Cho phép người dùng nhập điều kiện tiên quyết của test api
Test steps	-	Textarea		Cho phép người dùng nhập các bước thực hiện trong test api
Expect result	-	Textarea		Cho phép người dùng nhập kết quả mong muốn của test api
Test data	-	Textarea		Cho phép người dùng nhập dữ liệu test của test api
Task Code	-	Text	Task code là duy nhất	Nhập task code Trường hợp không nhập task code, hệ thống tự sinh task code theo số thứ tự task của project

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Task Name	Có	Text		Là tên của task được giao khi thực hiện xây dựng kiểm thử API
Task priority	Có	Combobox		Mức độ ưu tiên của task Highest: Cao nhất High: Cao Medium: Trung bình Low: Thấp Lowest: Thấp nhất
Task difficulty	Có	Combobox		Phân cấp mức độ khó của task Very Hard: Rất khó Hard: Khó Medium: Trung bình Easy: Dễ Very Easy: Rất dễ
Assign to	-	Select	Chỉ hiển thị danh sách user trong project Nếu bỏ trống, hệ thống mặc định Assign to người tạo API	Là người thực hiện task của API này Sau khi click vào ô Assign to, hệ thống hiển thị màn hình danh sách user, click chọn một user và click OK
Schedule End Date	-	Calendar		Là ngày kết thúc thực hiện task

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Save	-	Button		Click "Save", hệ thống validate thông tin, nếu thành công thì api được lưu và hiển thị trên lưới danh sách API của directory đã chọn
Cancel	-	Button		Click "Cancel" để huỷ bỏ thao tác tạo API, hệ thống reset form và quay lại màn hình thư viện API

Bước 3: Click "**Save**", hệ thống thông báo "Create successful" và điều hướng đến màn hình config API

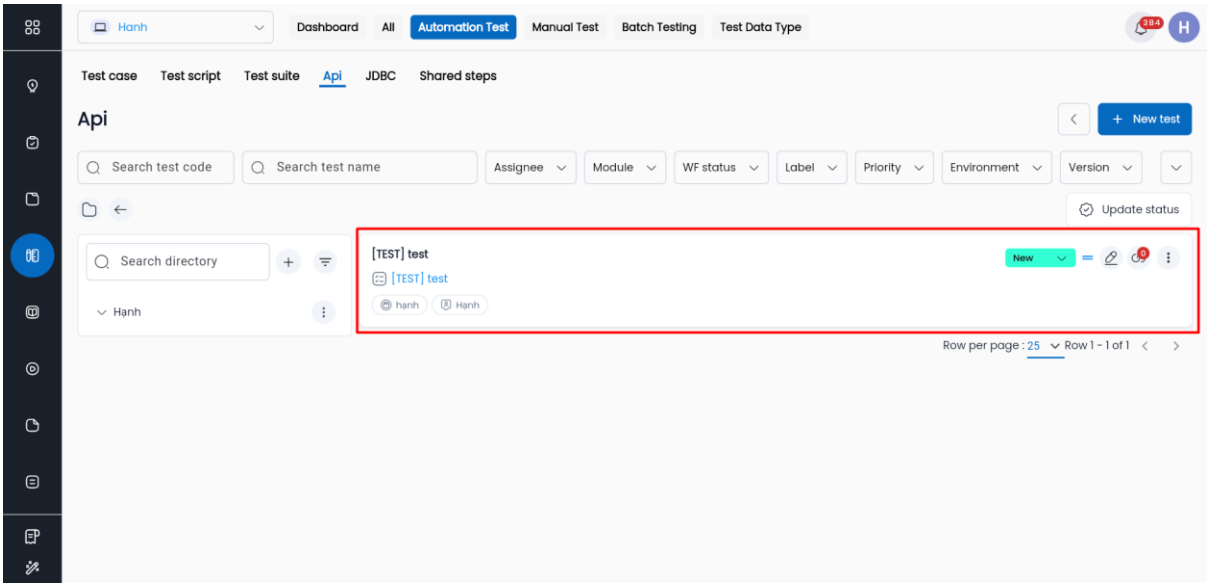


Click "**Back**" để quay lại màn hình Test Management tab API

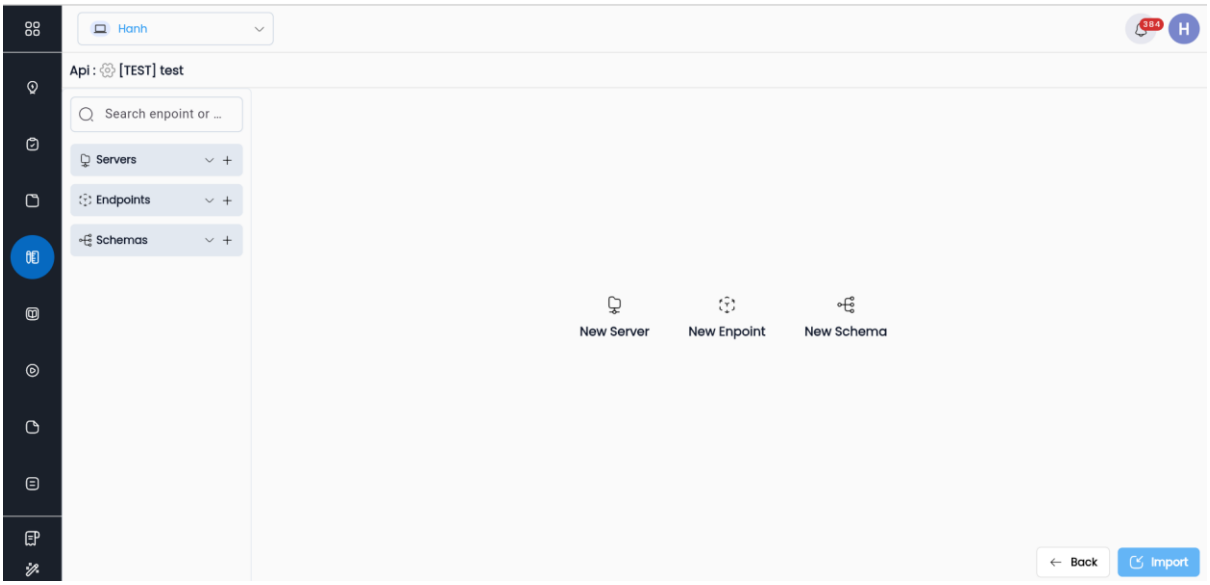
1.1.11.1.3.2 Cấu hình Test API – Low code

Low-code: là chức năng cấu hình api, xây dựng Servers, Endpoints, Schemas để thực hiện các bước trong 1 API Test.

Để truy cập vào màn hình low-code test api, người dùng thực hiện click vào dòng dữ liệu của API tương ứng.

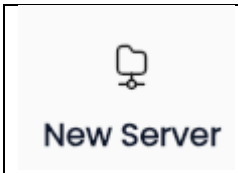
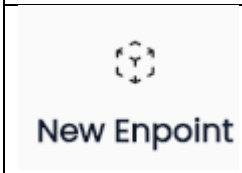
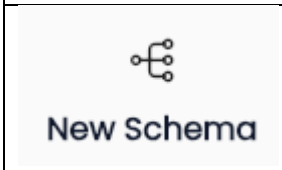
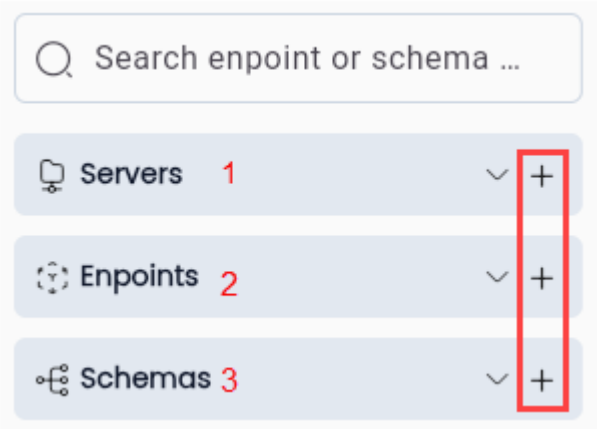
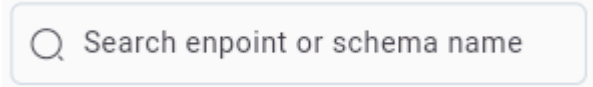
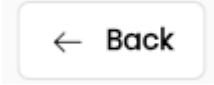


Hệ thống hiển thị màn hình Low-code



Các chức năng trong màn hình low-code

Chức năng	Mô tả

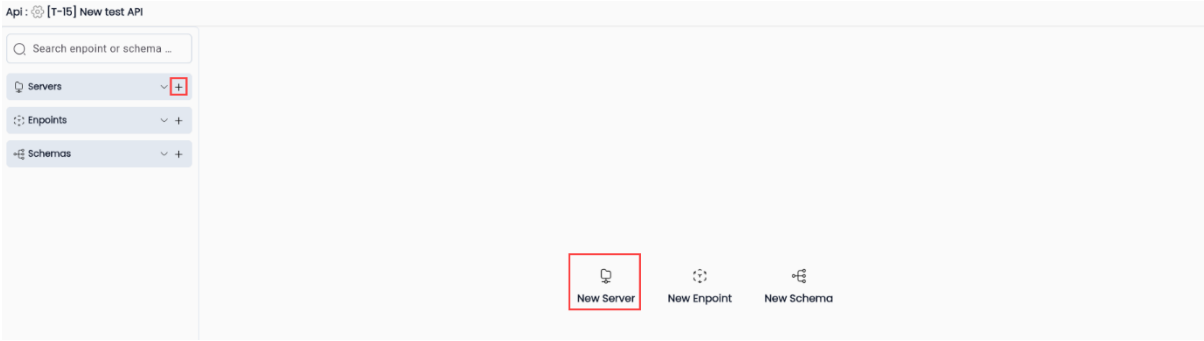
	Tạo mới Server
	Tạo mới Endpoint
	Tạo mới Schema
	1. Danh sách các servers 2. Danh sách các Endpoints 3. Danh sách các Schemas Click button + để tạo mới tương ứng với chức năng
	Search: Chức năng cho phép người dùng tìm kiếm các Endpoint hoặc Schema name
	Chức năng cho phép người dùng chuyển sang màn hình metadata của test api từ màn hình low-code
	Click button “Back” để quay lại màn hình thư viện script

1.1.11.1.4 Servers

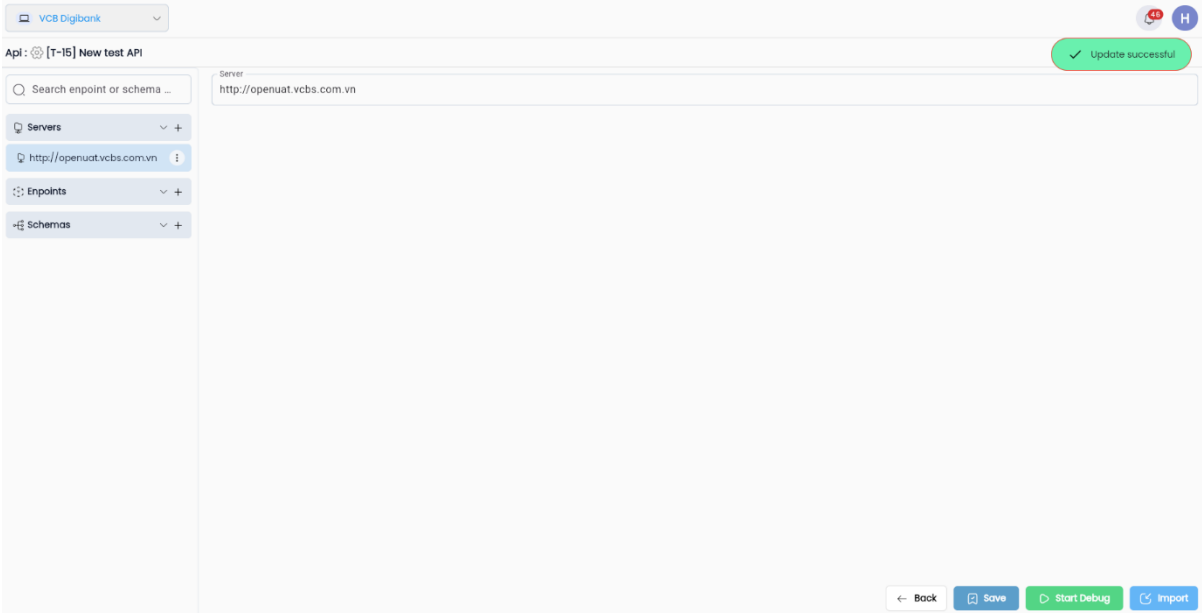
- Servers được tạo để nhằm mục đích thống kê, lưu trữ các server, chuỗi chứa tên miền của URL sẽ thực hiện gọi API.
- Server của API sẽ được khai báo khi thực hiện gọi test api trong test case, test script hay test suite. Tại màn hình config API Explore, khi play Endpoint, servers được select từ danh sách sever của API.

1.1.11.1.4.1 Thêm mới servers

Bước 1: Trong màn hình Low-code, thực hiện click "**New Server**" hoặc button " + "để thực hiện tạo mới 1 server



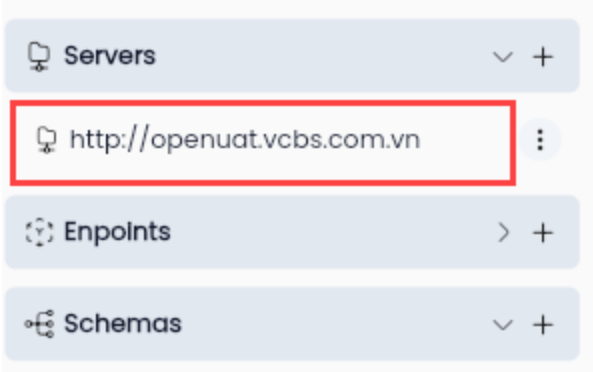
Bước 2: Hệ thống hiển thị màn hình thêm mới server, người dùng tiến hành nhập và click "**Save**"



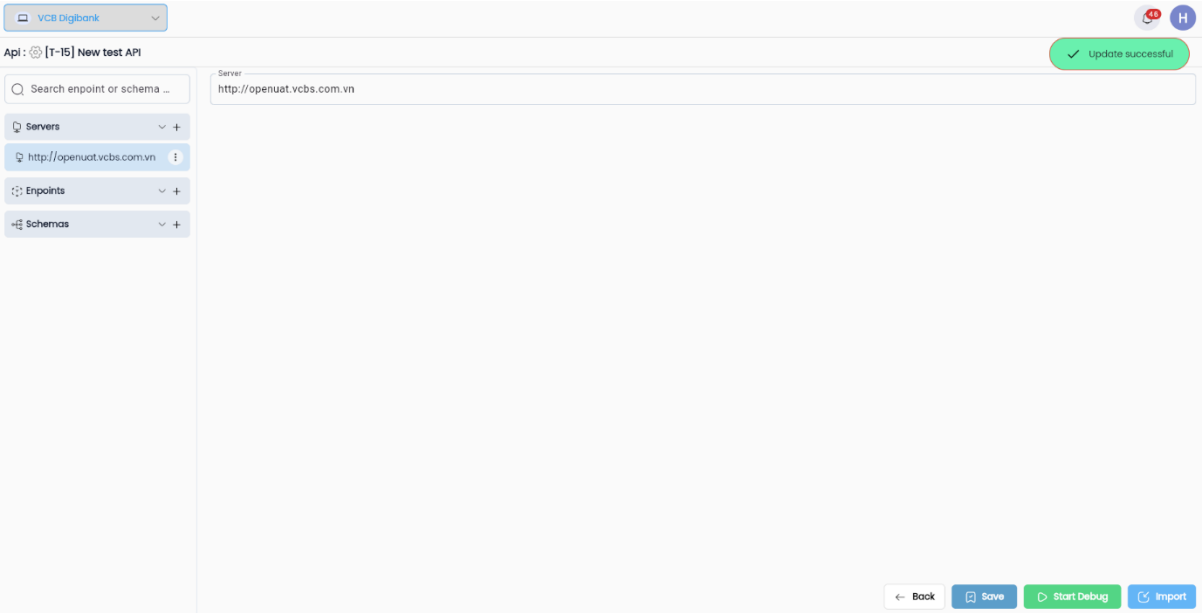
Thêm mới thành công hệ thống thông báo "Update successful" và hiển thị server đã khai báo trong danh sách servers

1.1.11.1.4.2 Chỉnh sửa server

Bước 1: Tại màn hình Low-code của test API, người dùng thực hiện chọn dòng dữ liệu server muốn thay đổi.



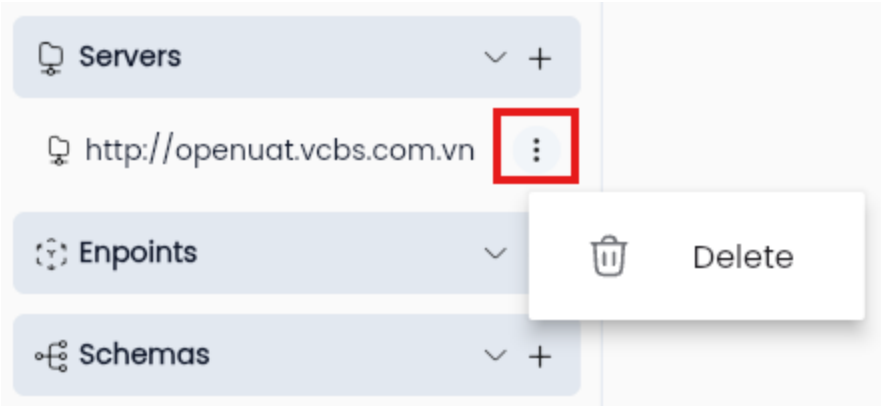
Bước 2: Chỉnh sửa thông tin cần thay đổi và click “Save”



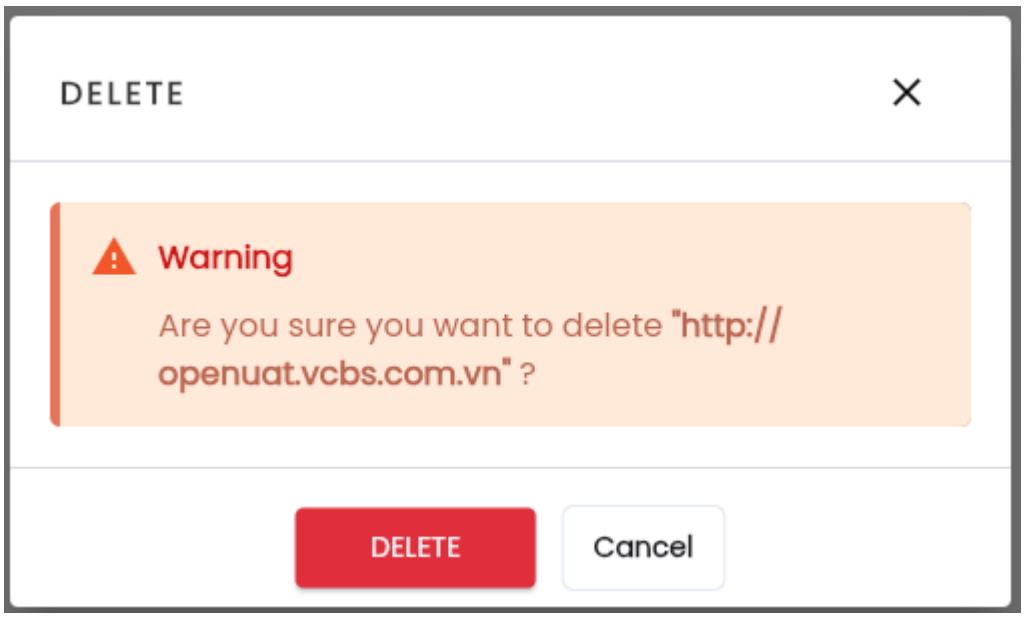
Hệ thống hiển thị thông báo “Update successful” và hiển thị server đã chỉnh sửa trong danh sách servers.

1.1.11.1.4.3 Delete Servers

Bước 1: Tại màn hình Low-code của test API, người dùng thực hiện click button “ ” trên dòng dữ liệu server muốn xóa và chọn **Delete**.



Hệ thống hiển thị popup xác nhận xoá



Bước 2: Click **"DELETE",** hệ thống thông báo "Update successful"

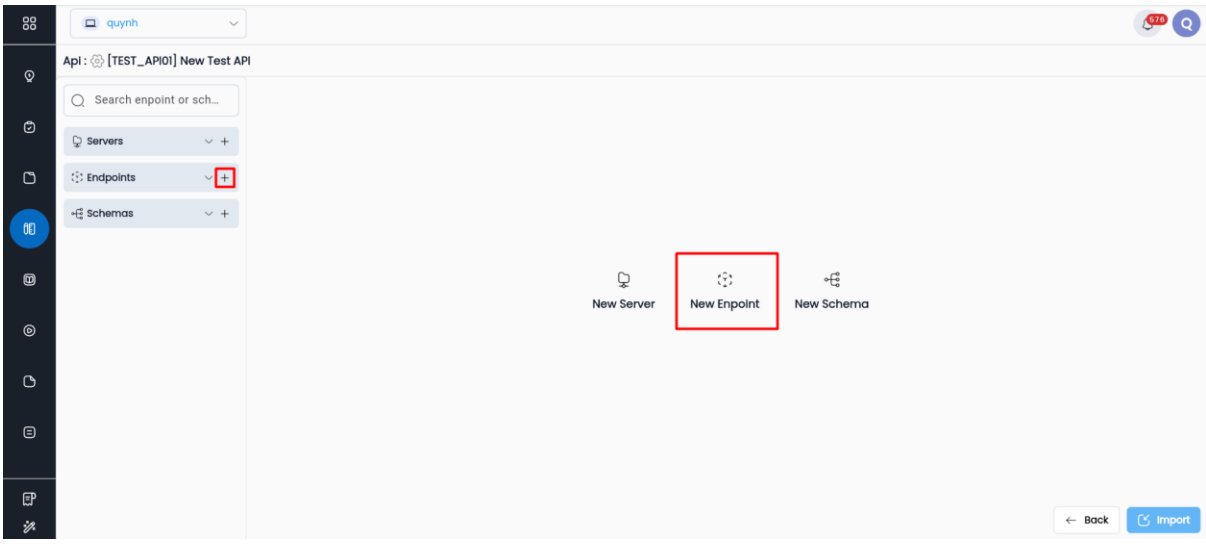
1.1.11.1.5 Endpoints

Endpoint là một URL cụ thể được thiết kế để thực hiện một hành động hoặc cung cấp một tài nguyên từ máy chủ.

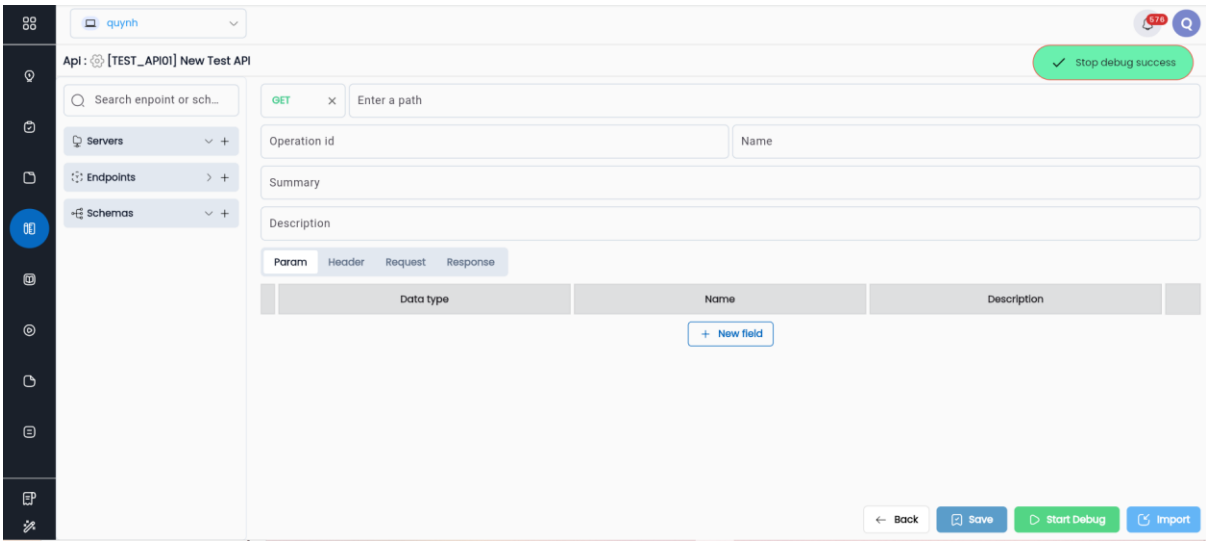
Endpoints được tạo để cấu hình các thông tin liên quan tới việc 1 lần gọi API thông qua HTTP Request

1.1.11.1.5.1 Thêm mới Endpoints

Bước 1: Trong màn hình Low-code, thực hiện click "**New Endpoint**" hoặc button " + "để thực hiện tạo mới 1 Endpoint



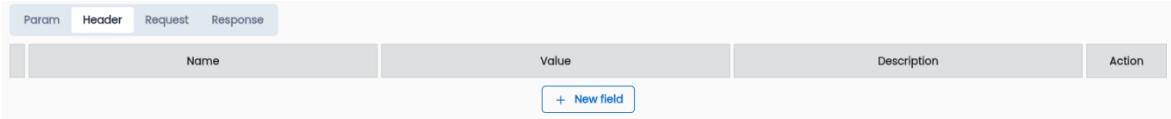
Bước 2: Hệ thống hiển thị màn hình thêm mới Endpoint, người dùng tiến hành nhập các thông tin cần thiết



Trong đó :

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
HTTP Method	Có	Combobox		Phương thức HTTP: là những hành động mà nhân viên kiểm thử thực hiện khi tiếp cận với API

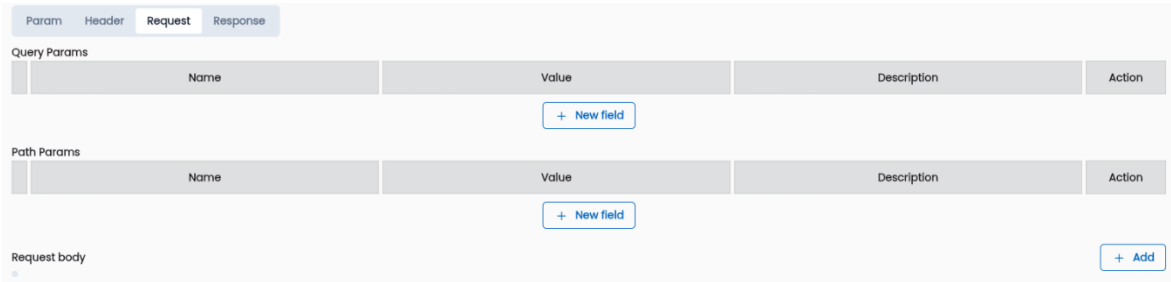
Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				+ Get: truy xuất dữ liệu từ máy chủ + Post: thêm dữ liệu vào một tệp hoặc tài nguyên hiện có trong một máy chủ + Put: thay thế một tệp hoặc tài nguyên hiện có của một máy chủ + Delete: xoá dữ liệu khỏi máy chủ + Patch: update các trường data được yêu cầu
Path	Có	Text		Đường dẫn đến vị trí thực hiện http method
Name	Có	Text		Tên của request Endpoint
Summary		Text		Tên gợi nhớ của api
Operation id		Text		Định dạng của api
Description		Text		Mô tả cho api
Param Quản lý params trên hệ thống được sử dụng để map với các param trong request API tại tab Header, Request, Response				

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
+ New field	-	Button		Click button để tạo thêm 1 tham số parameter mới
Name	Có	Text		Tên tham số parameter
Data type	Có	Combobox		Loại dữ liệu của tham số. Hệ thống mặc định khi tạo mới là "String"
Description	-	Text		Mô tả của tham số
Header Cho phép người dùng truyền Header cho API 				
+ New field	-	Button		Click button để tạo thêm 1 tham số header mới Khi click button, màn hình Header Property để tạo Value cho Header Tham chiếu tới mục Header
Name	Có	Text		Tên header

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Value	Có	AutoComplete		Sau khi "Save" tại màn hình Header Property, value/name tùy thuộc loại biến người dùng sử dụng sẽ là Value của tham số header này.
Description	-	Text		Mô tả của header

Request

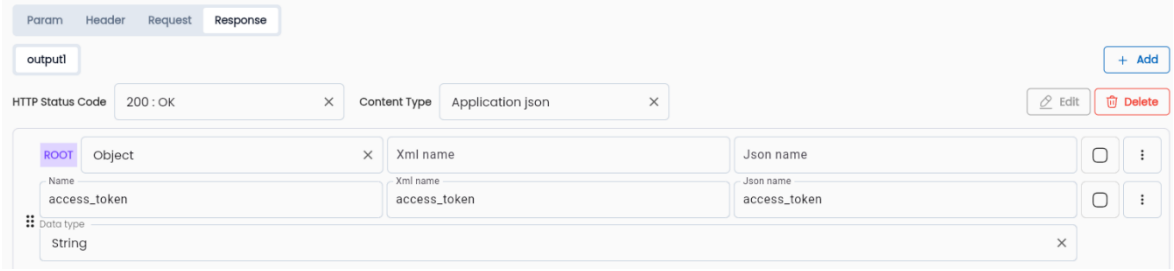
Cho phép người dùng truyền Request cho API



Request	-	Tab		Thông tin Api request tham chiếu tới mục Request API

Response

Cho phép người dùng cấu hình response cho API



Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Response	-	Tab		Thông tin Api response tham chiếu tới mục Response API
Save	-	Button		Click "Save" hệ thống validate thông tin, nếu thành công thì api được lưu và hiển thị thông báo "Update successful", thông tin Endpoints được thể hiện trên danh sách Endpoints
← Back	-	Button		Click "Back" để huỷ bỏ thao tác tạo API, hệ thống reset form và quay lại màn hình thư viện API

Thêm mới thành công hệ thống thông báo "***Update successfull***" và hiển thị server đã khai báo trong danh sách servers

1.1.11.1.5.2 Header

API header giống như một nguồn thông tin bổ sung cho mỗi lệnh gọi API mà nhân viên kiểm thử thực hiện, là đại diện cho meta-data được liên kết với Request và Response API.

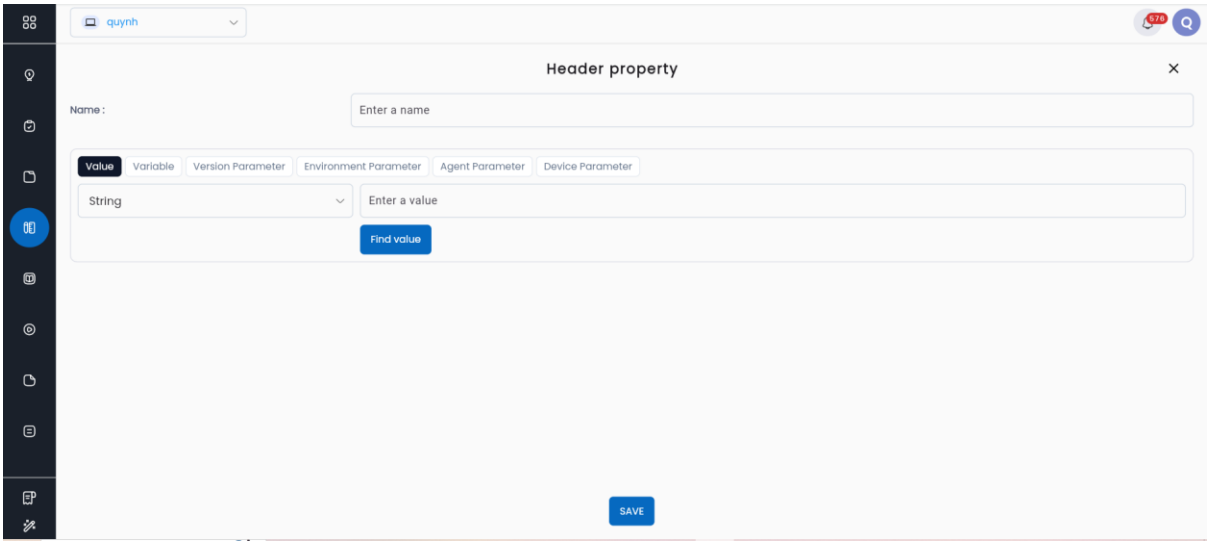
Headers giúp người dùng theo dõi mọi sự cố tiềm ẩn, cho biết các thông tin về:

- Request và Response Body
- Request Authorization

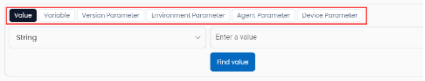
- Response Caching
- Response Cookie

Tạo Header cho API:

Bước 1: Trên tab Header, click button "**Add**", hệ thống hiển thị màn hình Header property của Header



Bước 2: Nhập đầy đủ và hợp lệ các trường thông tin bắt buộc, trong đó:

Trường thông tin	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Name	Có	Text		Tên của Header
Type 	Có			Các kiểu giá trị của tham số Header Có thể chọn 1 trong các tab phù hợp với mục đích sử dụng
Data type	Có	Combobox		Kiểu dữ liệu
Name	Có	Text	Chỉ hiển thị khi	Tên của tham số đã được khai báo trong project

			type khác “Value” Giá trị phải bắt đầu bằng 1 chữ cái, bắt buộc bao gồm chữ số và không được sử dụng kí tự đặc biệt (ngoại trừ “_”)	hoặc tên của biến (Variable)
Value	Có	Text	Chỉ hiển thị khi type = “Value”	Nhập giá trị trực tiếp của tham số Header
Find Value	-	Button	Chỉ hiển thị khi type = “Value”	Click vào Find Value hệ thống sẽ hiển thị danh sách các giá trị đã được định nghĩa trước đó của dự án
Find Variable	-	Button	Nếu Type là Variable	Click vào Find Variable hệ thống sẽ hiển thị danh sách các tham số đã được định

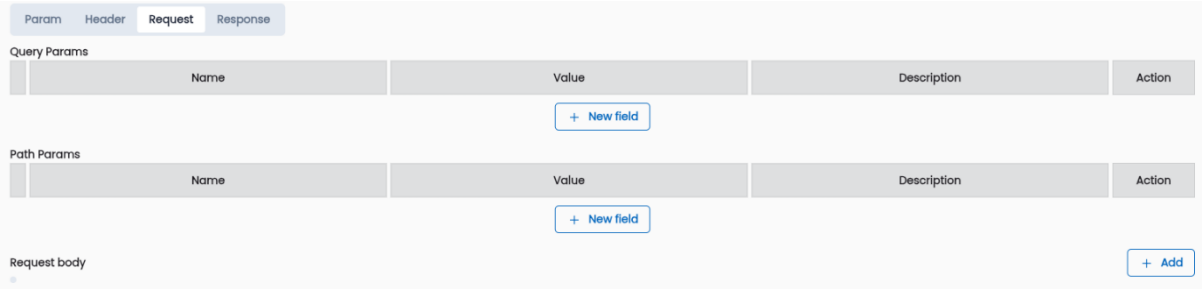
				nghĩa trước đó của dự án

Bước 3: Click **"Save"**

1.1.11.1.5.3 Request API

Request api bao gồm tất cả cả dữ liệu đầu vào của 1 Api

Trên tab Request của Api

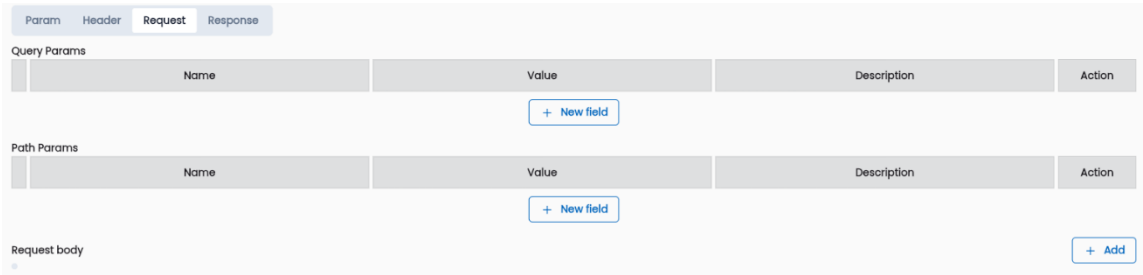


Trong đó:

Lưu ý: Nếu các param/variable được sử dụng trong API được khởi tạo bằng cách sử dụng các params trên hệ thống thì có thể khởi tạo tham số hệ thống tại tab Param của Endpoints và map với các param/variable trong 1 request API. Nếu không map thì hệ thống sẽ tự động tạo tham số hệ thống mới.

Trường thông tin	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Query Params list Quản lý các params được truyền trên đường dẫn URL theo giá trị "key-value" tương ứng với "name-value" 				
 	-	Button		Thêm mới 1 query params Click vào button hệ thống sẽ hiển thị màn hình

				“Query Params property”
	-	Button		Click vào để xoá 1 query params
	-	Button		Click vào để chỉnh sửa 1 query params Thông tin chỉnh sửa tương tự thông tin thêm mới
	-	Button		Di chuyển vị trí giữa các params(có thể khác loại params nếu)
Name	Có	Text		Tên query param
Value	Có	AutoComplete	Nếu thực hiện thêm mới thì giá trị là bắt buộc	Giá trị Value được xác định khi người dùng click button “Save” trong màn hình “Query Params property” sau khi hoàn thành thao tác thêm mới hoặc

				chỉnh sửa query param. Thao tác thêm mới hoặc chỉnh sửa tương tự Header Tham chiếu tới mục Header
Path params list Quản lý các path params được truyền trên đường dẫn URL theo giá trị “url/{path-param}”  				
New field	-	Button		Thêm mới 1 path params Click vào button hệ thống sẽ hiển thị màn hình “Path Params property”
	-	Button		Click vào để xoá 1 path params
	-	Button		Click vào để chỉnh sửa 1 path params

				Thông tin chỉnh sửa tương tự thông tin thêm mới
⠿	-	Button		Di chuyển vị trí giữa các params(có thể khác loại params)
Name	Có	Text		Tên path param
Value	Có	AutoComplete	Nếu thực hiện thêm mới thì giá trị là bắt buộc	Giá trị Value được xác định khi người dùng click button “Save” trong màn hình “Path Params property” sau khi hoàn thành thao tác thêm mới hoặc chỉnh sửa path param. Thao tác thêm mới hoặc chỉnh sửa tương tự Header Tham chiếu tới mục Header

Body list

Bao gồm 6 loại request Type:

None

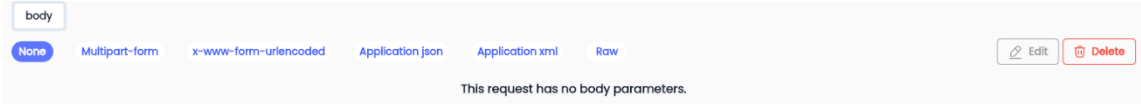
Multipart-form

x-www-form-urlencoded

Application json

Application xml

Raw



+ Add	-	Button		Thêm mới 1 body request của API Click button hệ thống sẽ hiển thị màn hình "Add a request"
Delete	-	Button		Xóa 1 body request của API

Add a request

Name	Có	Text		Tên của request body
Request Type	Có	Combobox		Yêu cầu người dùng chọn 1 trong 5 request Type tùy vào mục đích sử dụng Sau khi thêm mới có thể thay đổi.

Lưu ý: Mỗi 1 API có thể tạo nhiều request body, tuy nhiên mỗi type chỉ được tạo 1 lần trong 1 API.

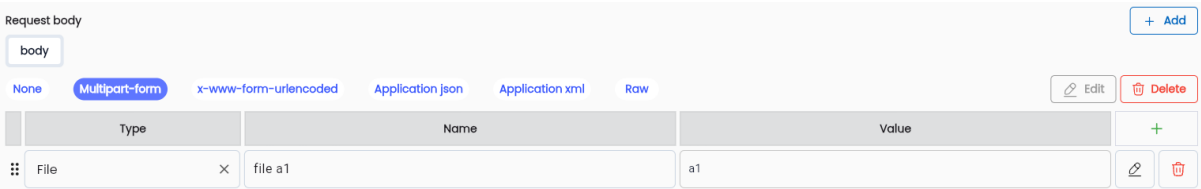
- Body request type: None

Request Type None sẽ xác định API Request không sử dụng body parameters.

Lưu ý: Nếu API không sử dụng request body, không cần thiết tạo ra 1 request body có type là None.

- Body request type: Multipart-form

Request Type Multipart-form sẽ xác định API Request có sử dụng nếu API có hành động tải lên File



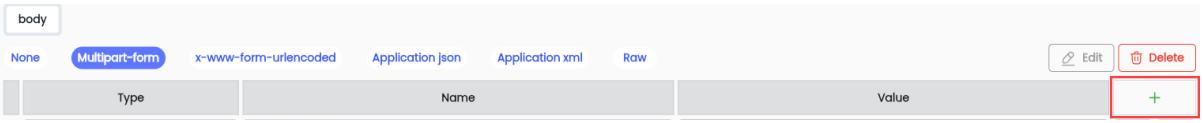
Trong đó

Trường thông tin	Bắt buộc	Định dạng	Ràng buộc	Mô tả

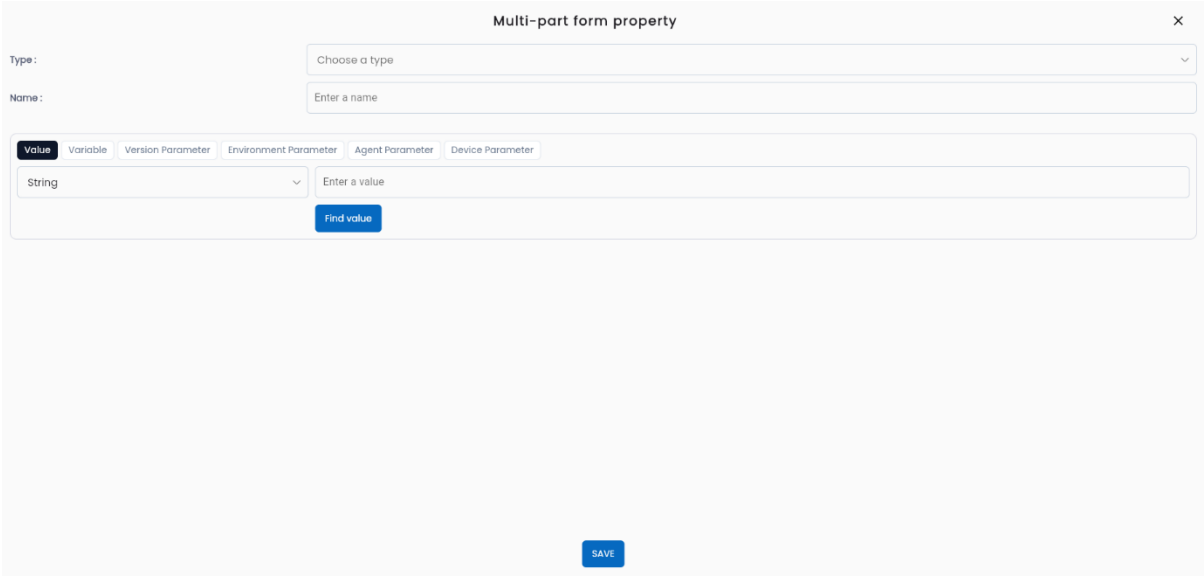
+	-	Button		Click vào để thêm mới 1 params trong multipart-form
	-	Button		Click vào để xoá 1 params trong multipart-form
	-	Button		Click vào để chỉnh sửa 1 params trong multipart-form
Type	Có	AutoComplete		Type của 1 params trong multipart-form, thể hiện 1 trong 2 giá trị: + Data + File
Name	Có	Text		Tên param
Value	Có	AutoComplete	Nếu thực hiện thêm mới thì giá trị là bắt buộc	Giá trị Value được xác định khi người dùng click button "Save" trong màn hình "Params property" sau khi hoàn thành thao tác thêm mới hoặc chỉnh sửa param.

Tạo Multipart-form request cho API:

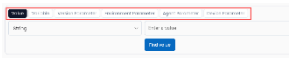
Bước 1: Trên tab Multipart-form, click button "+"



Bước 2: Nhập đầy đủ và hợp lệ các trường thông tin bắt buộc, trong đó:



Trong đó

Trường thông tin	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Type	Có	Combobox		Loại multi-part form, người dùng tiến hành chọn 1 trong 2 : + Data + File
Name	Có	Text		Tên của Header
Type 	Có			Các kiểu giá trị của tham số Header Có thể chọn 1 trong các tab phù hợp với mục đích sử dụng
Data type	Có	Combobox		Kiểu dữ liệu
Name	Có	Text	Chỉ hiển thị khi	Tên của tham số đã được khai báo trong

			type khác “Value” Giá trị phải bắt đầu bằng 1 chữ cái, bắt buộc bao gồm chữ số và không được sử dụng kí tự đặc biệt (ngoại trừ “_”)	project hoặc tên của biến (Variable)
Value	Có	Text	Chỉ hiển thị khi type = “Value”	Nhập giá trị trực tiếp của tham số Header
Find Value	-	Button	Chỉ hiển thị khi type = “Value”	Click vào Find Value hệ thống sẽ hiển thị danh sách các giá trị đã được định nghĩa trước đó của dự án
Find Variable	-	Button	Nếu Type là Variable	Click vào Find Variable hệ thống sẽ hiển thị danh sách các tham số đã được định

				nghĩa trước đó của dự án
Data type	Có	Combobox		Kiểu dữ liệu của multipart-form Ngoài các kiểu dữ liệu cho data thông thường , bổ sung thêm các kiểu dữ liệu dành cho File: + File + Elements + Repository Element + Password + Null + Empty String + RandomValue + Json + Excel Workbook + Excel Sheet + Excel Row + Excel Cell + Match Result + Schema

Bước 3: Click **"Save"**

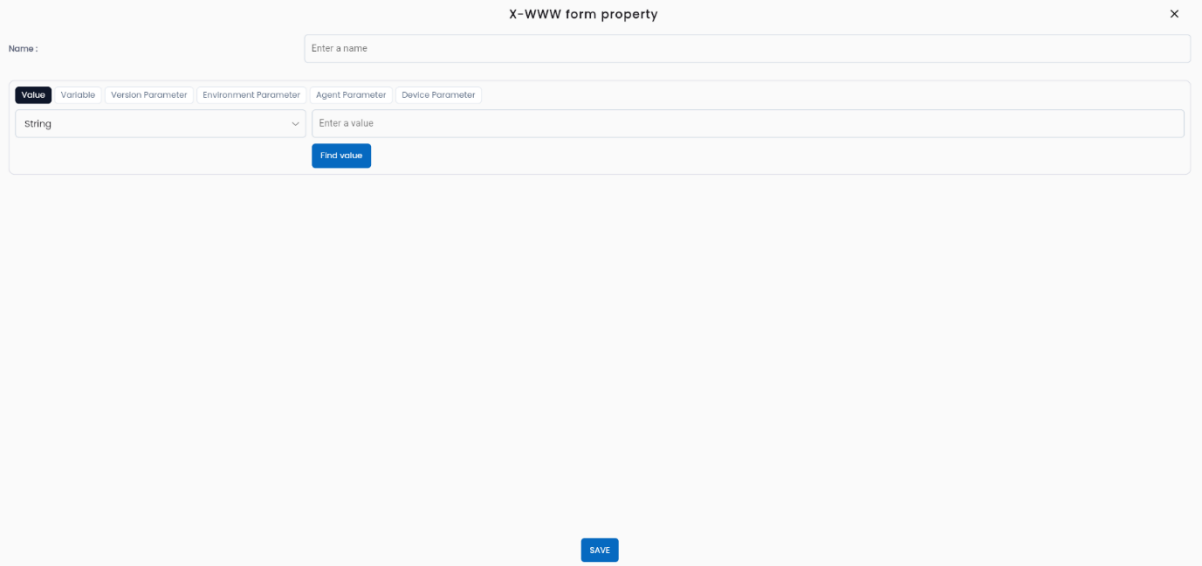
Bước 4: Click **"Ok"** để hoàn thành.

❖ Body request type: x-www-form-urlencoded

Tạo x-www-form-urlencoded request cho API:

Bước 1: Trên tab x-www-form-urlencoded, click button "+"

Bước 2: Nhập đầy đủ và hợp lệ các trường thông tin bắt buộc



Trong đó:

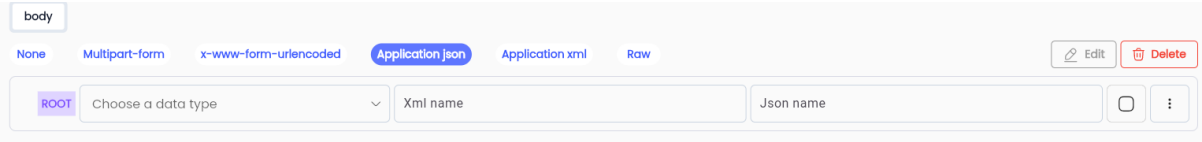
Trường thông tin	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Name	Có	Text	Không trùng lặp, không rỗng	Khóa định danh cho dữ liệu
Value	-	Text		Giá trị tương ứng với khóa

Bước 3: Click **"Save"**

Bước 4: Click **"OK"**

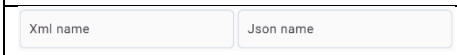
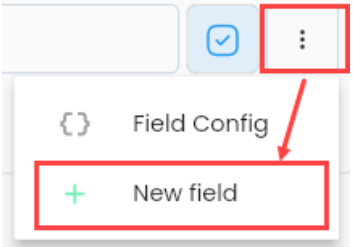
❖ **Body request type: Application json.**

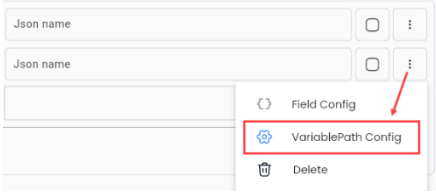
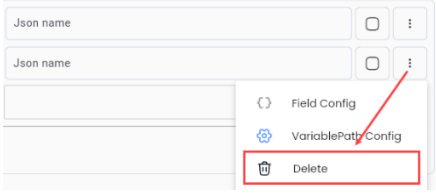
Màn hình body request type là Application json.



Trong đó

Trường thông tin	Bắt buộc	Định dạng	Ràng buộc	Mô tả
ROOT	-	Label		Thể hiện tham số được sử dụng sẽ là tham số gốc của request
Choose a data type	-	Combobox		Thực hiện chọn kiểu dữ liệu cho params json. Ngoại trừ các kiểu dữ liệu nguyên thủy thông thường , thêm vào đó có thể lựa chọn kiểu dữ liệu : + Object + Schema
Is list?	-	Checked Button		Nếu button checked : Xác định dữ liệu trên tham số trong json đó trở thành 1 List. VD : Nếu data Type của biến là “String” được chọn là

				1 list.Dữ liệu Type nhập vào cho biến này phải là 1 List<String>
	-	Text		Tên sẽ được sử dụng trong request để test api Nếu request là application json sẽ sử dụng "json name" Nếu request là application xml sẽ sử dụng "Json xml"
	-	Button		Click vào button hệ thống hiển thị màn hình "XML AND JSON OPTIONS".
	-	Button	Button chỉ xuất hiện trên tham số có kiểu	Click vào button hệ thống sẽ tạo thêm 1 tham số nằm trong Object

			dữ liệu “Object”	
	-	Button	Tham số gốc có kiểu dữ liệu là “Object”	Click vào button hệ thống sẽ tạo thêm 1 tham số nằm trong Object
	-	Button	Button chỉ xuất hiện trên tham số nằm trong tham số có kiểu dữ liệu “Object” hoặc tham số có kiểu dữ liệu là “Schema”	Click vào button “VariablePath Config” hệ thống hiện màn hình “Variable Path”
	-	Button	Button chỉ xuất hiện trên tham số nằm trong tham số có kiểu dữ liệu “Object”	Click vào button “Delete”, hệ thống sẽ xóa tham số nằm trong Object

Name	-	Text	Tham số gốc có kiểu dữ liệu là "Object"	Tên của tham số trên hệ thống 1 parameter hoặc variable đã được khai báo trên hệ thống

Lưu ý : Trường dữ liệu "name" và "jsonName" là khác nhau:

+ name là tên của tham số trên hệ thống chỉ được sử dụng để cấu hình giá trị cho tham số json, name sẽ không được sử dụng để cấu hình body request cho api.

+ jsonName là tên của tham số trong body request api.

Kiểu dữ liệu Schema sẽ hỗ trợ việc test API trong request body có các kiểu dữ liệu Object lồng nhau, lúc đó parameter Root có kiểu dữ liệu là Object và các parameter con sẽ có kiểu dữ liệu là Schema

- Chức năng Field Config

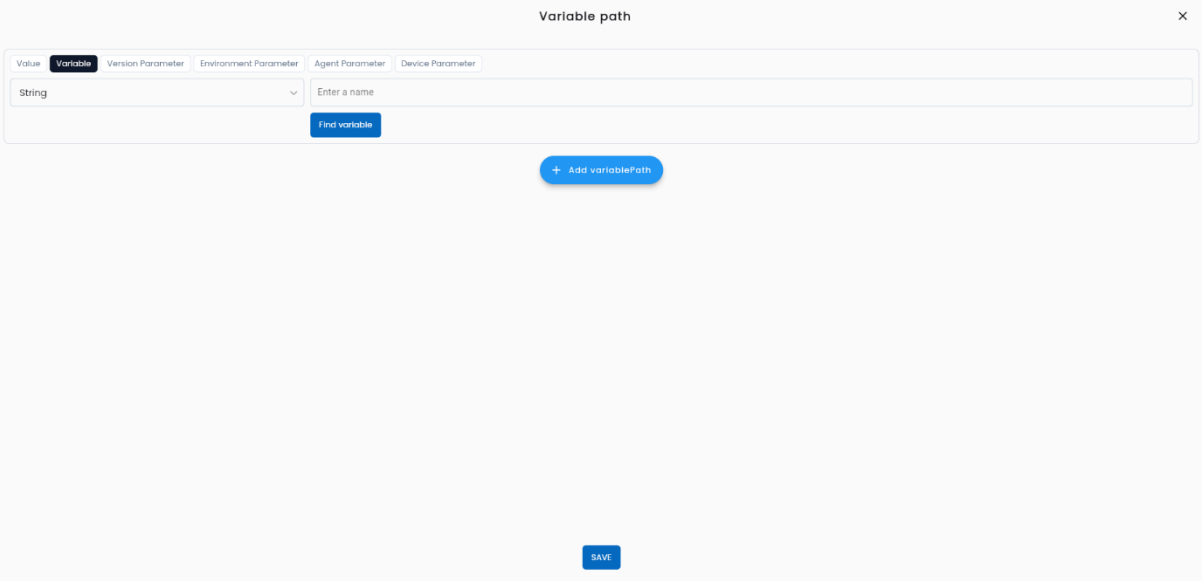
Click vào button "⋮" sau đó chọn "Field Config" hệ thống hiển thị màn hình "XML AND JSON OPTIONS".

Trong đó:

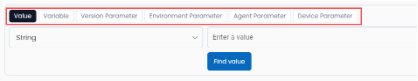
Trường thông tin	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Description	-	Text		Mô tả về tham số json
JSON	-	Switch button		Là button để kiểm soát tham số đang được khai báo có được sử dụng trong body request của api hay không.
Name	-	Text		Tên sẽ được sử dụng trong request để test api

❖ VariablePath Config

Click vào button “ ⋮ ” sau đó chọn “VariablePath Config” hệ thống hiển thị màn hình “Variable path”.



Trong đó

Trường thông tin	**Bắt buộc**	**Định dạng**	**Ràng buộc**	**Mô tả**
Type 	Có			Các kiểu giá trị của tham số Header Có thể chọn 1 trong các tab phù hợp với mục đích sử dụng
Data type	Có	Combobox		Kiểu dữ liệu
Name	Có	Text	Chỉ hiển thị khi type khác "Value" Giá trị phải bắt đầu bằng 1 chữ cái,	Tên của tham số đã được khai báo trong project hoặc tên của biến (Variable)

Trường thông tin	Bắt buộc	Định dạng	Ràng buộc	Mô tả
			bắt buộc bao gồm chữ số và không được sử dụng kí tự đặc biệt (ngoại trừ "_")	
Value	Có	Text	Chỉ hiển thị khi type = "Value"	Nhập giá trị trực tiếp của tham số Header
Find Value	-	Button	Chỉ hiển thị khi type = "Value"	Click vào Find Value hệ thống sẽ hiển thị danh sách các giá trị đã được định nghĩa trước đó của dự án
Find Variable	-	Button	Nếu Type là Variable	Click vào Find Variable hệ thống sẽ hiển thị danh sách các tham số đã được định nghĩa trước đó của dự án

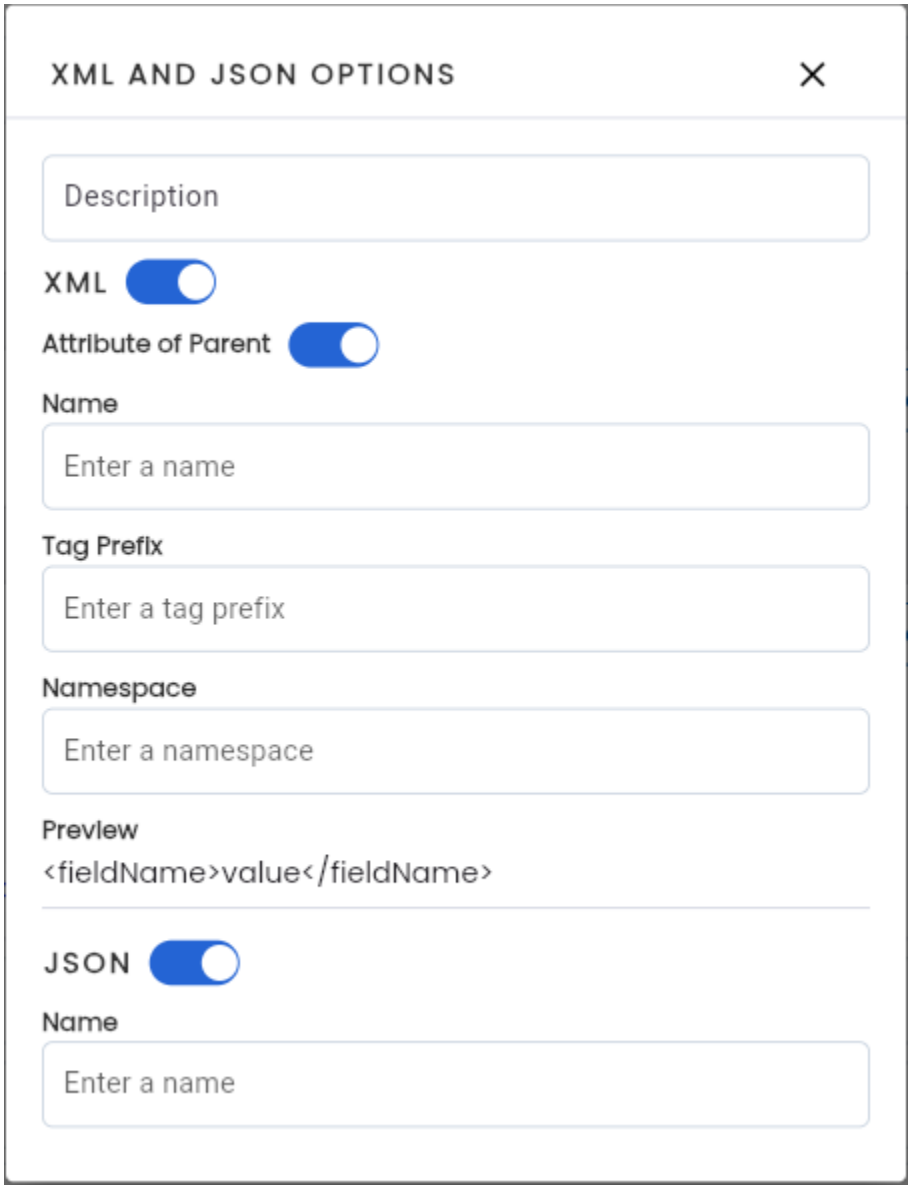
- Body request type: Application xml

Các chức năng tương tự request type là Application json thay vì sử dụng jsonName thì sẽ sử dụng xmlName

Ngoài ra

- Chức năng Field Config

Click vào button “ ⋮ ” sau đó chọn “Field Config” hệ thống hiển thị màn hình “XML AND JSON OPTIONS”.

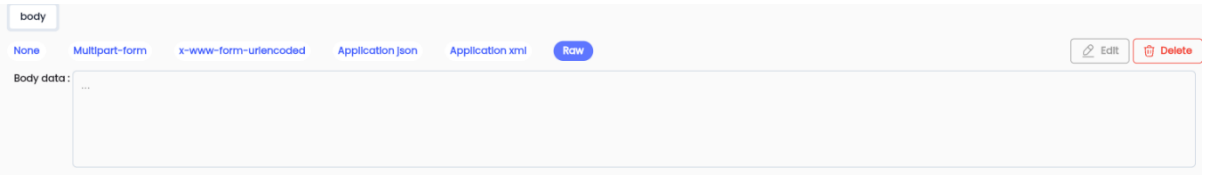


Trong đó

Trường thông tin	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Description	-	Text		Mô tả về tham số json
XML	-	Switch button		Là button để kiểm soát tham số đang được khai báo có được sử dụng trong body request của api hay không.
Attribute of Parent	Có	Switch button		Xác định liệu trường dữ liệu XML đang config có thuộc trong phần tử XML cha là Object hay Schema ở trên hay không?
Name	-	Text		Tên mà người dùng đặt cho tham số để xác định phần tử trong tài liệu XML.
Tag Prefix	-	Text		Tag prefix được sử dụng để phân biệt các phần tử XML với nhau.
Namespace		Text		Namespace được sử dụng để phân biệt các phần tử XML với nhau.

❖ Body request type: Raw

Request Type Raw sẽ xác định API Request có sử dụng nếu API sử dụng có kiểu là “json” hoặc “xml” nhưng sẽ không được xử lý trước khi tạo request api.



Trong đó

Trường thông tin	**Bắt buộc**	**Định dạng**	**Ràng buộc**	**Mô tả**
Body data	-	Textarea		Là dữ liệu body trong api request

1.1.11.1.5.4 Response API

API response là dữ liệu api phản hồi lại sau khi xử lý 1 request api.

Tạo Response cho API:

Bước 1: Trên tab Response, click button “Add”, hệ thống hiển thị màn hình “Add a response”

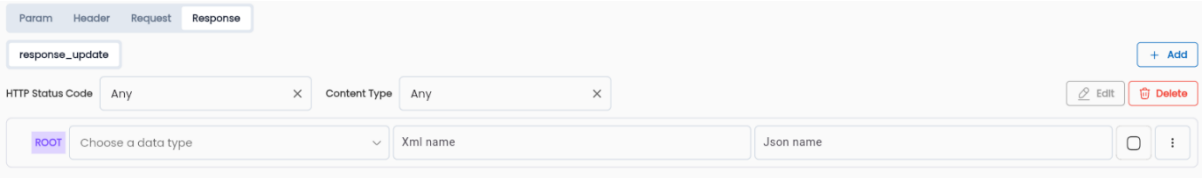
Bước 2: Nhập đầy đủ và hợp lệ các trường thông tin bắt buộc, trong đó:

Trường thông tin	**Bắt buộc**	**Định dạng**	**Ràng buộc**	**Mô tả**
Save	-	Button		Xác nhận tạo mới 1 Api response
Cancel	-	Button		Hủy bỏ việc tạo mới 1 Api response
Name	-	Text		Tên của Api Response lưu trong hệ thống
HTTP Status Code	Có	Combobox		Người dùng thực hiện lựa chọn các mã trạng thái http cụ thể hoặc có thể lựa chọn "Any" để hệ thống sẽ tự động xác định.
Content Type	Có	Combobox		Người dùng thực hiện lựa chọn các kiểu dữ liệ cụ thể hoặc có thể lựa chọn "Any"

				để hệ thống sẽ tự động xác định.

Bước 3: Click **"Save"**

Thêm mới thành công, thông tin response vừa tạo hiển thị trên màn hình.



Content Type bao gồm 4 kiểu dữ liệu:

+ Any

+ Application json

+ Text xml

+ Application xml

Các bước thao tác cấu hình dữ liệu thông tin dữ liệu trả về tương tự như việc cấu hình request có request type: Application json và Application xml.

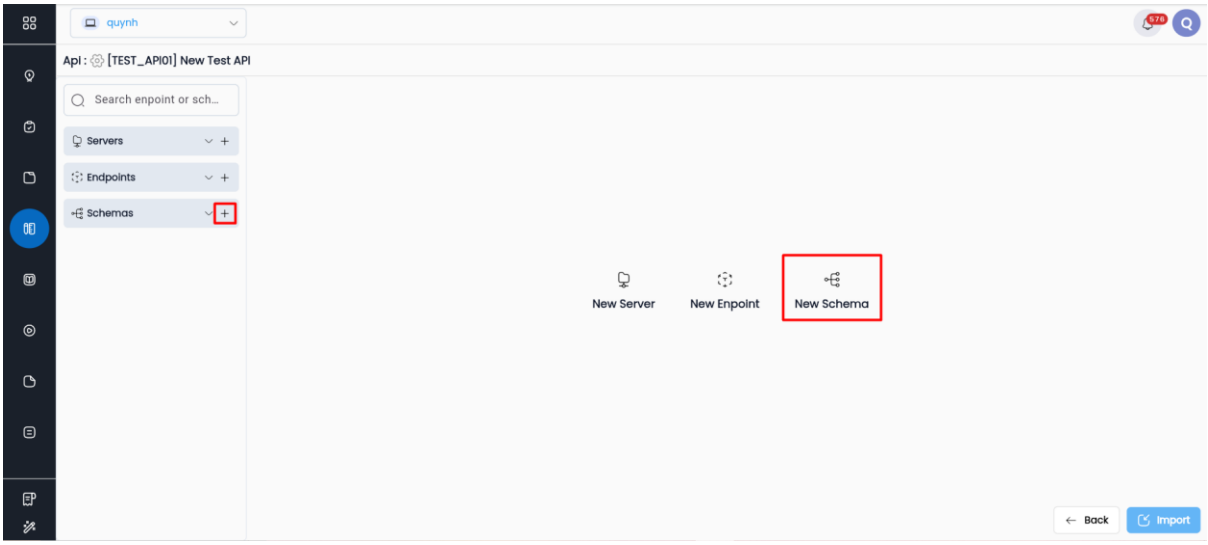
Tham chiếu tới mục Request API

1.1.11.1.5.5 Schemas

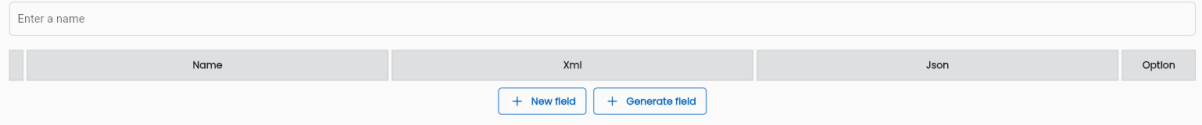
- Schemas được tạo để hỗ trợ cho việc quản lý các tham số có kiểu dữ liệu "Object" trong phần Endpoints

❖ Thêm mới Schemas

Bước 1: Trong màn hình Low-code , thực hiện click "New Schema" hoặc button " + "để thực hiện tạo mới 1 Schema



Bước 2: Hệ thống hiển thị màn hình thêm mới schema, người dùng tiến hành nhập các thông tin cần thiết



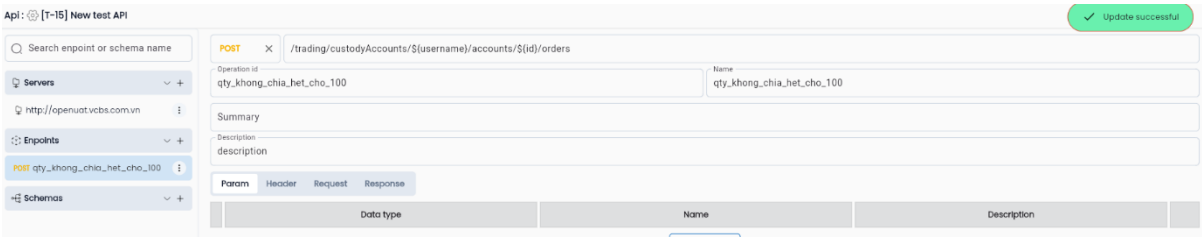
Trong đó :

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
New Field	-	Button		Click button để thêm 1 trường dữ liệu trong 1 schema
Generate field	-	Button		Là tính năng hỗ trợ cấu hình schemas từ một đoạn json
Enter a name	Có	Text		Tên của Schema

1.1.11.1.5.6 Chỉnh sửa Endpoints

Bước 1: Tại màn hình Low-code của test API, người dùng thực hiện chọn dòng dữ liệu Endpoint muốn thay đổi.

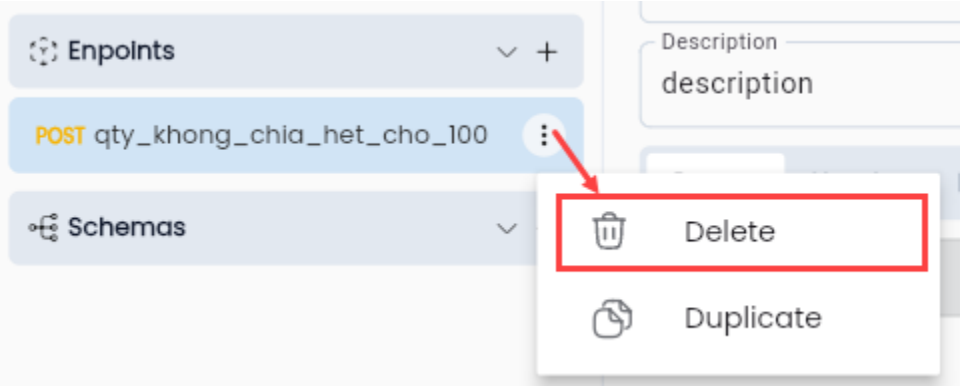
Bước 2: Chỉnh sửa thông tin cần thay đổi và click "Save"



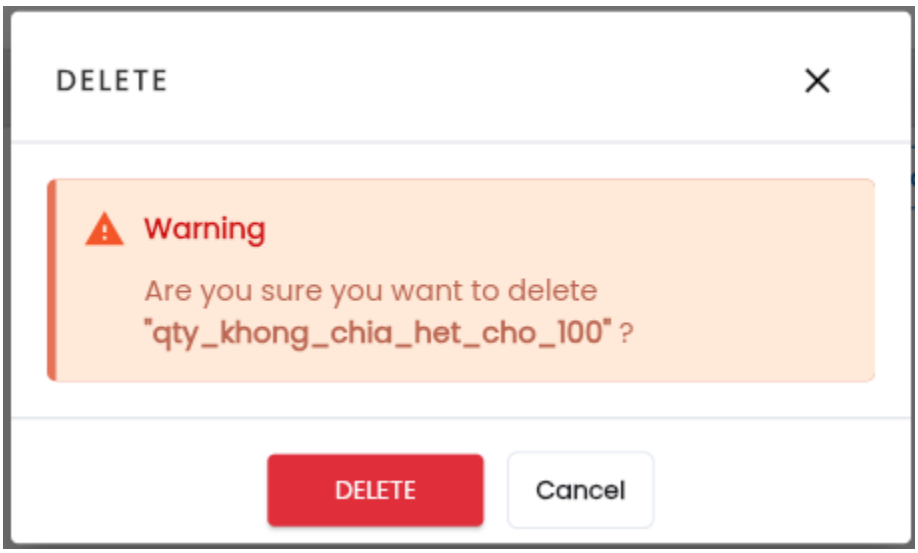
Hệ thống hiển thị thông báo "Update successful" và hiển thị Endpoint đã chỉnh sửa trong danh sách Endpoints.

1.1.11.1.5.7 Delete Endpoint

Bước 1: Tại màn hình Low-code của test API, người dùng thực hiện click button "⋮" trên dòng dữ liệu Endpoint muốn xóa và chọn Delete.



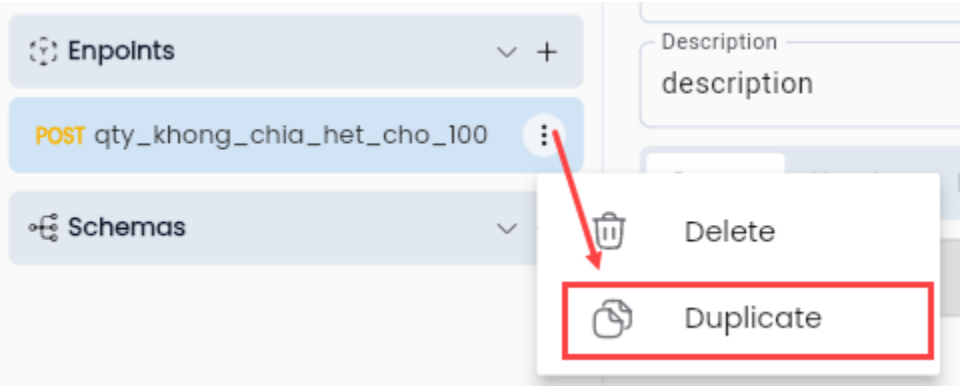
Hệ thống hiển thị popup xác nhận xoá



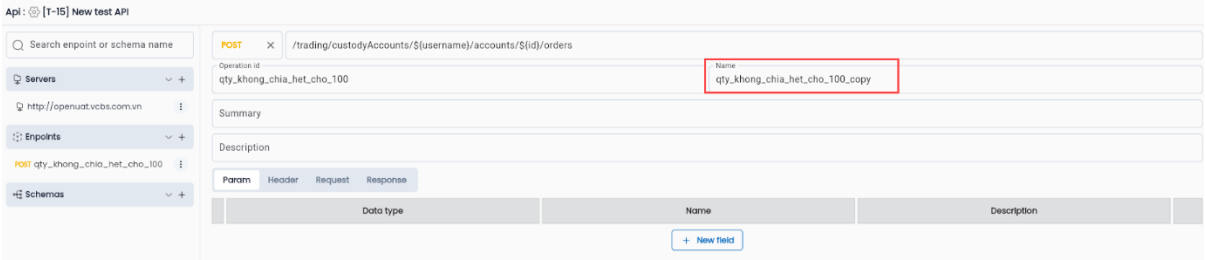
Bước 2: Nhấn "**DELETE**", hệ thống thông báo **"Delete successful"**

1.1.11.1.5.8 Duplicate Endpoint

Bước 1: Tại màn hình Low-code của test API, người dùng thực hiện click button "⋮" trên dòng dữ liệu Endpoint và chọn Duplicate.



Hệ thống hiển thị màn hình Endpoint đã được duplicate thành công với đầy đủ thông tin Url, Param, Header, Request, Response cùng với Name = name của Endpoint được duplicate + "_copy"

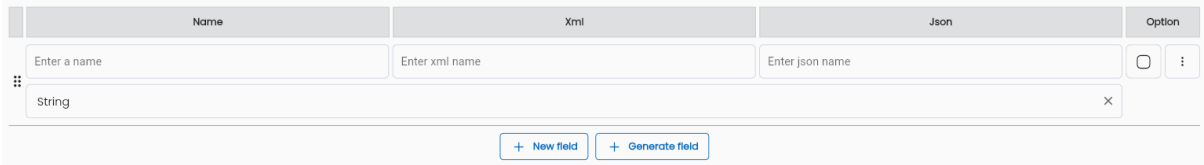


Bước 2: Click "Save", hệ thống thông báo Update successful, danh sách Endpoints được cập nhật trong đó có Endpoint đã được Dupicate thành công.

1.1.11.1.6 Cấu hình API

1.1.11.1.6.1 Sử dụng New field thêm thủ công tham số cho Endpoint

Bước 1: Click New field



Bước 2: Nhập thông tin cho tham số, trong đó:

Trường thông tin	**Bắt buộc**	**Định dạng**	**Ràng buộc**	**Mô tả**
Name	Có	Text		Tên của trường dữ liệu trong schema
Data Type	Có	Combobox		Kiểu dữ liệu của 1 trường dữ liệu. Hệ thống khi khởi tạo parameter mới mặc định để là "String". Ngoài các kiểu dữ liệu thông thường, có thể tạo 1 trường dữ liệu kiểu Schema trong 1 Schema.

Trường thông tin	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Xml		Text		Tên sẽ được sử dụng trong request để test api nếu request là application xml.
Json		Text		Tên sẽ được sử dụng trong request để test api nếu request là application json.
Is list	-	Checkbox		Nếu trường thông tin có value là dạng list, thực hiện tick vào checkbox

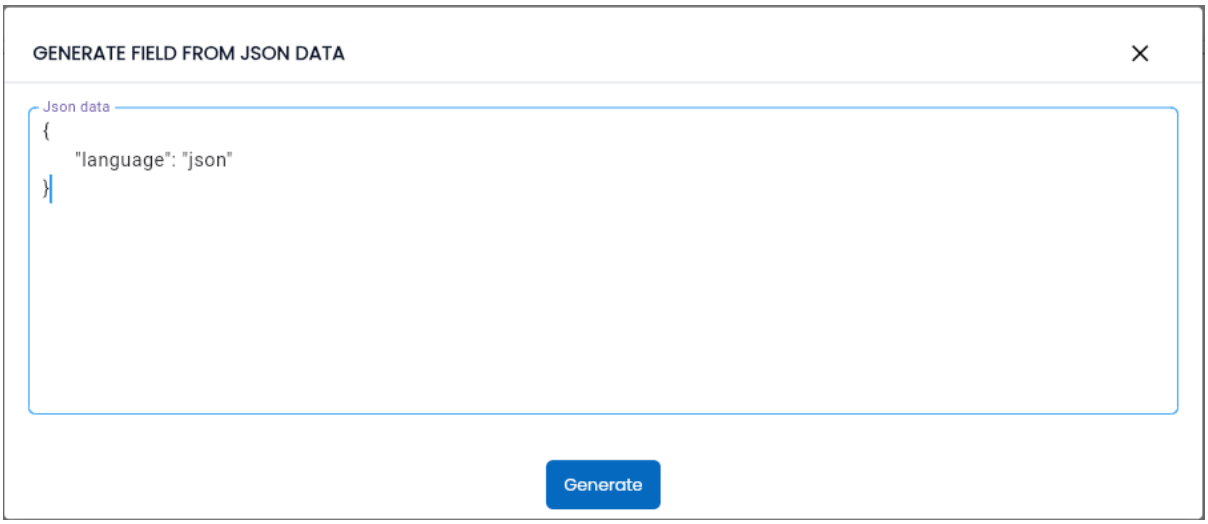
Bước 3: Thao tác tương tự với các trường thông tin còn lại và click **Save**

1.1.11.1.6.2 Sử dụng Generate field

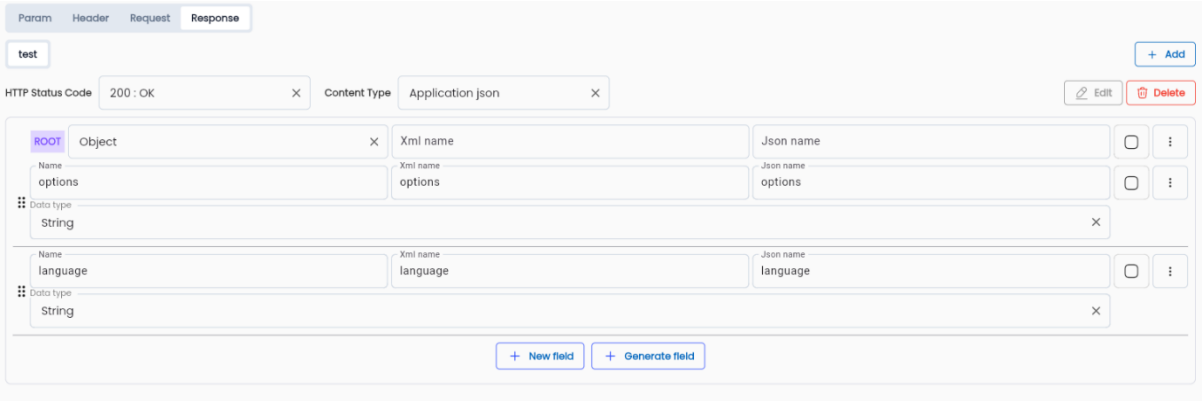
Generate field hỗ trợ chuyển đổi một đoạn Json có sẵn thành các tham số của request/ respone trong API

Bước 1: Click Generate field

Bước 2: Dán đoạn json cần thực hiện chuyển đổi vào vùng nhập liệu Json data



Bước 3: Click Generate

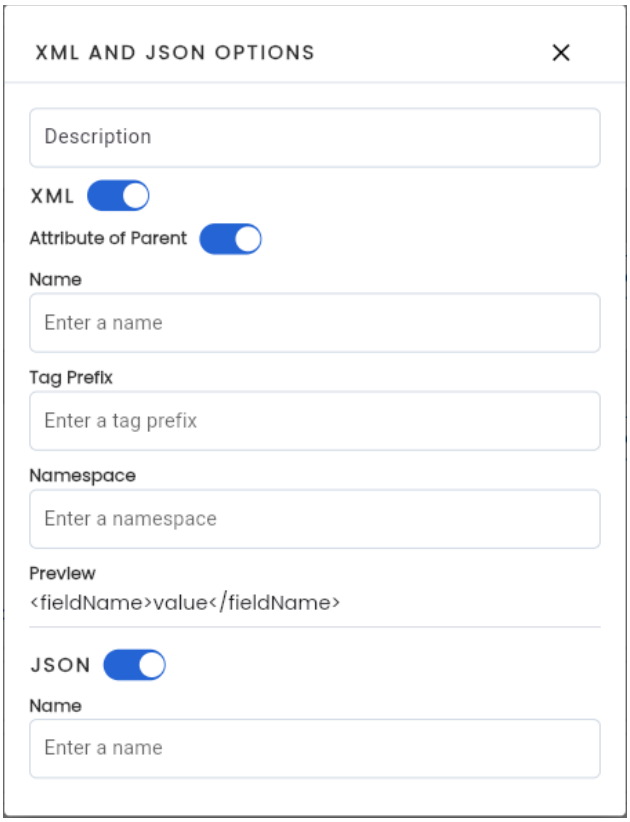


Các tham số có trong json đã được tạo tự động

Lưu ý: Với những json chứa tham số có định dạng là schema thì cần được tạo trước qua New field hoặc Generate field để có thể lựa chọn schema cho tham số cho đó

1.1.11.1.6.3 Field Config

Tương ứng với mỗi trường thông tin, click vào button “ ⋮ ” sau đó chọn “Field Config” hệ thống hiển thị màn hình “XML AND JSON OPTIONS”.



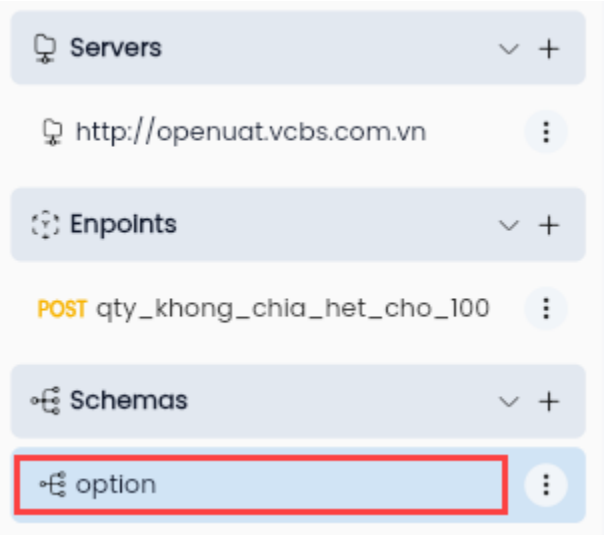
Trong đó:

Trường thông tin	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Description	-	Text		Mô tả về trường dữ liệu
XML Options				
XML	-	Switch button		Là button để kiểm soát tham số đang được khai báo có được sử dụng trong body request của api hay không.
Attribute of Parent	-	Switch button		
Name	-	Text		Tên mà người dùng đặt cho tham số để xác định phần tử trong tài liệu XML.
Tag Prefix	-	Text		Tag prefix được sử dụng để phân biệt các phần tử XML với nhau.
Namespace		Text		Namespace được sử dụng để phân biệt các phần tử XML với nhau.

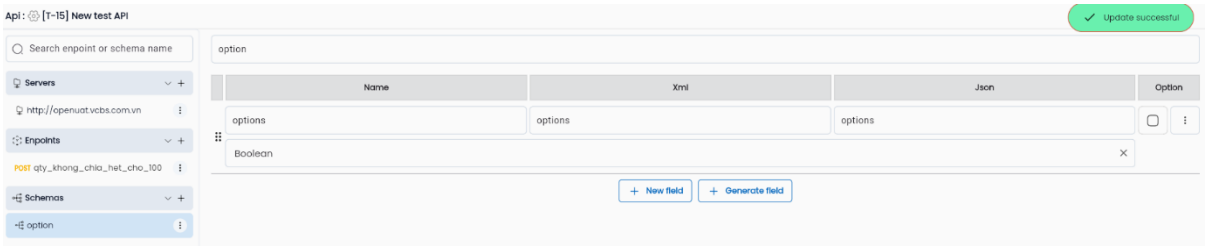
Trường thông tin	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Json Options				
JSON	-	Switch button		Là button để kiểm soát tham số đang được khai báo có được sử dụng trong body request của api hay không.
Name	-	Text		Tên sẽ được sử dụng trong request để test api

❖ Chỉnh sửa Schemas

Bước 1: Tại màn hình Low-code của test API, người dùng thực hiện chọn dòng dữ liệu Schemas muốn thay đổi.



Bước 2: Chỉnh sửa thông tin cần thay đổi và click “Save”



Hệ thống hiển thị thông báo “Update successful” và hiển thị Schema đã chỉnh sửa trong danh sách Schemas.

❖ Delete Schema

Thực hiện tương tự như Endpoints. Tham chiếu tới mục Endpoints

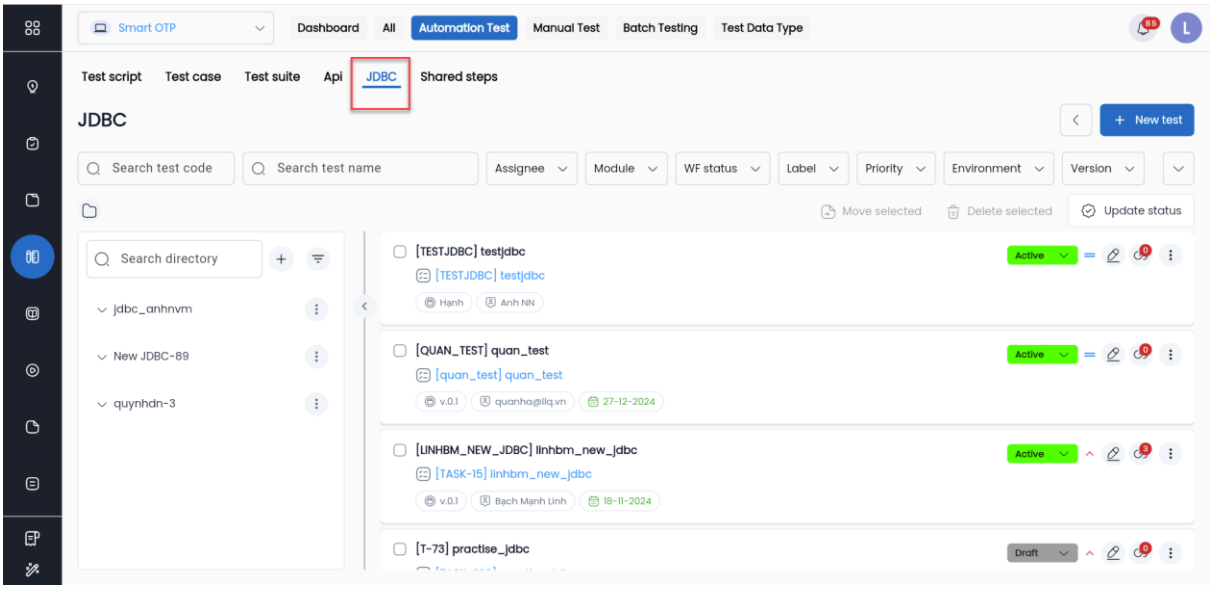
❖ Duplicate Endpoint

Thực hiện tương tự như Endpoints. Tham chiếu tới mục Endpoints

1.1.12 Test JDBC

1.1.12.1 Test JDBC là gì?

- Test JDBC là một dạng kịch bản kiểm thử tương tác với cơ sở dữ liệu bao gồm hành động lấy dữ liệu và chỉnh sửa dữ liệu.

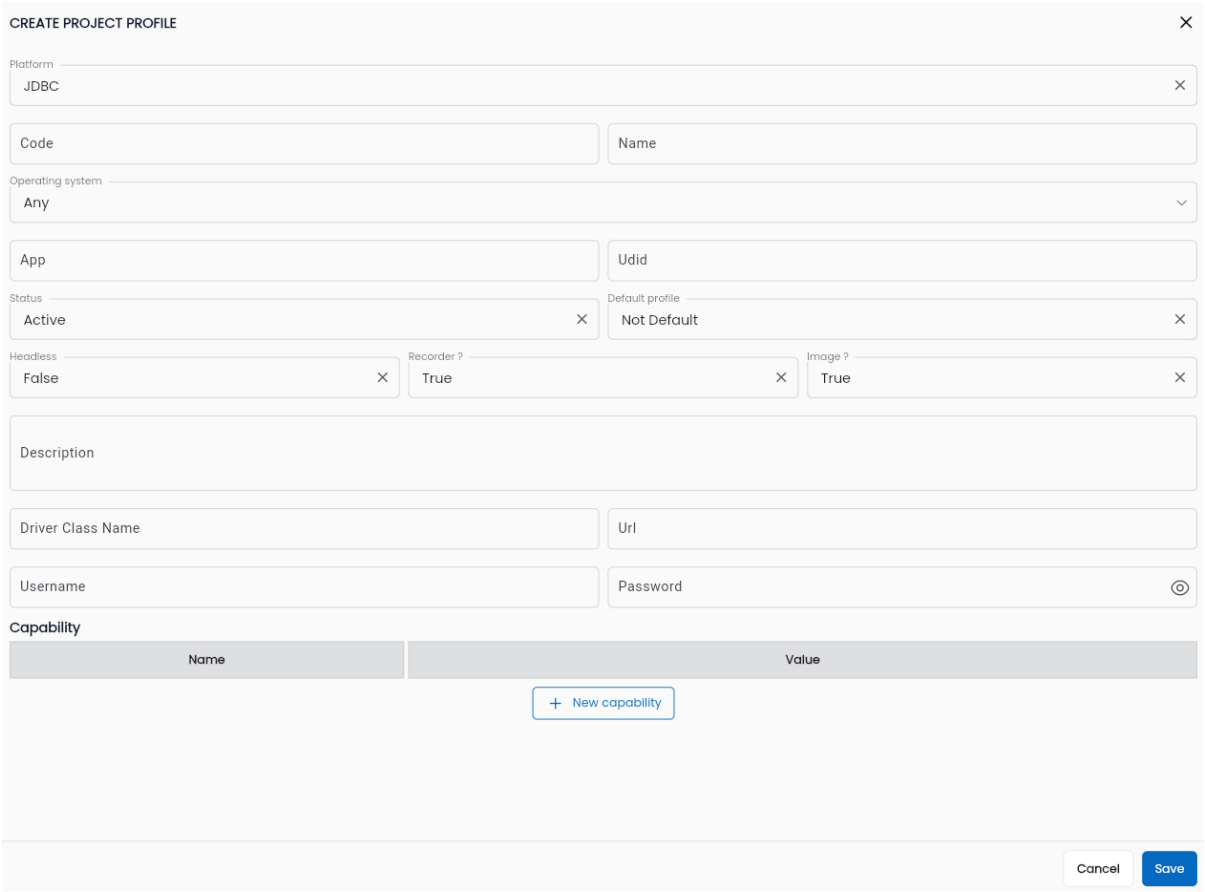


1.1.12.2 Tạo mới một Profile JDBC

Profile JDBC được sử dụng để lưu trữ thông tin của cơ sở dữ liệu mà Test JDBC sẽ tương tác.

Các bước để truy cập vào màn hình Profile tham chiếu tới mục Profile

Bước 1: Tại màn hình "CREATE PROJECT PROFILE" thực hiện chọn Platform JDBC



Bước 2: Tiến hành nhập các thông tin được mô tả tại mục Profile. Ngoài ra các thông tin đặc biệt của platform JDBC:

Tên trường	**Bắt buộc**	**Định dạng**	**Ràng buộc**	**Mô tả**
Driver Class Name	Có	Text		Thông tin driver định danh của từng loại cơ sở dữ liệu
Url	Có	Text		Url để kết nối tới cơ sở dữ liệu

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Username	Có	Text		Thông tin tài khoản của cơ sở dữ liệu
Password	Có	Text		Thông tin mật khẩu của cơ sở dữ liệu

Lưu ý:

+ Hiện tại hệ thống đang hỗ trợ 2 driver test jdbc:

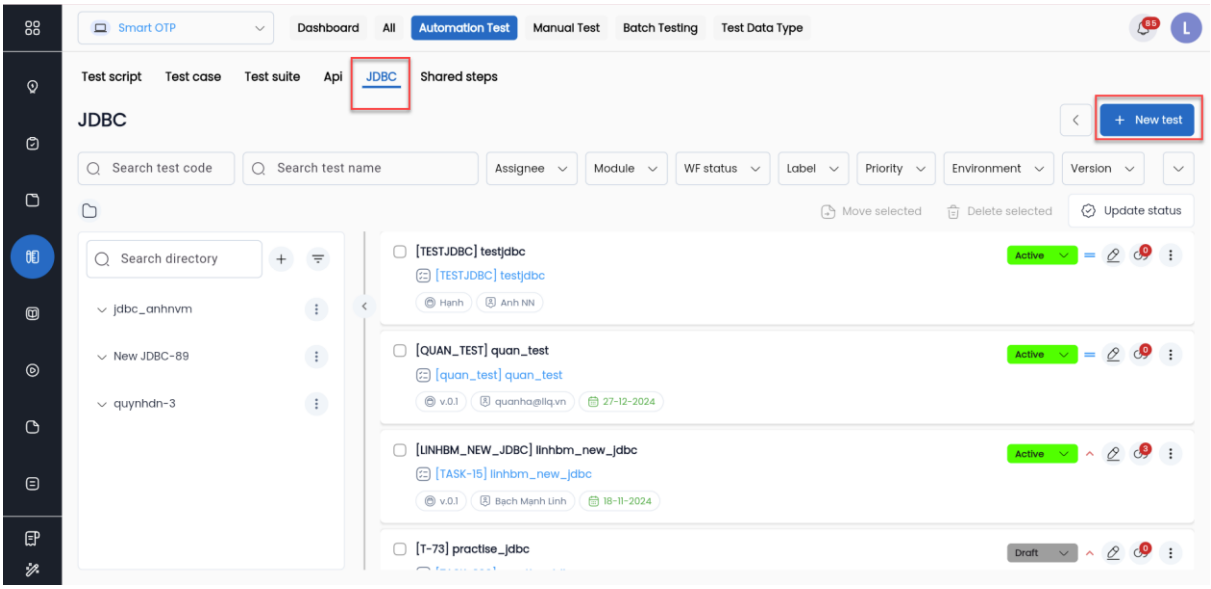
- Postgresql: org.postgresql.Driver
- Oracle: oracle.jdbc.OracleDriver

+ Định dạng URL:

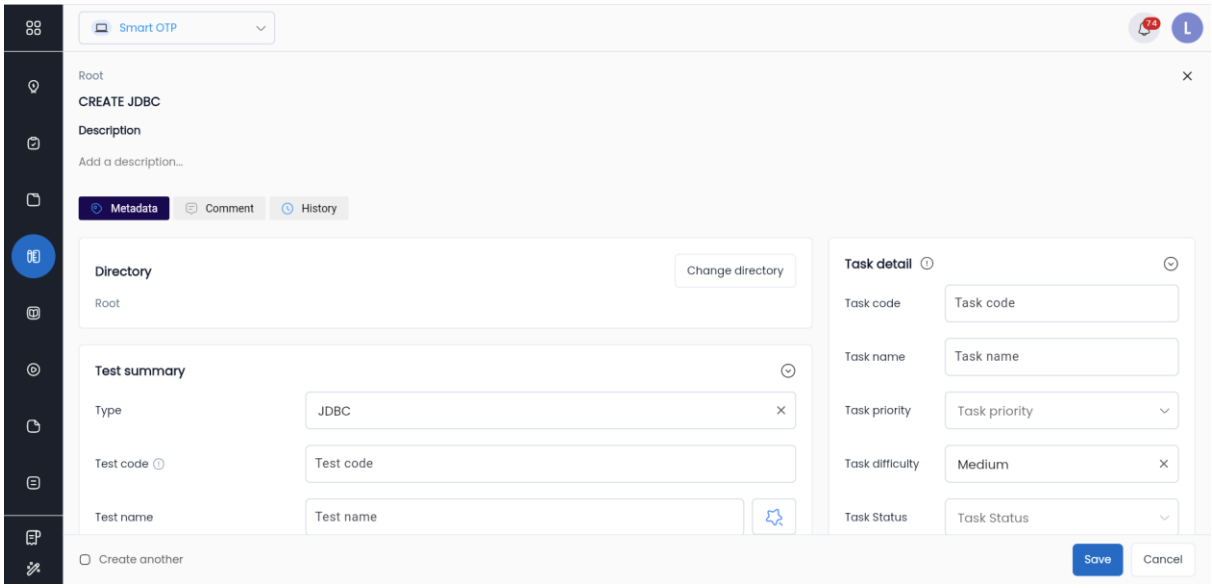
- Postgresql: **jdbc:postgresql://[host]:[tcpPort]/[databaseName]**
- Oracle: **jdbc:oracle:thin:@//[host]:[tcpPort]/[service_name]**
- Oralce: **jdbc:oracle:thin:@//[host]:[tcpPort]:[sid]**

1.1.12.3 Tạo mới một test jdbc

Bước 1: Trên màn hình tab JDBC, nhấn "**New test**", hệ thống hiển thị màn hình metadata của JDBC



Bước 2: Nhập đầy đủ và hợp lệ cho các trường thông tin:



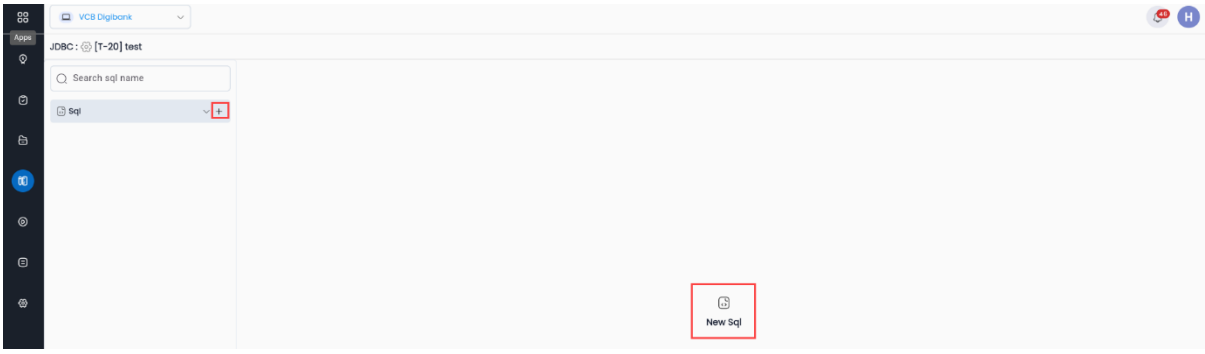
Màn hình Metadata của JDBC

Trong đó: Phần thông tin chung của 1 Test JDBC, tương tự Test script, tham chiếu mục [Tạo mới một test script]

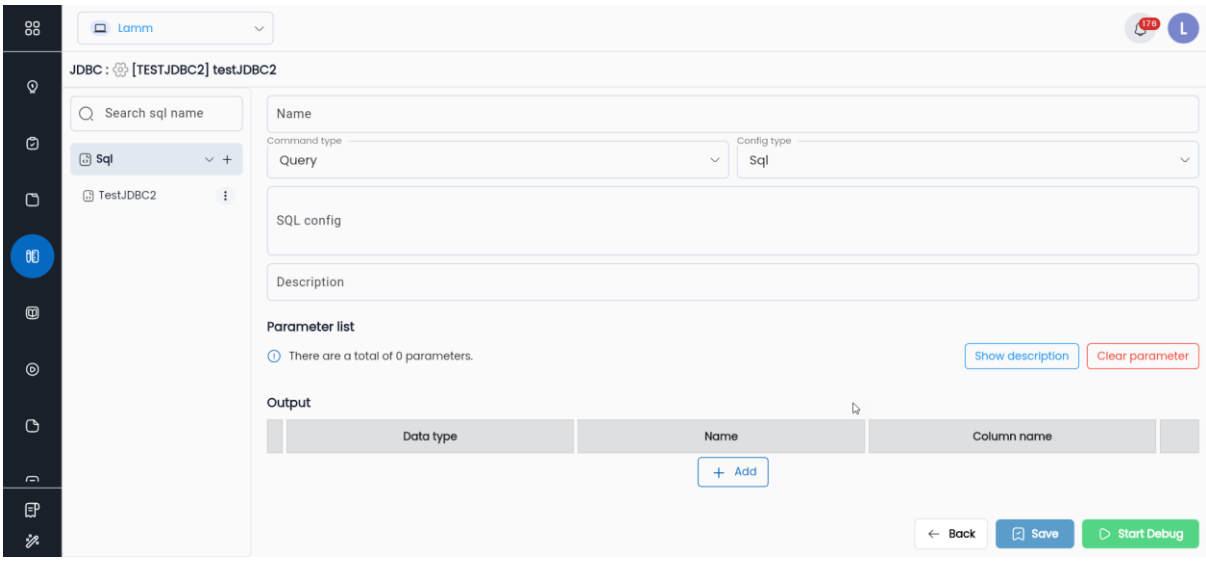
Bước 3: Nhấn "**Save",** hệ thống thông báo **"Create successful"** và hiển thị màn hình cấu hình.

Tạo mới SQL:

Bước 1: Nhấn vào icon "+", hoặc nhấn **New Sql** giữa màn hình.



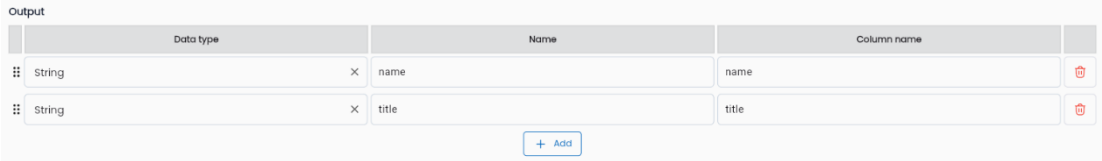
Hệ thống hiện thị màn hình config



Mô tả các trường thông tin

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Section Config Lưu trữ cấu hình tương tác với cơ sở dữ liệu của test jdbc				
Command Type	Có	Combobox		Hệ thống hiển thị danh sách kiểu tương tác khi sử dụng JDBC: + Query: Lấy dữ liệu + Execute Update: Chỉnh sửa dữ liệu

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				Người dùng được chọn 1 giá trị.
Config Type	Có	Combobox	Query Builder hiện tại chỉ hỗ trợ thực hiện với kiểu tưởng tác Query để lấy dữ liệu.	Hệ thống hiển thị danh sách kiểu thao tác khi sử dụng trên hệ thống: + Sql: Sử dụng câu sql trực tiếp + Query Builder: Sử dụng hệ thống hỗ trợ cấu hình theo cấu trúc câu SQL.
Sql Config	Có	Text	Sql Type = "Sql"	Chứa câu lệnh sql để tương tác với cơ sở dữ liệu
Edit query builder	Có	Combobox	Chỉ hiển thị khi Sql Type = "Query Builder"	Click button hệ thống sẽ hiển thị màn hình Edit query builder hỗ trợ xây dựng 1 câu lệnh sql để tương tác với cơ sở dữ liệu
Show description	-	Button		Cho phép người dùng ẩn/hiện trường dữ liệu "Description" của một Parameter trong test jdbc.
+ New parameter	-	Button		Click button cho phép người dùng tạo

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				một Parameter mới cho test jdbc.
Clear parameter	-	Button		Click button cho phép người dùng xóa tất cả Parameter đã khai báo cho test jdbc.
Output Lưu trữ các output chứa giá trị là đầu ra của 1 test jdbc Chỉ xuất hiện khi Command Type = “Query”  				
Add	-	Button		Cho phép người dùng tạo một trường dữ liệu output cho test jdbc.
Data Type	Có	Combobox		Kiểu dữ liệu của 1 trường dữ liệu output. Hệ thống khi khởi tạo trường dữ liệu output mới mặc định để là “String”.
Name	Có	Text	Tên phải bắt đầu bằng chữ cái, bắt buộc bao gồm cả số	Tên của 1 trường dữ liệu output trên hệ thống.

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
			và không chứa kí tự đặc biệt ngoài "_"	
Column name	Có	Text		Tên của 1 trường dữ liệu trong output mà jdbc trả về cho hệ thống.

Lưu ý: Trường dữ liệu "name" và "columnName" là khác nhau:

Name là tên của tham số trên hệ thống chỉ được sử dụng để định danh cho biến output trên hệ thống.

Column name là tên nằm trong kết quả trả về của 1 lần query JDBC test.

Bước 2: Nhấn "**Save**" để lưu lại cấu hình, hệ thống hiển thị thông báo **"Update successful"**

1.1.12.3.1 SQL

Cấu trúc 1 câu lệnh sql cơ bản bao gồm:

- Đối với Command Type là "Query"
  SELECT: Gồm các cột mà người dùng muốn truy vấn
  - FROM: Chỉ định bảng dữ liệu người dùng muốn truy vấn
  - WHERE: Điều kiện truy vấn (Tùy chọn)
  - GROUP BY: Nhóm dữ liệu dựa trên 1 cột hoặc nhiều cột (Tùy chọn)
  - ORDER BY: Sắp xếp kết quả (Tùy chọn)
  - LIMIT: Giới hạn số lượng kết quả trả về (Tùy chọn)
  - OFFSET: Số lượng dòng dữ liệu sẽ được bỏ qua trước khi trả về kết quả (Tùy chọn)
- Đối với Command Type là "Execute Update"
  UPDATE: Gồm các cột mà người dùng muốn cập nhật dữ liệu
  SET: Xác định cột và giá trị thay đổi dữ liệu
  WHERE: Điều kiện lọc các bản ghi cần thay đổi (Tùy chọn)

***Lưu ý*:**

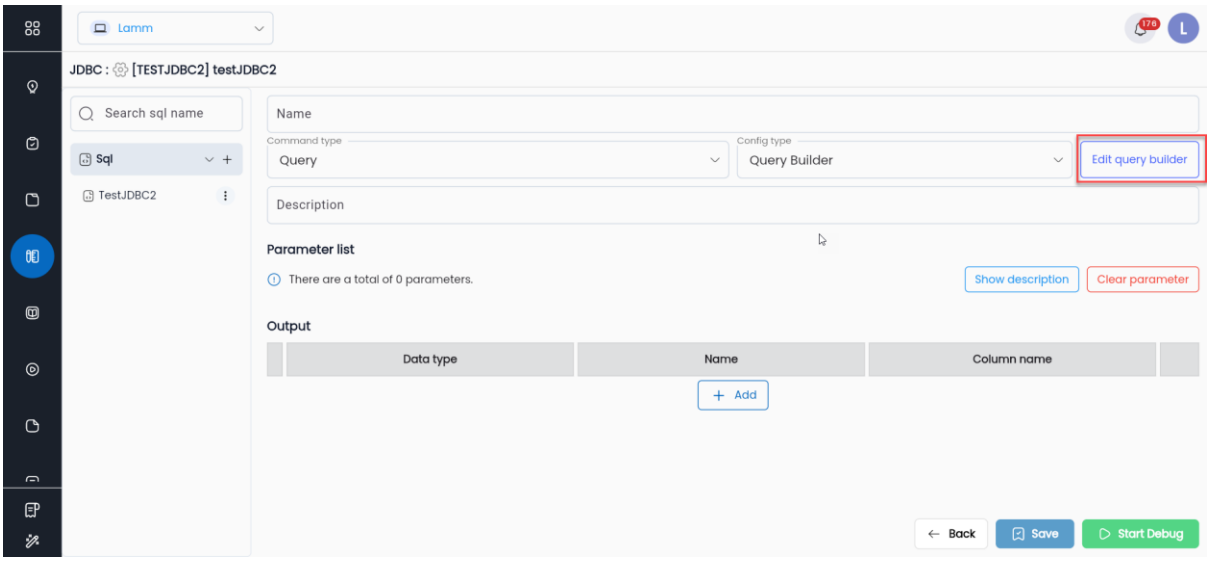
+ Yêu cầu đối với 1 câu SQL là phải đủ các thành phần bắt buộc, các thành phần tùy chọn có thể có hoặc không phụ thuộc vào mục đích của người dùng.
+ Hệ thống sẽ không kiểm tra câu lệnh SQL của người dùng có hợp lệ hay không, nếu kịch bản test JDBC bị lỗi, có thể là do câu lệnh SQL có sai sót.

1.1.12.3.2 Query Builder

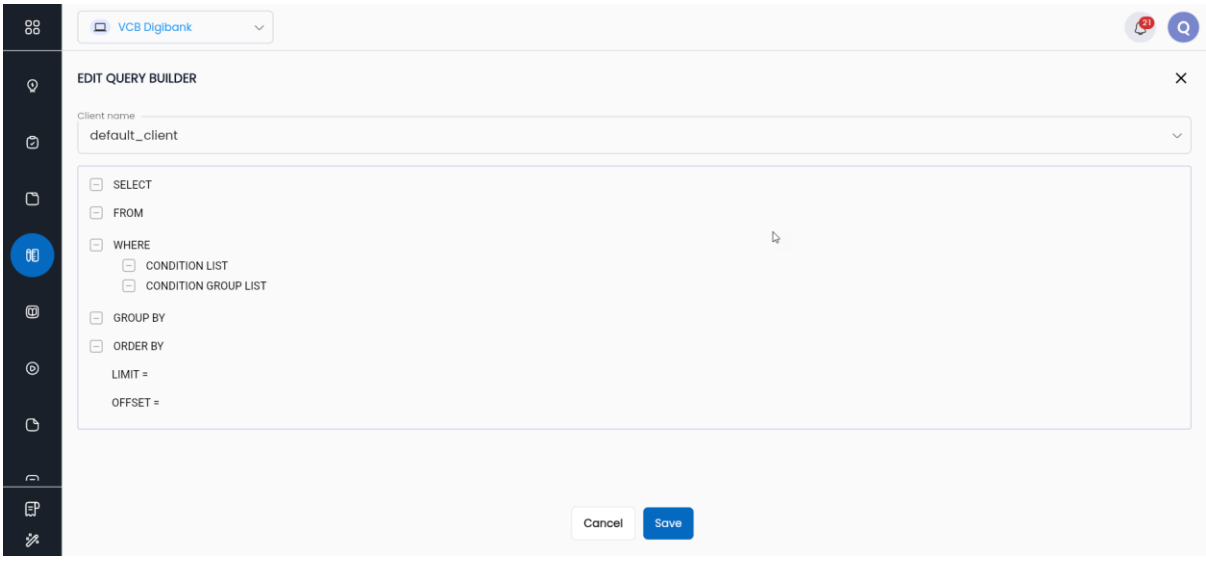
Query Builder hiện tại chỉ hỗ trợ được Type Query

Edit query Builder

Bước 1: Nhấn Người dùng chọn giá trị cho **Config type** là **Query builder,** sau đó nhấn **Edit query builder**



Hệ thống hiển thị màn hình **EDIT QUERY BUILDER**



Mô tả các trường thông tin:

Tên trường	**Bắt buộc**	**Định dạng**	**Ràng buộc**	**Mô tả**
Client name	Có	Combobox		Là client ứng với profile của JDBC
Add	-	Button		Thêm cấu hình
Edit	-	Button		Chỉnh sửa cấu hình
Clear	-	Button		Xóa toàn bộ cấu hình bên trong tính từ dòng của button "Clear"
Delete	-	Button		Xóa 1 cấu hình

Bước 2: Tiến hành cấu hình 1 câu query hoàn chỉnh.

Lưu ý: Cấu hình query builder nên bắt đầu từ mục FROM để xác định được các bảng dữ liệu sẽ được sử dụng.

1.1.12.3.2.1 SELECT

Bước 1: Nhấn **Add** tại dòng SELECT để tạo thông tin đầu ra của jdbc, hệ thống tạo mới 1 dòng thông tin trong list "SELECT"

Bước 2: Nhấn **Edit** tại dòng muốn chỉnh sửa, hệ thống hiển thị màn hình thông tin

Trong đó: Thông tin đầu ra phụ thuộc vào "Select column type"

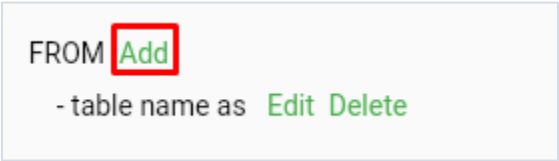
Select column type bao gồm

+ Min, max, avg, sum, count, countOne: Các hàm tổng hợp (aggregate functions) là các phép tính trên một tập hợp các giá trị và trả về một giá trị duy nhất.
+ TruncDate: Hàm xử lý Date
+ Case: Biểu thức điều kiện là Case
+ Column: Hàm lấy dữ liệu theo trường dữ liệu
+ QuerySelect: Hàm lấy dữ liệu từ 1 block subquery bên trong SELECT

Các thông tin đầu ra được gắn vào Column alias và được map tới các Output được khai báo tại tab Config.

1.1.12.3.2.2 FROM

Bước 1: Nhấn **Add** tại dòng FROM để tạo thông tin bảng sử dụng, hệ thống tạo mới 1 dòng thông tin table name as trong list "From"



Bước 2: Nhấn **Edit** tại table muốn chỉnh sửa, hệ thống hiển thị màn hình thông tin của 1 Table

Trong đó: Type query là kiểu query giúp xác định cách thức sử dụng bảng trong jdbc, bao gồm 3 loại:

- Query Table: Sử dụng 1 bảng độc lập
- Query Join: Sử dụng từ nhiều bảng
- Query Select: Sử dụng dữ liệu từ 1 block subquery

+ Query Table

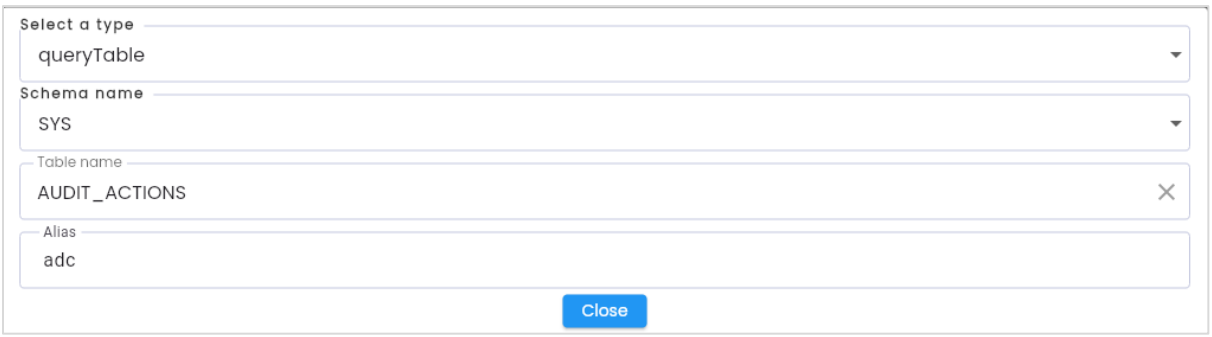
Màn hình tùy chỉnh queryTable

Trong đó

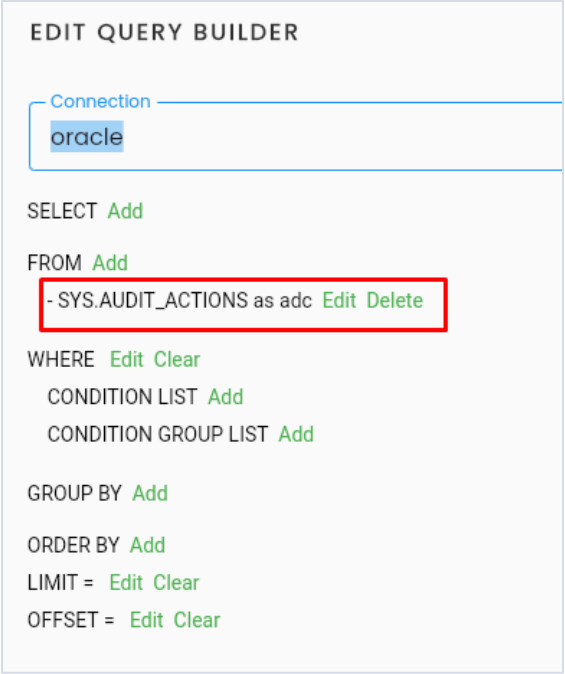
Tên trường	**Bắt buộc**	**Định dạng**	**Ràng buộc**	**Mô tả**
Schema name	Có	Combobox		Người dùng chọn Schema có chứa Table muốn sử dụng trong đó
Table name	Có	Combobox		Người dùng chọn Table name muốn sử dụng
Alias	Có	Text		Tên tạm thời của bảng được sử dụng trong cái mục khác

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				giúp rút gọn tên bảng hơn. Hệ thống sẽ tự sinh Alias từ Table Name, người dùng có thể chỉnh sửa

VD: Thực hiện cấu hình câu query “Select … from SYS.AUDIT_ACTIONS as adc”

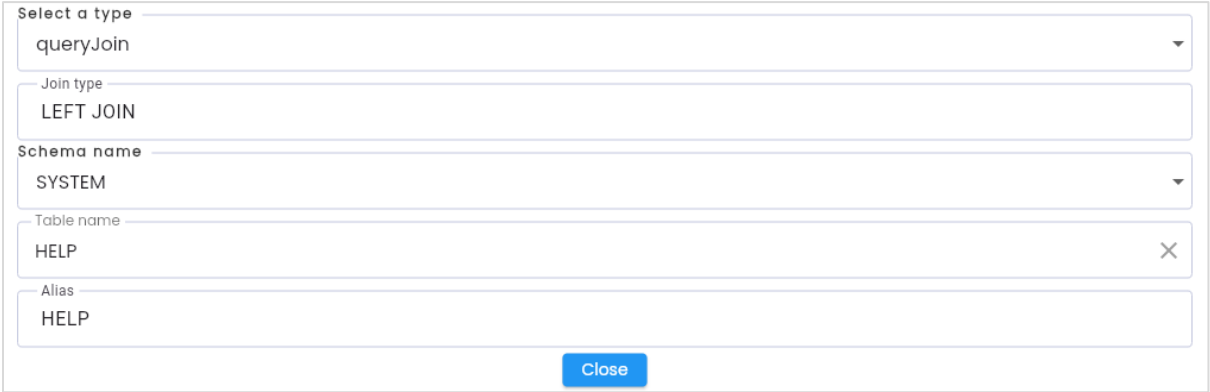


Sau khi lựa chọn các thành phần, người dùng ấn Close, câu query vừa cấu hình được thay đổi trong FROM



+ Query Join

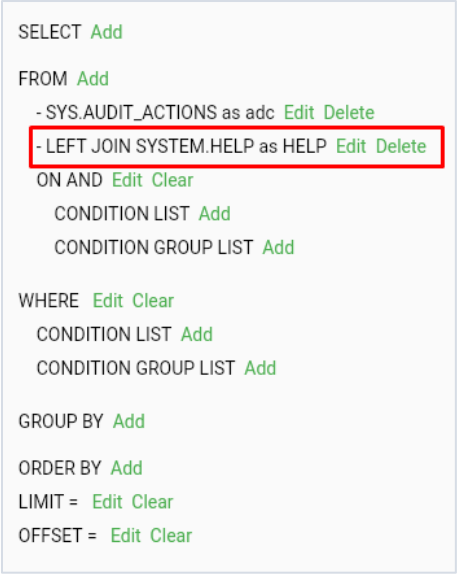
Màn hình tùy chỉnh queryJoin



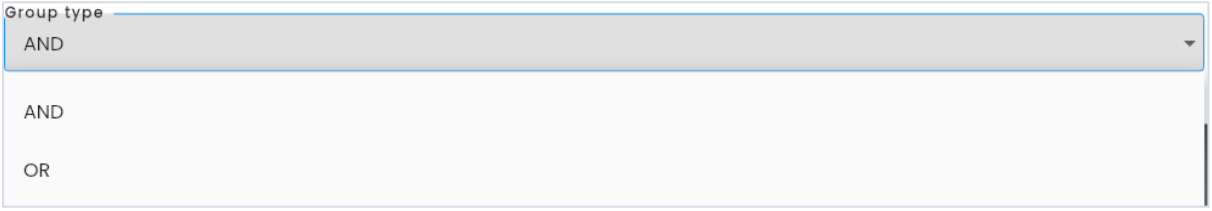
Query có kết nối ít nhất 2 bảng lại với nhau trong đó Join type bao gồm tất cả các kiểu join được cho phép trong driver của Connection đã chọn như “LEFT JOIN”, “RIGHT JOIN”, “INNER JOIN”, …

Trong đó, các trường thông tin của bảng được join thao tác chọn thông tin tương tự một bảng độc lập tại query table.

Sau khi click “Close”, hệ thống cập nhật câu query trên danh sách FROM



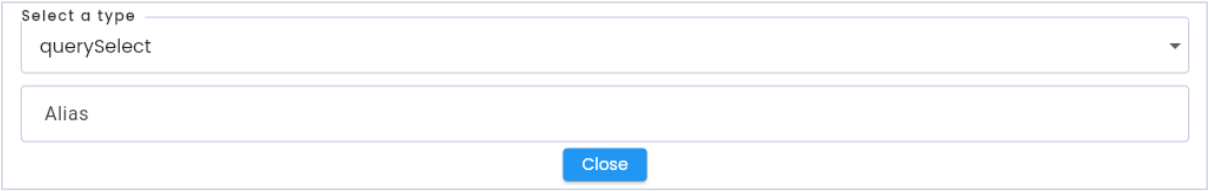
Cấu hình kết nối giữa 2 bảng, Button “Edit” tại dòng ON được sử dụng để kiểm soát điều kiện kết nối giữa 2 bảng “AND” hoặc “OR”.



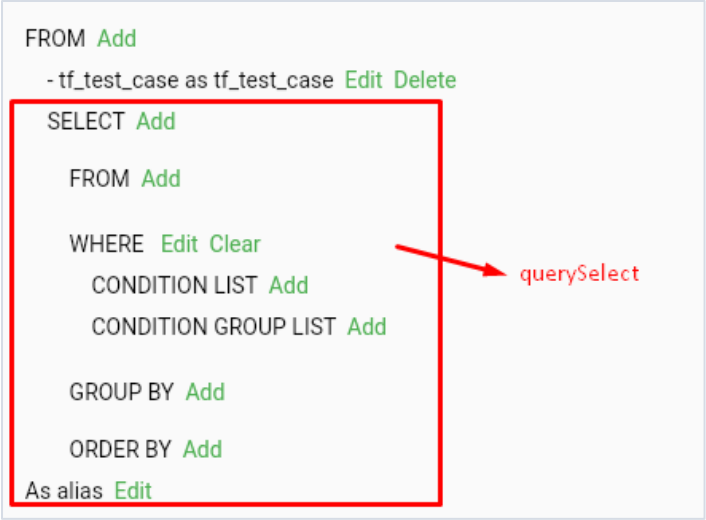
Các bước cấu hình cho phần CONDITION xử lý tương tự phần CONDITION trong mục “WHERE”.

+ Query Select

Màn hình tùy chỉnh querySelect



Khi click “**Close**”, hệ thống tự động sinh ra 1 block cấu hình subquery mới tại dòng table được chọn chỉnh sửa.



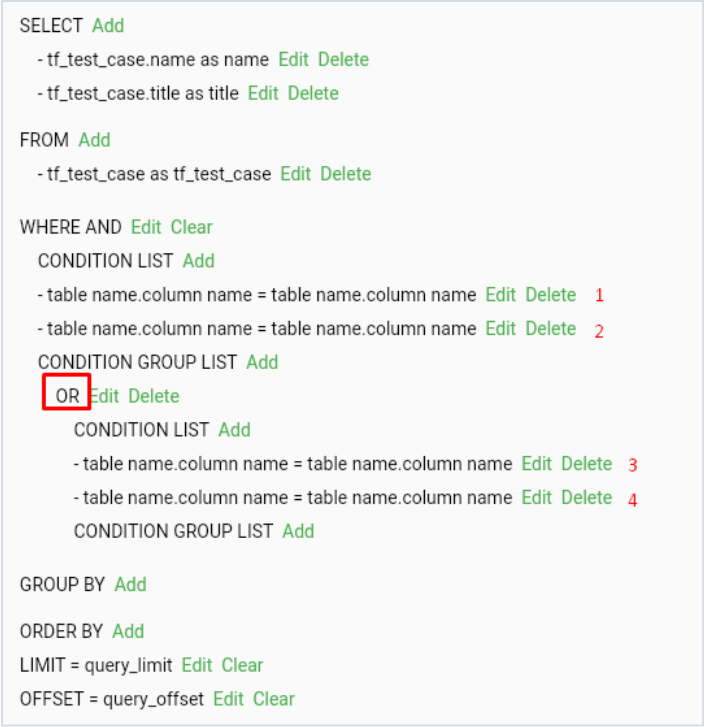
Thao tác thực hiện và các trường thông tin tương tự như cấu hình Builder cho block query ban đầu.

Dữ liệu trả về từ block query sẽ được gắn vào 1 table tạm và có thể sử dụng trong toàn bộ query builder.

1.1.12.3.2.3 WHERE

Trong mỗi câu điều kiện gồm 2 thành phần:

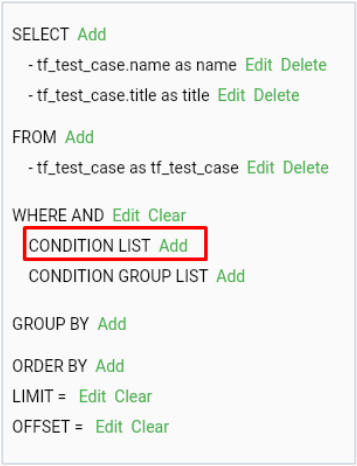
+ Condition List: Chứa các điều kiện lọc dữ liệu.
+ Condition Group List chứa 1 câu điều kiện cùng cấp với Condition List kết nối bằng điều kiện “AND” hoặc “OR”.



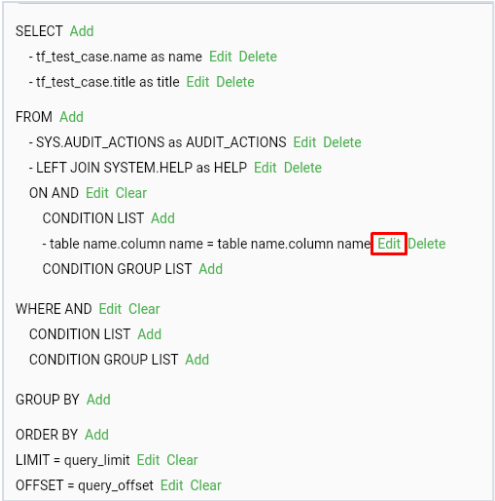
Tại hình minh họa có ý nghĩa tìm các bản ghi có điều kiện (1) và (2) đúng hoặc là (3) và (4) đúng.

EDIT CONDITON

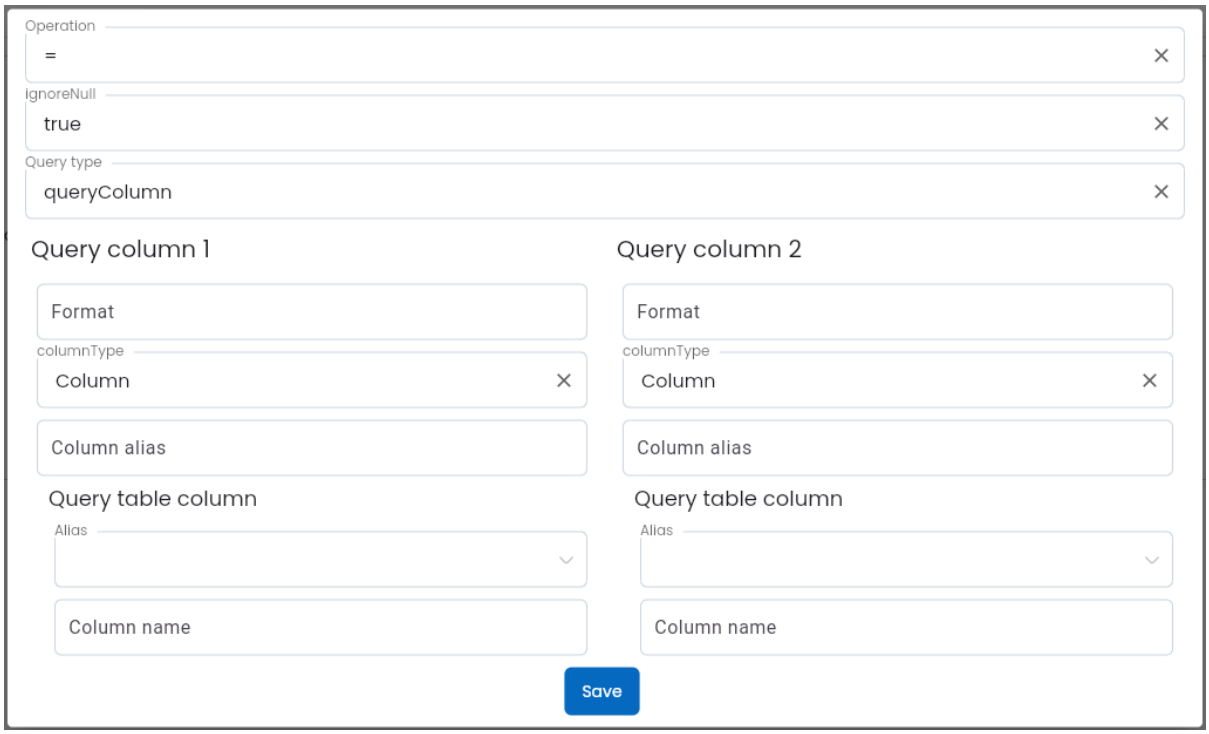
Bước 1: Tại dòng CONDITION LIST, click button **Add**, hệ thống thêm một câu điều kiện



Bước 2: Tại các dòng điều kiện nằm trong CONDITION LIST, click button **Edit**, hiển thị màn hình cấu hình cho câu điều kiện



Bước 3: Điền các thông tin phù hợp cho điều kiện cần sử dụng



Trong đó

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Operation	Có	Combobox		Hiển thị danh sách các toán tử điều kiện được cho phép sử dụng trong driver Connection đã lựa chọn. Người dùng chọn 1 toán tử.
IgnoreNull	Có	Combobox		Hiển thị 2 lựa chọn để kiểm soát: + True: nếu câu điều kiện có thể null + False: nếu câu điều kiện không được null Người dùng thực hiện chọn 1 .
Query Type	Có	Combobox		Hiển thị 2 lựa chọn loại query: + queryColumn: Thực hiện query giữa 2 trường dữ liệu + querySelect: Thực hiện viết câu subquery để lấy dữ liệu trong câu điều kiện

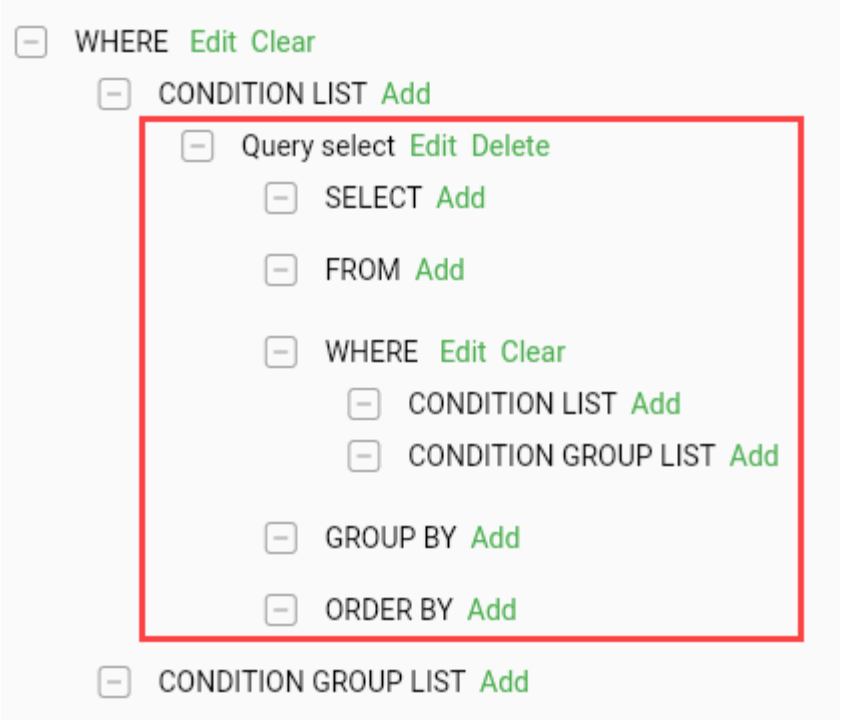
Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				Người dùng thực hiện chọn 1 lựa chọn.
QueryColumn 1 **QueryColumn 2** Chỉ hiển thị khi queryType = “queryColumn”				
Format	-	Text	Chỉ sử dụng cho các điều kiện liên quan tới kiểu dữ liệu Date	Định dạng Date của dữ liệu VD : dd/MM/yyyy
ColumnType	Có	Combobox		Hiển thị giá trị cho phép người dùng chọn 1: + Column: Sử dụng dữ liệu từ cột trong jdbc. + Value: Sử dụng dữ liệu truyền vào từ hệ thống
Column alias	Có	Combobox		Hiển thị giá trị cho phép người dùng chọn 1: + Column: Sử dụng dữ liệu từ cột trong jdbc. + Value: Sử dụng dữ liệu truyền vào từ hệ thống

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Alias	Có	Combobox	Chỉ xuất hiện khi ColumnType = “Column”	Hiển thị giá trị của các table alias đã được khai báo trong giai đoạn cấu hình Edit Query Buider. Người dùng được phép chọn 1.
Column name	Có	Combobox	Chỉ xuất hiện khi ColumnType = “Column”	Hiển thị giá trị các cột trong table alias đã được chọn. Người dùng được phép chọn 1.
Type	Có	Combobox	Chỉ xuất hiện khi ColumnType = “Value”	Hiển thị giá trị của Value: + Value: Dữ liệu được truyền thẳng vào phần cấu hình + parameterName: Dữ liệu sẽ được truyền vào khi call JDBC trong các kịch bản test Người dùng được phép chọn 1.
Value	Có	Text	Chỉ xuất hiện khi Type = “Value”	Giá trị của cột so sánh.

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
isList	Có	Combobox	Chỉ xuất hiện khi ColumnType = “Value”	Hiển thị giá trị Boolean: + True: Dữ liệu truyền vào là 1 List. + False: Dữ liệu truyền vào không phải là 1 List. Người dùng được phép chọn 1.
isNull	Có	Combobox	Chỉ xuất hiện khi ColumnType = “Value”	Hiển thị giá trị Boolean: + True: Dữ liệu truyền vào sẽ là null. + False: Dữ liệu truyền vào không được null. Người dùng được phép chọn 1.
ValueType	Có	Combobox	Chỉ xuất hiện khi ColumnType = “Value”	Hiển thị các kiểu giá trị của dữ liệu. Người dùng được phép chọn 1. Hệ thống mặc định khi tạo mới là “String”

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
IgnoreNull (Value)	Có	Combobox	Chỉ xuất hiện khi ColumnType = “Value”	Hiển thị 2 lựa chọn để kiểm soát: + True: nếu dữ liệu có thể null + False: nếu dữ liệu không được null Người dùng thực hiện chọn 1 .
ParameterName	Có	Text	Chỉ xuất hiện khi ColumnType = “parameterName”	Nhập vào parameter đã được khởi tạo tại phần params List thuộc tab Config

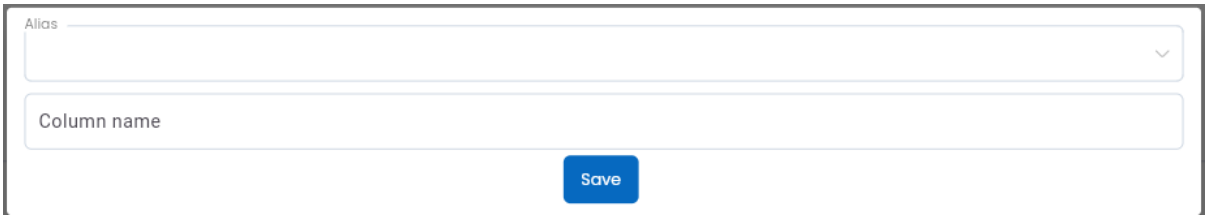
Khi queryType = “querySelect”, hệ thống tự động sinh thêm block subquery tại dòng CONDITION đã edit.



1.1.12.3.2.4 GROUP BY

Bước 1: Click nút add tại dòng GROUP BY để tạo thông tin gom nhóm của JDBC, hệ thống tạo mới 1 dòng thông tin trong list "GROUP BY".

Bước 2: Nhấn **Edit** tại dòng muốn chỉnh sửa, hệ thống hiển thị màn hình thông tin



Trong đó:

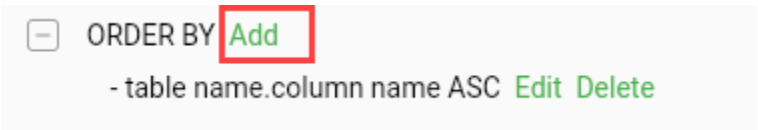
Tên trường	**Bắt buộc**	**Định dạng**	**Ràng buộc**	**Mô tả**
Alias	Có	Combobox		Hiển thị tên tạm thời của table đã

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				được khai báo tại phần FROM. Người dùng được phép chọn 1.
Column name	Có	Button		Hiển thị cột dữ liệu của table alias đã chọn. Người dùng được phép chọn 1.

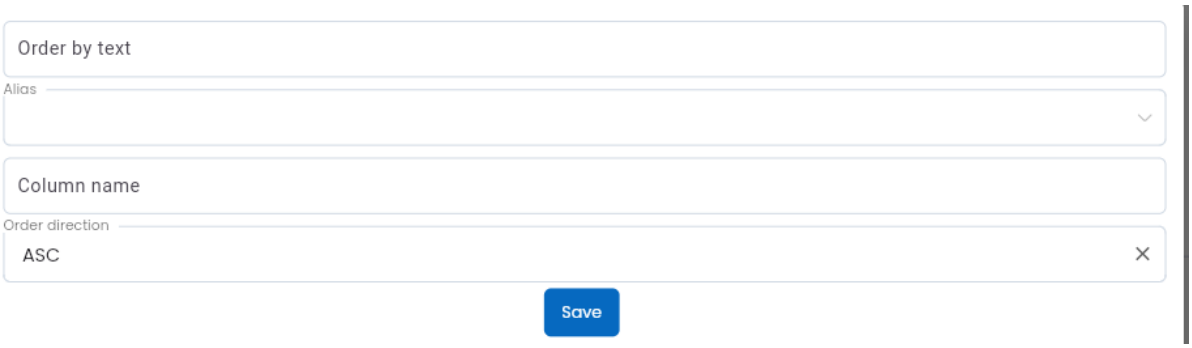
Sau khi nhấn "**Close**", hệ thống cập nhật câu query trên danh sách GROUP BY

1.1.12.3.2.5 ORDER BY

Bước 1: Nhấn **Add** tại dòng ORDER BY để tạo thông tin sắp xếp của dữ liệu jdbc trả về, hệ thống tạo mới 1 dòng thông tin trong list "ORDER BY"



Bước 2: Nhấn **Edit** tại dòng muốn chỉnh sửa, hệ thống hiển thị màn hình thông tin



Trong đó:

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Order by text		Text		Nhập điều kiện để thực hiện sắp xếp
Alias	Có	Combobox		Hiển thị tên tạm thời của table đã được khai báo tại phần FROM. Người dùng được phép chọn 1.
Column name	Có	Button		Hiển thị cột dữ liệu của table alias đã chọn. Người dùng được phép chọn 1.
Order Direction	Có	Combobox		Hiển thị tùy chọn sắp xếp được cho phép trong driver Connection đã chọn: + ASC: Sắp xếp tăng dần + DESC: Sắp xếp giảm dần Người dùng được phép chọn 1.

Sau khi nhấn "**Close**", hệ thống cập nhật câu query trên danh sách ORDER BY.

1.1.12.3.2.6 LIMIT

Bước 1: Nhấn **Edit** tại dòng LIMIT để tạo thông tin số lượng kết quả đầu ra của jdbc:

Bước 2: Hệ thống hiển thị màn hình thông tin, người dùng tiến hành nhập:

Trong đó:

Tên trường	**Bắt buộc**	**Định dạng**	**Ràng buộc**	**Mô tả**
Type	Có	Combobox		Hiển thị giá trị của Value: + Value: Dữ liệu được truyền trực tiếp vào phần cấu hình + parameterName: Dữ liệu sẽ được truyền vào khi call JDBC trong các kịch bản test Người dùng được phép chọn 1.
Value	Có	Text	Chỉ xuất hiện khi Type = “Value”	Giá trị của cột so sánh.

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
isList	Có	Combobox		Hiển thị giá trị Boolean: + True: Dữ liệu truyền vào là 1 List. + False: Dữ liệu truyền vào không phải là 1 List. Người dùng được phép chọn 1.
isNull	Có	Combobox	Chỉ xuất hiện khi ColumnType = “Value”	Hiển thị giá trị Boolean: + True: Dữ liệu truyền vào là 1 List. + False: Dữ liệu truyền vào không phải là 1 List. Người dùng được phép chọn 1.
ValueType	Có	Combobox		Hiển thị các kiểu giá trị của dữ liệu. Người dùng được phép chọn 1. Hệ thống mặc định khi tạo mới là “String”
IgnoreNull	Có	Combobox		Hiển thị 2 lựa chọn để kiểm soát:

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
				+ True: nếu câu điều kiện có thể null + False: nếu câu điều kiện không được null Người dùng thực hiện chọn 1.
ParameterName	Có	Text	Chỉ xuất hiện khi ColumnType = “parameterName”	Nhập vào parameter đã được khởi tạo tại phần params List thuộc tab Config
isNull	Có	Combobox	Chỉ xuất hiện khi ColumnType = “Value”	Hiển thị giá trị Boolean: + True: Dữ liệu truyền vào sẽ là null. + False: Dữ liệu truyền vào không được null. Người dùng được phép chọn 1.

Sau khi nhấn “**Close**”, hệ thống cập nhật câu query trên dòng LIMIT.

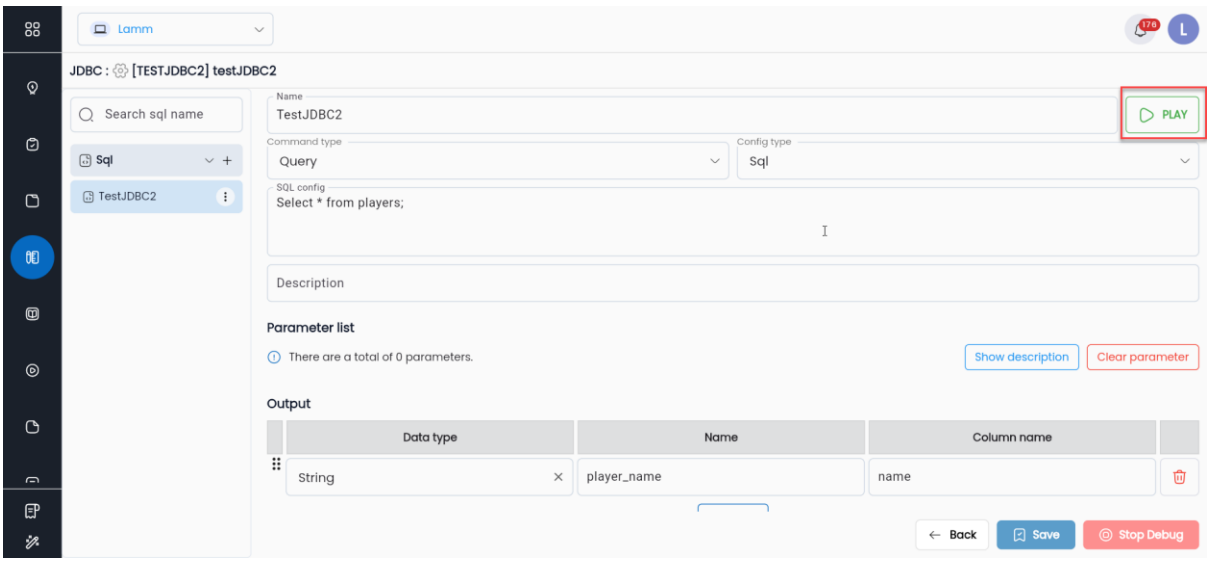
1.1.12.3.2.7 OFFSET

Các trường thông tin và thao tác tương tự như mục “LIMIT”

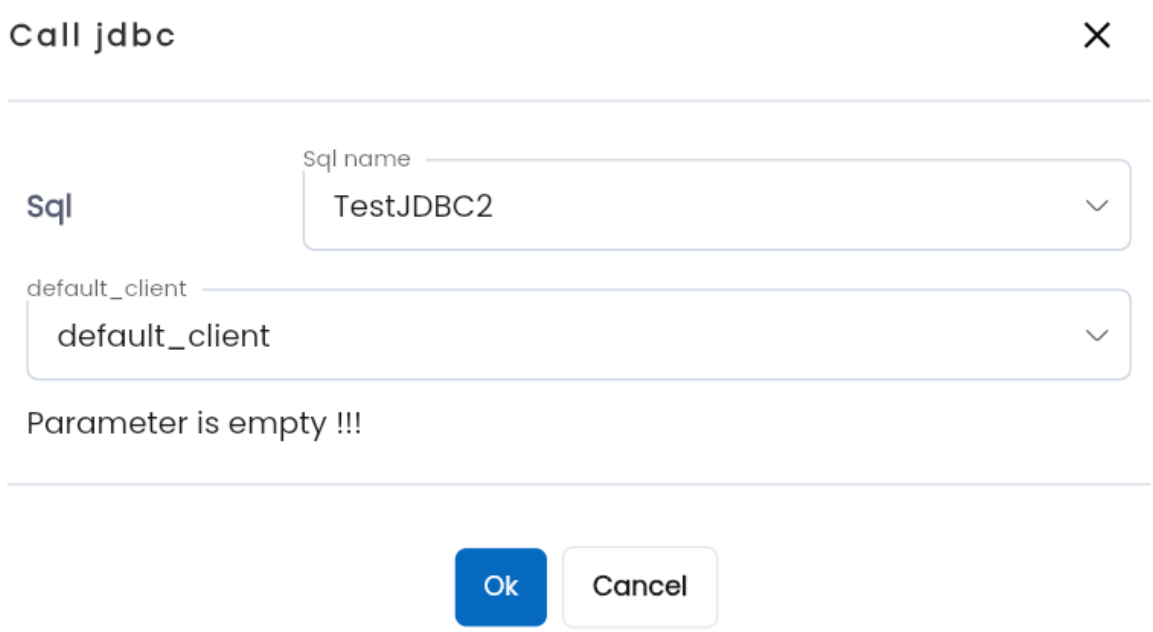
1.1.12.4 Edit Test JDBC

Người dùng nhấn “ ” ở phía bên phải bản ghi test JDBC để chỉnh sửa trường thông tin nhưthao tác thêm mới. Tham chiếu mục [Tạo mới một test jdbc]

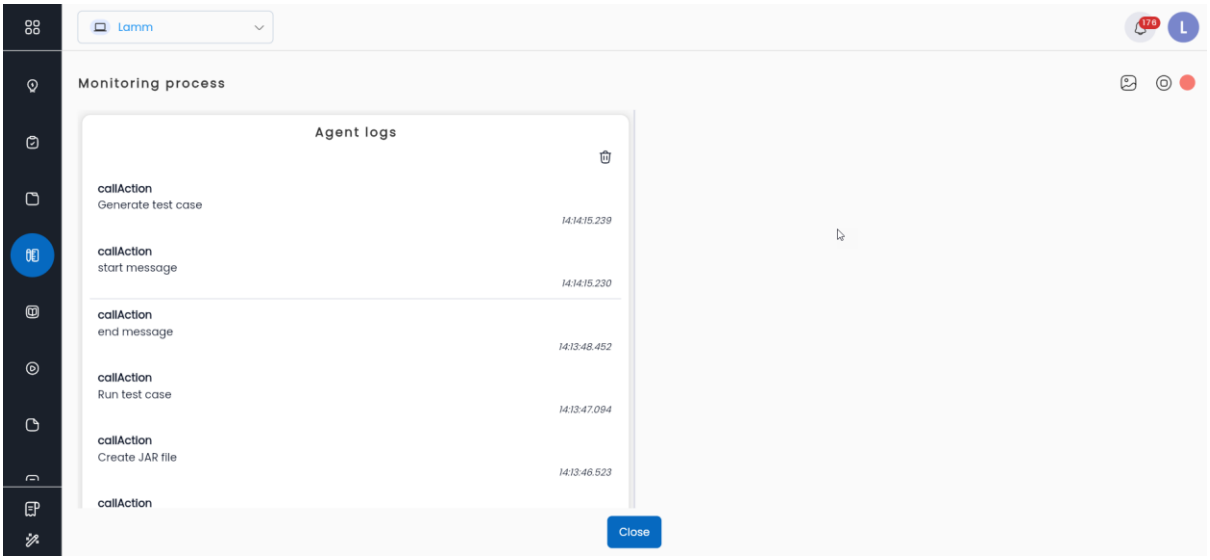
Để thực thi test JDBC, Nhấn **Start debug**, sau đó nhấn “ PLAY ”. Đảm bảo Agent đã chọn JDBC và được khởi động trước đó.



Hệ thống hiển thị màn hình bao gồm thông tin profile và các parameters đã khai báo



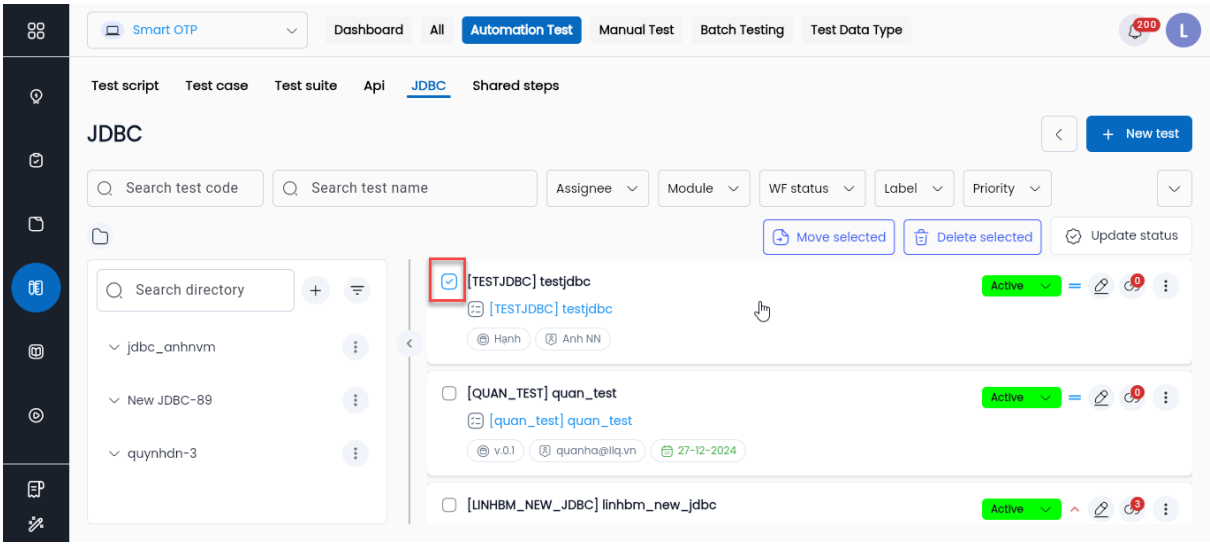
Người dùng nhập giá trị tham số đầu vào (nếu có) và nhấn "**OK**", hệ thống tiến hành thực thi test jdbc và hiển thị màn hình tiến trình thực thi:



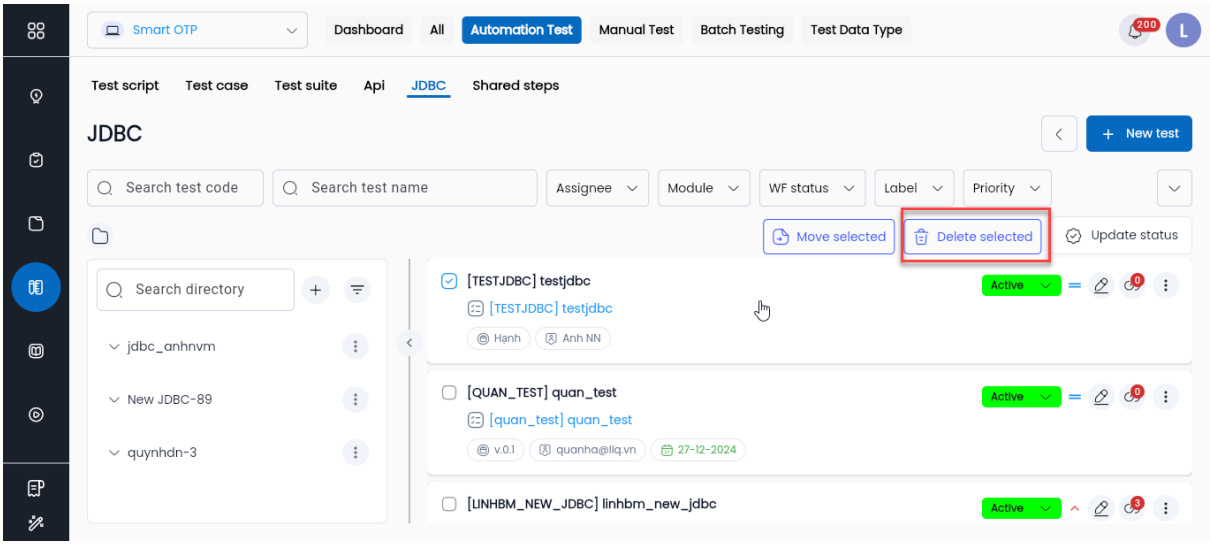
Kết quả run test jdbc từ 1 kịch bản cấu hình bằng query builder trả về name và title.

1.1.12.5 Delete Test JDBC

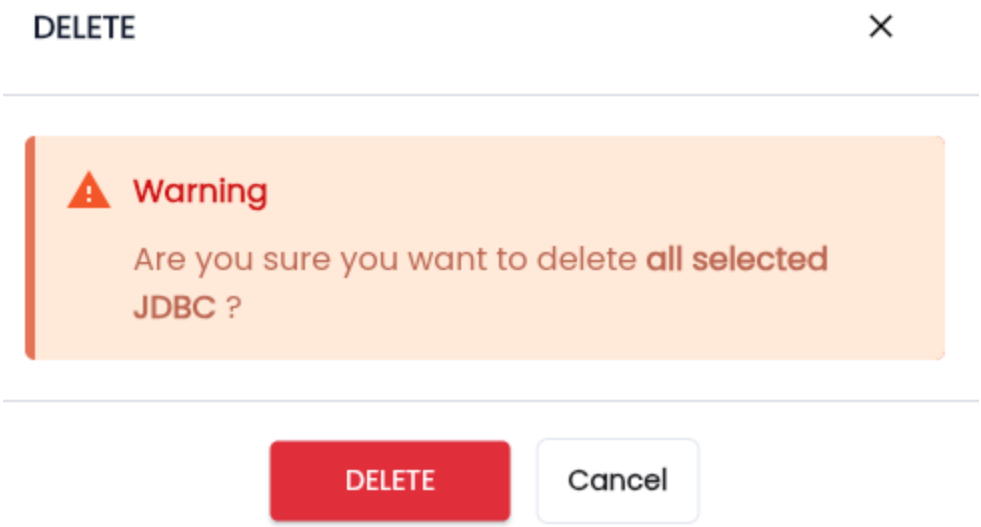
Bước 1: Nhấn chọn kịch bản test JDBC muốn xóa



Bước 2: Nhấn **Delete selected**



Bước 3: Hệ thống hiển thị popup xác nhận xóa



Nhấn **DELETE**, hệ thống hiển thị thông báo thành công, kịch bản test JDBC được xóa khỏi hệ thống.

1.1.12.6 Sử dụng API trong test script, test suite

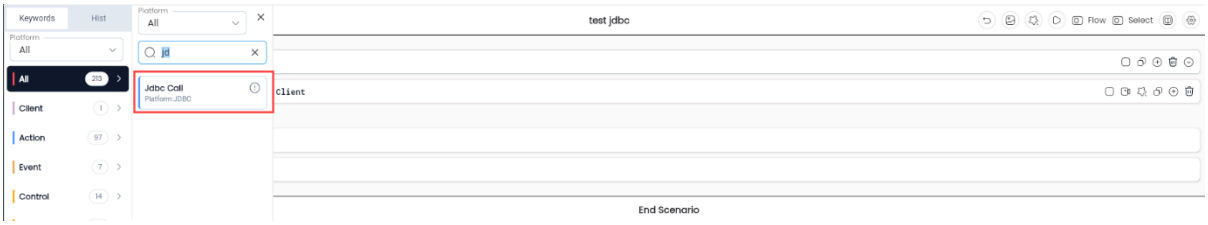
Cấu hình test JDBC là bước cấu hình tạo ra các câu lệnh, còn việc sử dụng test JDBC ở đâu, và sử dụng như thế nào sẽ hoàn toàn do người dùng cấu hình trên test script.

Cách sử dụng JDBC trong Test Script

Bước 1: Truy cập vào màn hình Low-code thuộc các kịch bản kiểm thử. Tham chiếu Low-code Test script.

Bước 2: Sử dụng Key-word Controller là "Client" và cấu hình Platform là JDBC.

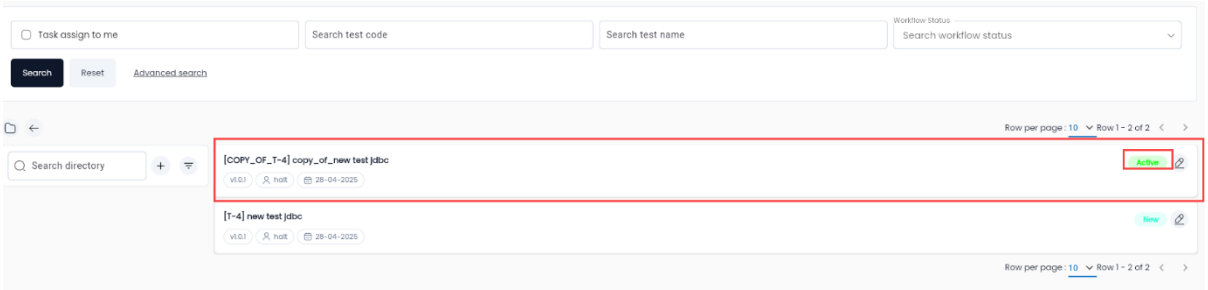
Bước 3: Kéo key-word JDBC CALL vào màn hình low-code bên trong client JDBC



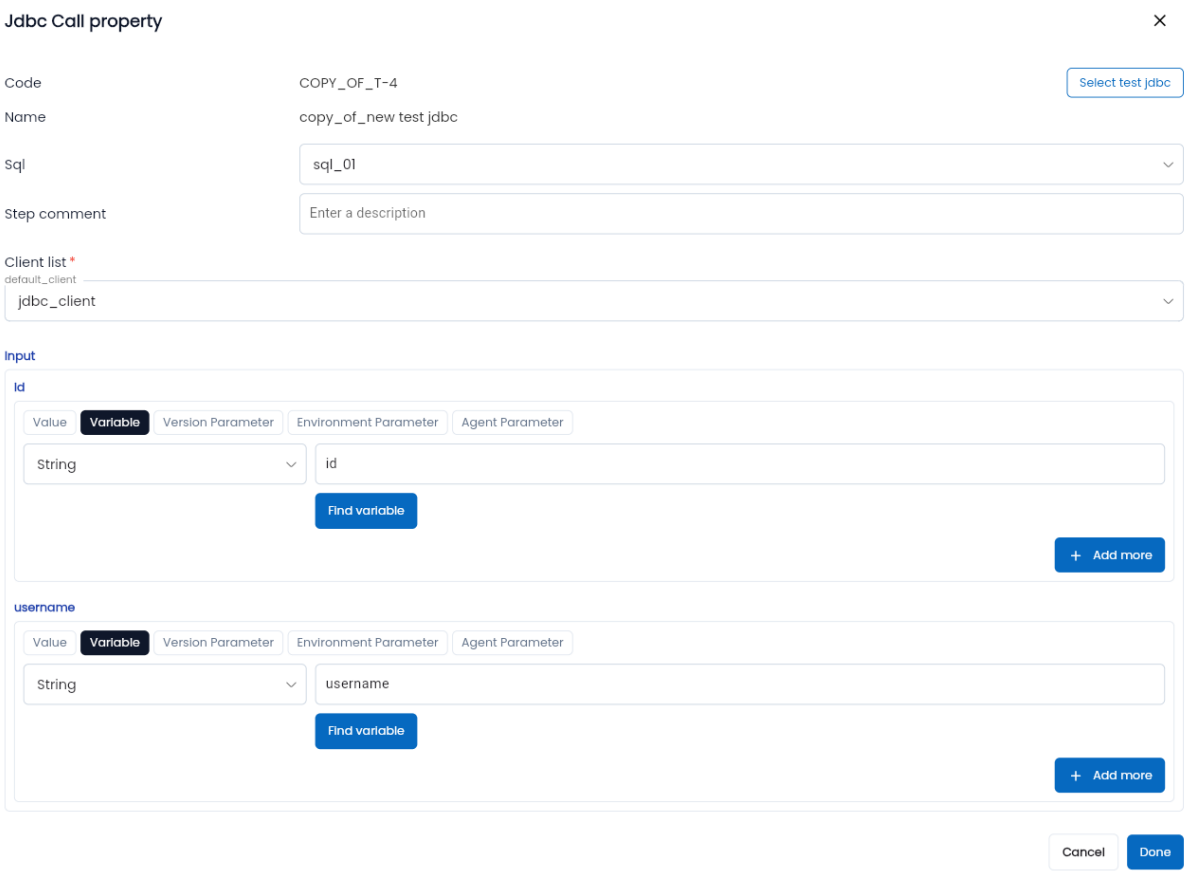
Bước 4: Tiến hành cấu hình JDBC Call Property, lựa chọn dữ liệu code tại màn hình JDBC call property

Bước 5: Màn hình hiển thị danh sách JDBC đã được khởi tạo, click chọn 1 JDBC muốn gọi

Lưu ý: Chỉ có thể gọi test jdbc đang ở trạng thái Active



Bước 6: Chọn jdbc thành công, màn hình JDBC call property của key word JDBC call có sự thay đổi



Nhập các thông tin cần thiết

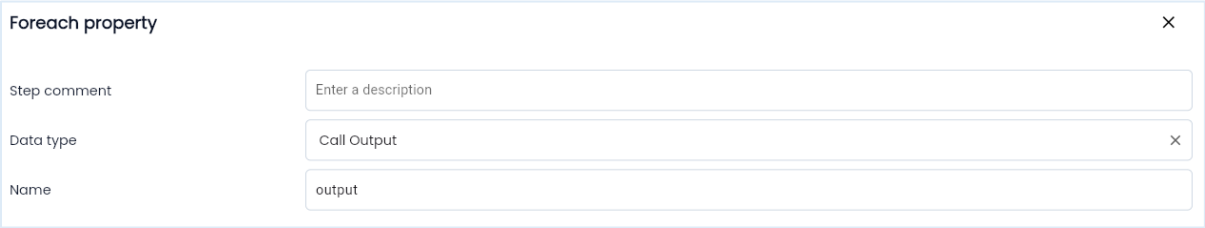
Trường thông tin	**Bắt buộc**	**Định dạng**	**Ràng buộc**	**Mô tả**
Code	Có	AutoComplete		Mã code của jdbc đã chọn tại JDBC Lib
Step Comment	-	Text		Mô tả cho việc gọi jdbc
Input Param đầu vào sử dụng cho JDBC				
Các Parameter Input	-		Hiển thị nếu trong Parameter list của	Nhập dữ liệu sử dụng làm tham số đầu vào của

			jdbc đã chọn có khai báo Param hoặc trong cấu hình query buider (nếu có) có khai báo param.	các parameter trong jdbc. Lựa chọn nguồn dữ liệu và giá trị của phù hợp với nguồn giá trị của param đã chọn.
Output Param đầu ra của JDBC				
Type	Có	Combobox		Hiển thị danh sách kiểu dữ liệu cho biến output. Người dùng được phép chọn 1 giá trị.
Data Type	Có	Combobox	Mặc định lựa chọn Call Output	Kiểu dữ liệu của biến output hứng Output trả về của jdbc
Name	Có	Text		Tên của biến nhận giá trị output.

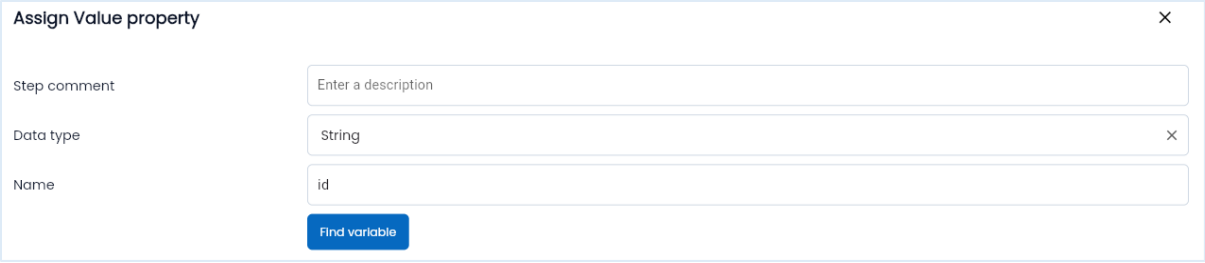
Bước 7: Click button "**Done" để lưu** JDBC call property, ta sẽ được dòng low-code như sau

Bước 8: Sử dụng key-word Assign Value, cấu hình tạo biến để nhận giá trị và giá trị được gán. Dữ liệu trả về khi query trên database là dạng list. Do vậy, sẽ phải dung vòng for để cấu hình lấy ra output.

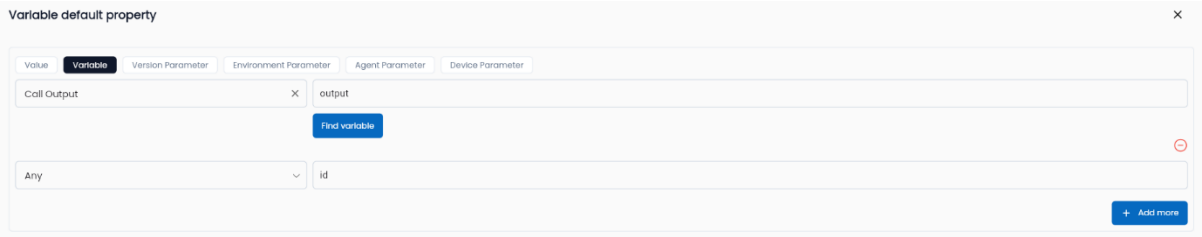
Jdbc_result là 1 output chứa danh sách callOutput vậy nên khi sử dụng vòng for lặp qua output từ jdbc Call sẽ có tính chất là đi qua từng dòng dữ liệu trong output trả về.



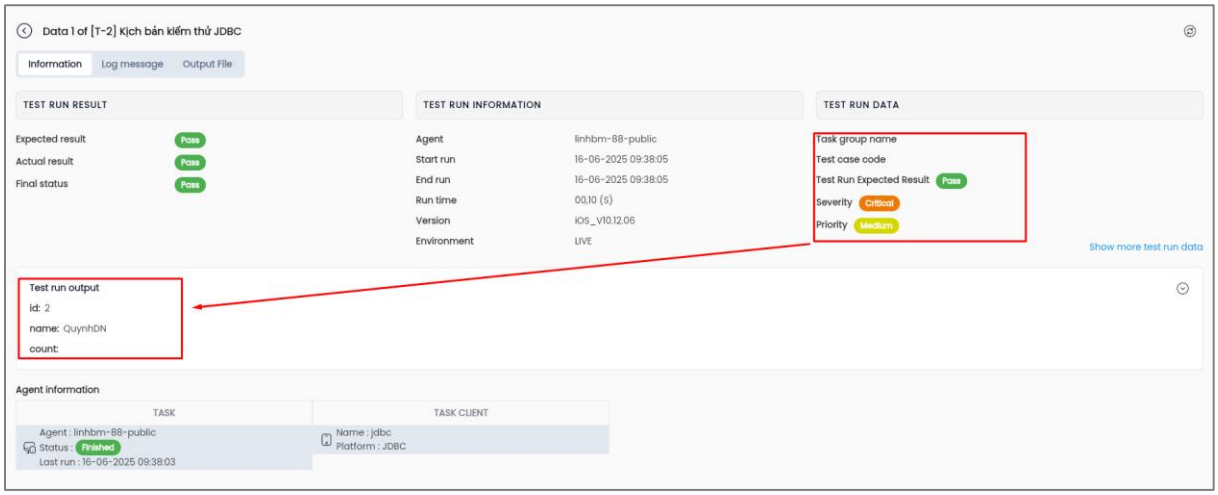
Cấu hình biến để nhận giá trị



Cấu hình giá trị được lấy từ response, tại output được trả về từ jdbc, bộ dữ liệu được trả về theo dạng Call Output. Lấy giá trị của 1 phần tử trong Call Output.



Bước 9: Click **Save** phần Low-code, khi tiến hành thực thi kịch bản thì được dữ liệu trả về theo tham số output đã khởi tạo.



1.1.12.7 Nhân bản test jdbc

Tham chiếu Duplicate Test script

1.1.12.8Thực hiện chức năng multi-select

Tham chiếu mục Thực hiện chức năng multi-select của test script

1.1.12.9 Import Json file

Tham chiếu Import Json file

1.1.12.10 Export Json file

Tham chiếu Export Json file

1.1.13 Test Data Type

1.1.13.1 Test data type là gì?

Test data type: là cách chuẩn hóa dữ liệu test một lần để dùng nhiều lần. Giống như một "mẫu dữ liệu chuẩn" (shared schema) đã được cấu hình sẵn. Nó cho phép truyền

tham số (parameters) giữa các test script và shared step mà không cần tạo mới lại nhiều lần.

Ví dụ: khi test hóa đơn, ta có thể định nghĩa sẵn một schema gồm:

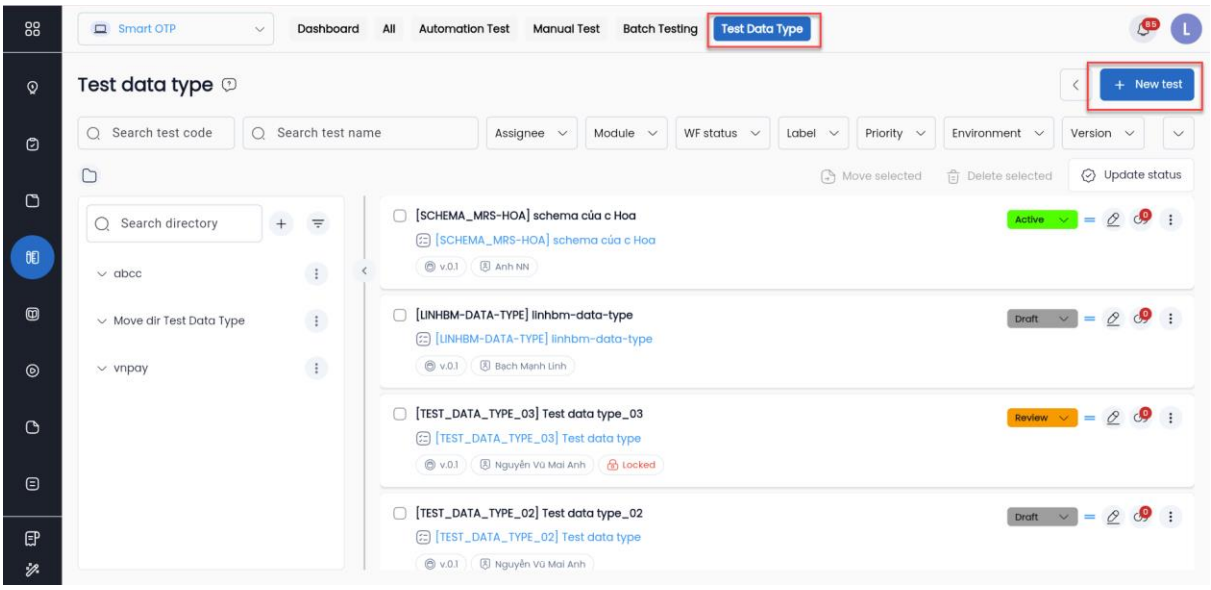
- Thông tin hóa đơn (mã hóa đơn, ngày, khách hàng…)

- Danh sách item trong hóa đơn (tên sản phẩm, số lượng, giá…)

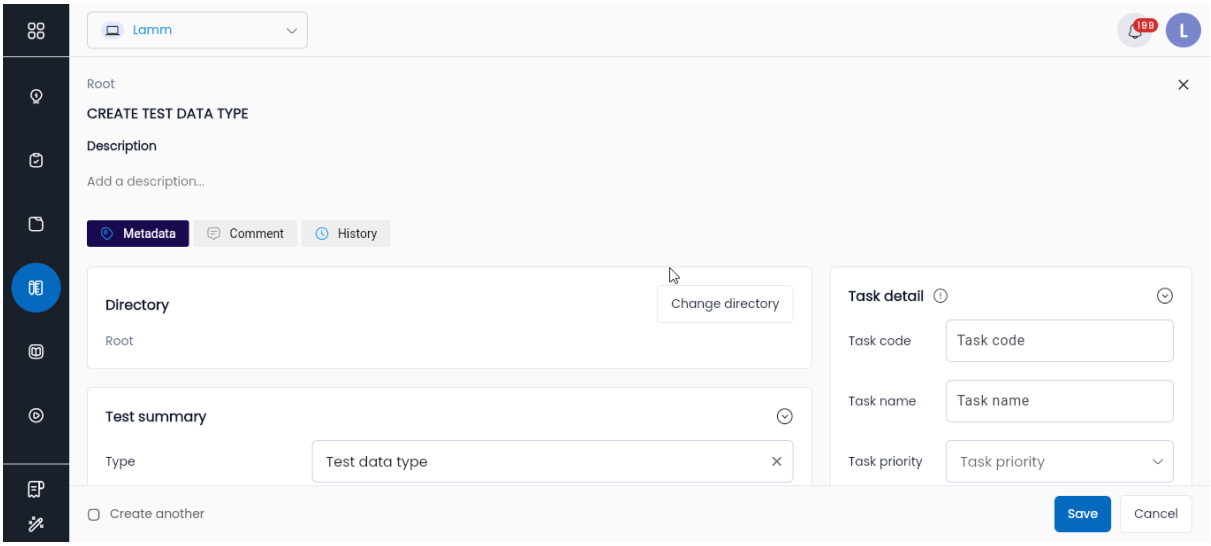
Sau đó chỉ cần gọi shared step và truyền bộ dữ liệu này vào, thay vì phải khai báo lại từng thông tin trong mỗi test.

1.1.13.2 Tạo mới Test data type

Bước 1: Trên màn hình Test Management, chọn tan Test Data Type, nhấn **New Test**



Hệ thống hiển thị màn hình **CREATE TEST DATA TYPE,** tab Metadata



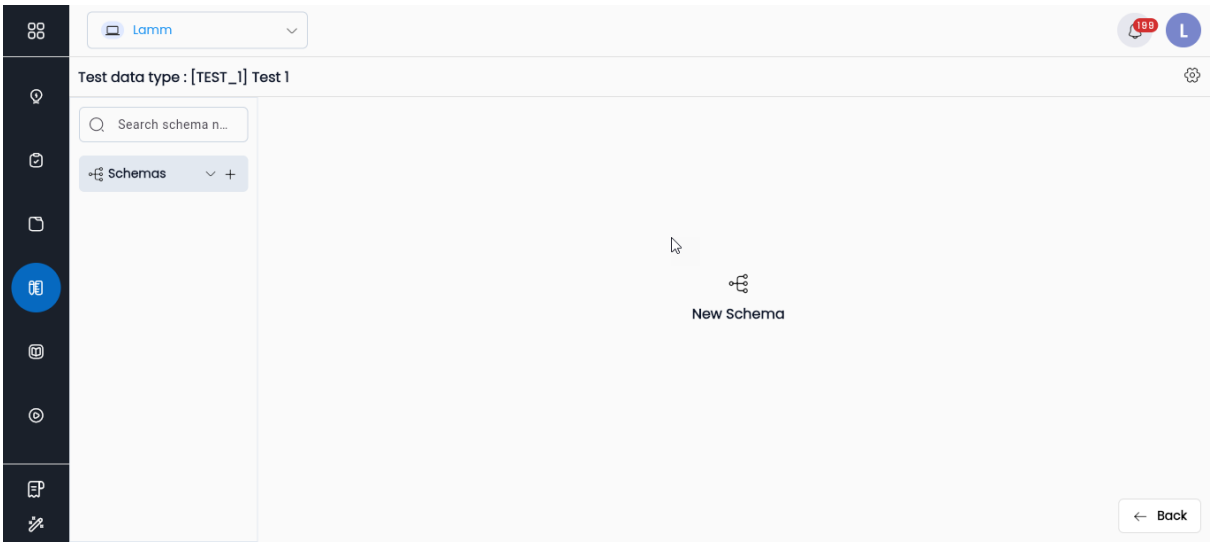
Mô tả các trường thông tin:

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Thông tin cơ bản				
Directory		Button	Hiển thị danh sách các directory	Là thư mục chứa kịch bản kiểm thử sau khi được khởi tạo trong project. Cách chọn 1 directory
Type	Có	Combobox		Là tên của loại test đang được tạo mới.
Test code	Có	Textbox	Nếu nhập người dùng nhập code thì code phải là duy nhất, độ dài tối thiểu 2 ký tự, bắt đầu bằng chữ cái, chỉ chứa các ký tự từ a-z	Là mã dùng để định danh kịch bản kiểm thử trên hệ thống Nếu code bỏ trống, hệ thống sẽ tự sinh theo cấu trúc T-stt, trong đó stt là số thứ tự của test trong số các kịch bản kiểm thử được đánh mã tự động của Project

Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
			(viết thường, viết hoa), chữ số, "_" và "-"	
Test name	Có	Textbox		Là tên của kịch bản kiểm thử Người dùng nên đặt tên có nghĩa, liên quan đến chức năng/ trường hợp kiểm thử của test được khởi tạo
Module	Có	Combobox	Những module ở trạng thái Active	Tên của Module
Version	Có	Combobox	Những version đang ở trạng thái Active	Tên của Version
Environment	Có	Combobox	Những environment đang ở trạng thái Active	Tên của environment
Label	-	Searchbox	Label ở trạng thái Active	Chọn 1 hoặc nhiều nhãn mà người dùng muốn gắn. Cách gắn nhãn
Tab **Task Details**				
Task code		Textbox		Nhập mã cho task
Task name		Textbox		Nhập tên cho task

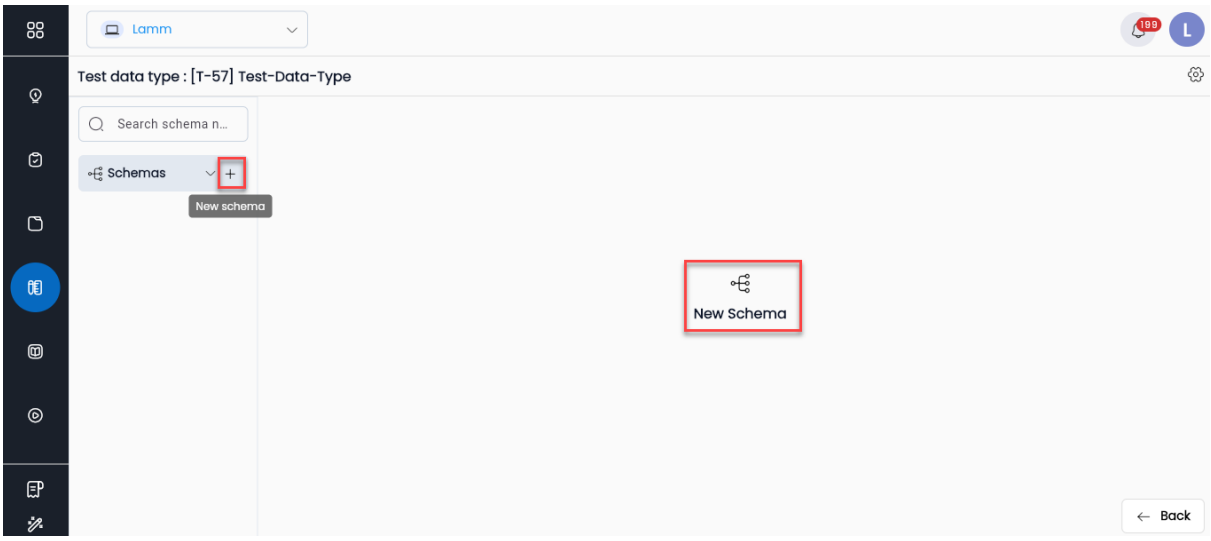
Tên trường	Bắt buộc	Định dạng	Ràng buộc	Mô tả
Task priority		Combobox		Mức độ ưu tiên của task. Bao gồm các giá trị: - Highest - High - Medium - Low - Lowest
Task difficulty		Combobox		Mức độ khó thực hiện của task. Bao gồm các giá trị: - Very Hard - Hard - Medium - Easy - Very Easy Hệ thống mặc định là Medium
Task status		Text		Trạng thái của task. Được hiển thị mặc định theo trạng thái ban đầu của task.
Tab **Assign to**				
Assign to		Button		Lựa chọn thành viên có trong dự án để chỉ định cho task. Hệ thống mặc định hiển thị thành viên đang tạo task. Có thể chọn nhiều thành viên.
Scheduled start date		Date picker		Ngày dự kiến bắt đầu thực hiện task.
Due date		Date picker		Ngày kết thúc task.

Bước 2: Nhập đầy đủ các thông tin cần thiết, nhấn **Save**. Hệ thống hiển thị thông báo thành công, đồng thời hiển thị màn hình cấu hình Test Data Type

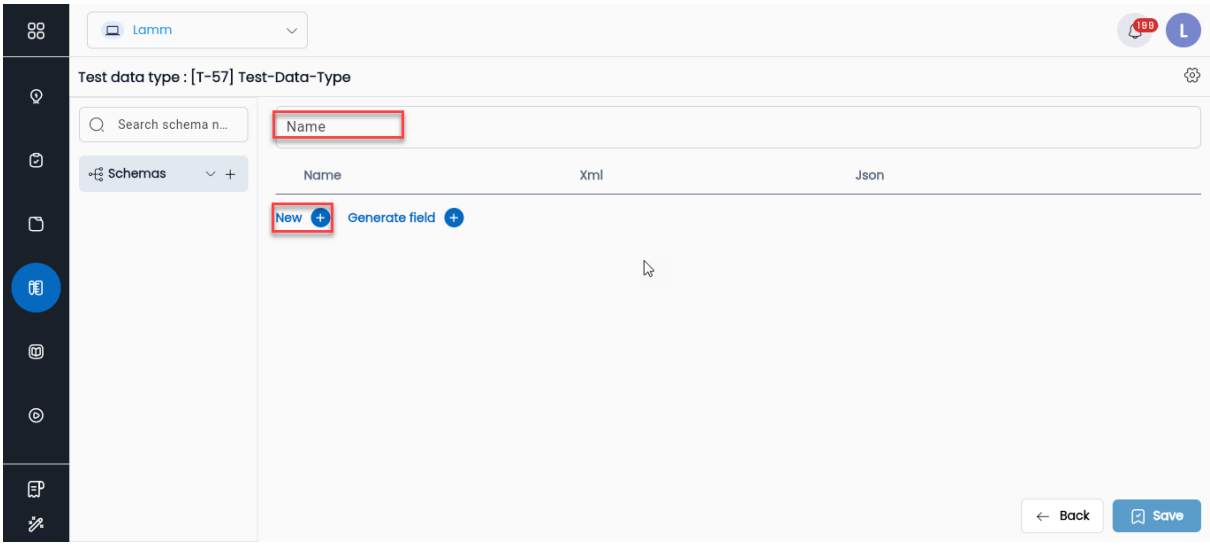


1.1.13.3 Cấu hình Test Data Type

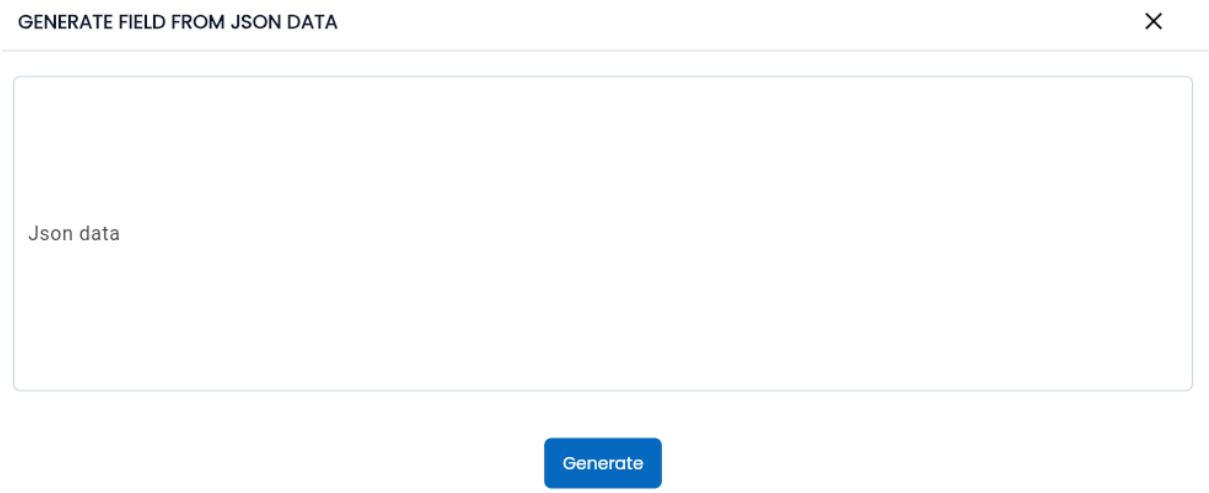
Bước 1: Nhấn **New Schema** chính giữa màn hình hoặc nhấn biểu tượng New schema phía bên phải



Bước 2: Hệ thống hiển thị màn hình cấu hình Schema



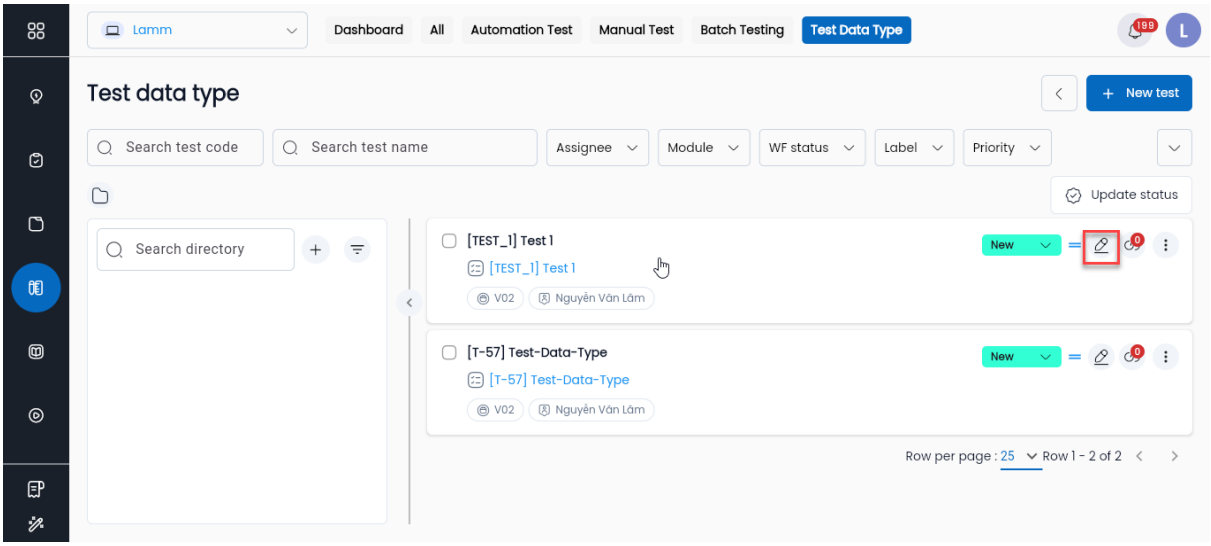
Người dùng nhập tên cho Schema, Nhấn New để bắt đầu tạo một schema. Người dùng có thể nhấn Generate field nếu đã có một Json Data trước đó, hệ thống sẽ tự động tạo schema dựa theo Json Data người dùng đã cung cấp



Bước 3: Nhấn **Save**. Hệ thống hiển thị thông báo thành công.

1.1.13.4 Chỉnh sửa thông tin Test Data Type

Bước 1: Trên màn hình thư viện Test data type, nhấn Edit metadata ở phía bên phải bản ghi Test data type muốn chỉnh sửa thông tin



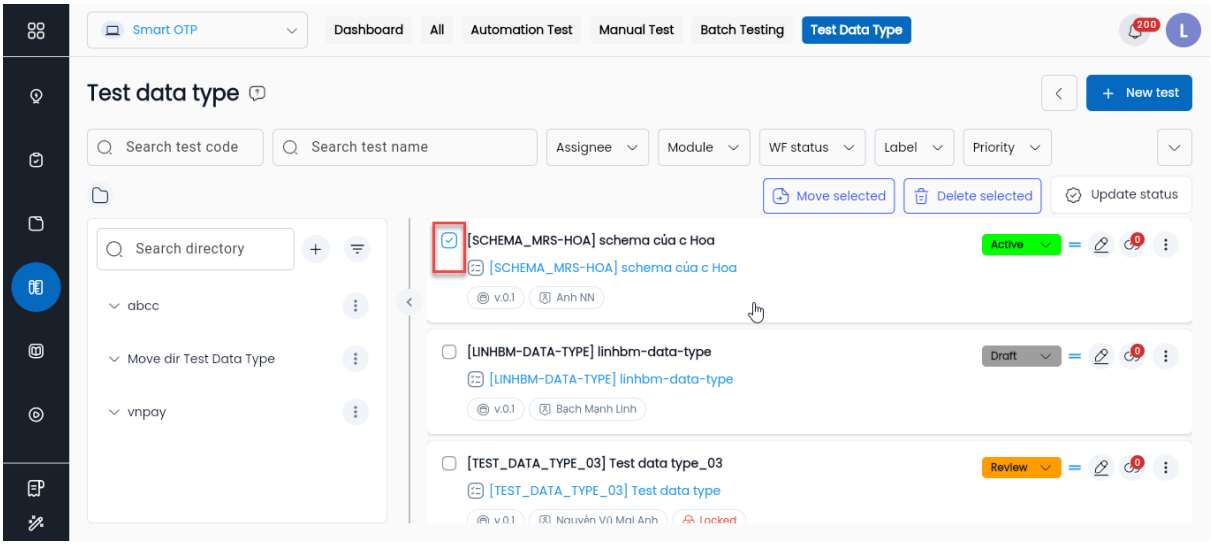
Hệ thống hiển thị màn hình chỉnh sửa, các trường thông tin tương tự khi thêm mới. [Thêm mới Test data type]

Lưu ý: Trường **Code** không cho phép chỉnh sửa.

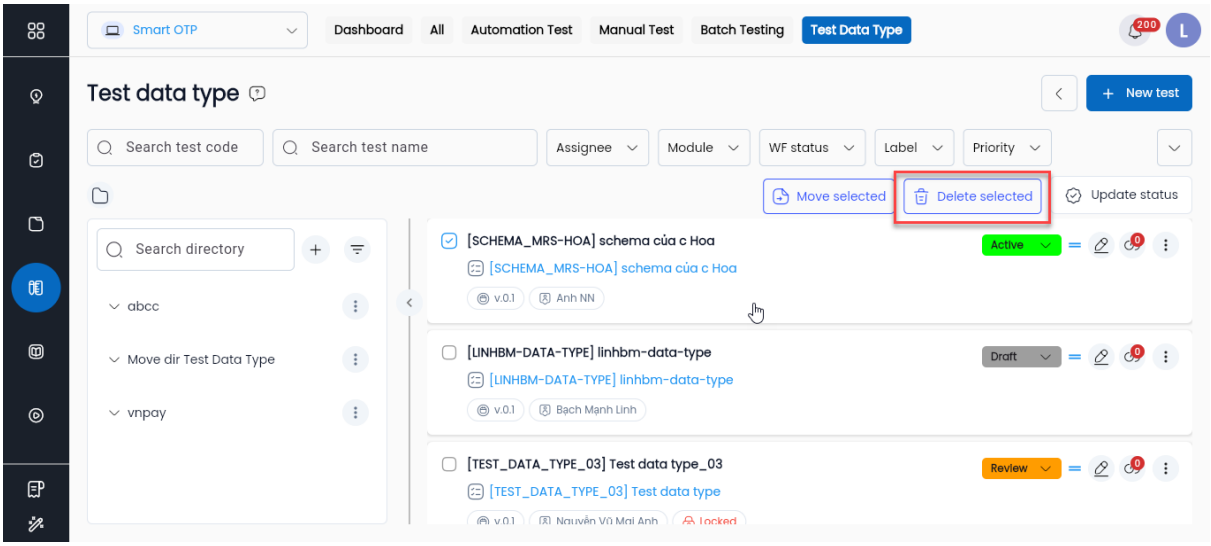
Bước 2: Thực hiện cập nhật thông tin theo nhu cầu. Nhấn Save, hệ thống hiển thị thông báo thành công.

1.1.13.5Xóa Test Data Type

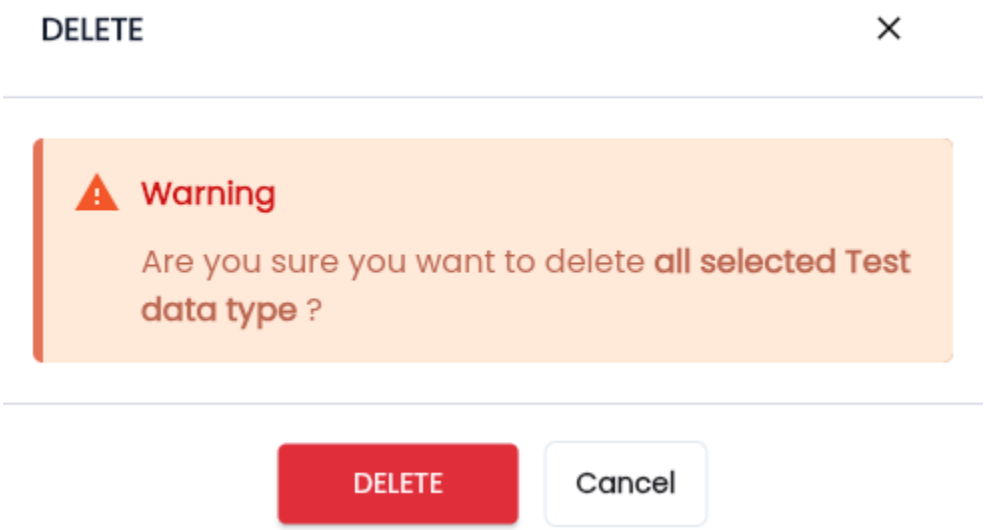
Bước 1: Nhấn chọn bản ghi Test Data Type cần xóa



Bước 2: Chọn **Delete selected**



Bước 3: Hệ thống hiển thị popup xác nhận xóa.



Nhấn **DELETE**, hệ thống hiển thị thông báo thành công, bản ghi Test Data Type được xóa khỏi hệ thống.

1.1.13.6 Nhân bản Test Data Type

Tham chiếu Nhân bản kịch bản kiểm thử

1.1.13.7 Import Json File

Tham chiếu Import json file

1.1.13.8 Export Json File

Tham chiếu Export json file