

1 Kiến trúc hệ thống

Hệ thống bao gồm hai thành phần chính:

- **Ứng dụng Web:** Trung tâm quản lý, cấu hình và thực thi kiểm thử tự động.
- **Agent (ứng dụng Client):** Trung gian điều phối kết nối giữa hệ thống và ứng dụng mục tiêu.

2 Tính năng nổi bật

- Giao diện đơn giản, dễ thao tác.
- Hoạt động hoàn toàn trên nền **web**, hỗ trợ **Windows** và **Tablet**.
- Hỗ trợ kiểm thử cho:
 - Web application
 - iOS/Android applications
 - Windows applications
 - API
 - JDBC
- Kiểm thử đa môi trường trong cùng một Test case/Test script.
- Đối tượng sử dụng:
 - Người dùng nghiệp vụ (không yêu cầu kỹ năng lập trình)
 - Manual Tester
 - Automation Tester

3 Các chức năng chính

- **Thiết lập Test Project:** quản lý dự án, phiên bản, môi trường, người dùng, profile...
- **Dashboard & Báo cáo:** thống kê test script theo trạng thái.
- **Quản lý Test Case/Script/Test Suite:** tạo và cấu hình các kịch bản kiểm thử.
- **Test Runs:** theo dõi lịch sử thực thi.

- **Xóa dữ liệu Test runs chạy ở chế độ draft (nháp):** làm sạch dữ liệu tạm.
- **Quản lý tệp đính kèm:** quản lý tài liệu và dữ liệu hỗ trợ kiểm thử.

Với **Centralize Automation Testing Suite**, việc kiểm thử tự động trở nên **dễ tiếp cận, hiệu quả và trực quan** – phù hợp cho cả người mới bắt đầu lẫn các chuyên gia kiểm thử.

Trong tài liệu này, bạn sẽ được hướng dẫn **từng bước** để cài đặt, cấu hình và sử dụng bộ công cụ, kèm theo những **mẹo và chiến lược tối ưu** giúp quy trình kiểm thử của bạn đạt hiệu quả cao nhất.

Hãy cùng bắt đầu và khám phá cách mà **Centralize Automation Testing Suite** có thể **nâng cao chất lượng sản phẩm và tăng tốc hiệu suất phát triển phần mềm** của bạn!

4 Các khái niệm chính

Khái niệm	Mô tả
Test case	Test case là đơn vị kiểm thử nhỏ nhất trong quá trình kiểm thử phần mềm. Mỗi Test case mô tả một điều kiện đầu vào, các bước thực hiện, kết quả mong đợi, và trạng thái thực tế. Nó giúp đảm bảo rằng một chức năng cụ thể của phần mềm hoạt động đúng như mong đợi. Test case thường được xây dựng dựa trên test script, là một tình huống kiểm thử tổng quát, và một Test script có thể bao gồm nhiều Test case với các dữ liệu kiểm thử (Test data) khác nhau.
Test script	Test script (kịch bản kiểm thử) là một mô tả ở cấp độ cao về chức năng hoặc hành vi của hệ thống cần được kiểm thử. Nó thể hiện một luồng chức năng mà người dùng thực hiện và thường đại diện cho một mục tiêu kiểm thử cụ thể. Test script giúp nhóm kiểm thử hình dung được các tình huống thực tế mà người dùng có thể gặp phải, từ đó phát triển các Test case chi tiết cho từng điều kiện và dữ liệu khác nhau.
Test suite	Test suite là một tập hợp các Test case hoặc Test script được nhóm lại với nhau nhằm phục vụ cho một mục đích kiểm thử chung. Ví dụ như kiểm thử một chức năng, một module, hoặc một chu kỳ kiểm thử cụ thể trong dự án phần mềm. Nói cách khác, Test suite giúp tổ chức và quản lý các bài kiểm thử theo một cách có hệ thống để dễ dàng thực thi, theo dõi và phân tích kết quả.

Test manual	Manual testing (kiểm thử thủ công) là một phương pháp kiểm thử phần mềm trong đó các tester thực hiện kiểm thử mà không sử dụng công cụ tự động hóa. Tester sẽ thực hiện các bước kiểm thử theo Test case đã định sẵn, quan sát kết quả và ghi lại lỗi (nếu có) bằng cách sử dụng chính phần mềm như người dùng thực tế.
Profile	Là môi trường kiểm thử dùng để mô phỏng điều kiện chạy thực tế của ứng dụng. Một profile giúp xác định rõ các yếu tố kỹ thuật mà ứng dụng sẽ được kiểm thử trên đó. Bao gồm: Nền tảng (Web / Mobile / Desktop), Hệ điều hành (Windows, macOS, Linux, Android, iOS...), Trình duyệt (Chrome, Firefox, Safari, Edge...), Độ phân giải màn hình (khi kiểm thử giao diện) ...
Environment	Là môi trường hệ thống nơi phần mềm được triển khai để kiểm thử hoặc vận hành. Mỗi environment đại diện cho một giai đoạn cụ thể trong quy trình phát triển phần mềm. Các loại Environment phổ biến như: Development (dành cho lập trình viên phát triển và kiểm tra nhanh các thay đổi (unit test)), Testing / QA (dành cho kiểm thử chức năng, kiểm thử tích hợp, do đội QA thực hiện), UAT-User Acceptance Testing (môi trường cho người dùng chấp nhận phần mềm), Production (môi trường thực tế, nơi phần mềm phục vụ người dùng cuối)
Version	Là phiên bản phần mềm thể hiện một trạng thái cụ thể của hệ thống tại một thời điểm trong quá trình phát triển và phát hành. Mỗi version thường phản ánh các thay đổi như tính năng mới, sửa lỗi, cải tiến hiệu năng hoặc thay đổi kiến trúc.
Module	Là một thành phần chức năng độc lập trong hệ thống phần mềm, được thiết kế để thực hiện một hoặc một nhóm chức năng cụ thể. Mỗi module thường tương ứng với một màn hình, trang, hoặc tính năng chính trong ứng dụng.

5 Thiết lập ban đầu

5.1 Agent là gì?

Agent là một ứng dụng Java, đóng vai trò là thành phần quan trọng trong hệ thống **LLQ QA Platform**. Nhiệm vụ của Agent là nhận lệnh từ LLQ QA Platform Server và truyền các lệnh đó đến trình duyệt hoặc hệ điều hành trên máy đã cài đặt Agent, cũng như các thiết bị được kết nối. Ngoài ra, Agent còn chịu trách nhiệm kiểm soát các nền tảng và môi trường kiểm thử như hệ điều hành (Android, iOS) và trình duyệt (Chrome, Firefox, v.v.).

Agent được chia thành hai loại chính:

- **Private Agent:** Là dạng Agent cá nhân, chỉ tài khoản đã tạo Agent mới có quyền sử dụng. Thường được dùng để cấu hình Test script và thực hiện Test run với mục đích cá nhân. Các thiết bị client thường sử dụng loại Agent này.
- **Public Agent:** Là dạng Agent dùng chung, cho phép bất kỳ người dùng nào trong hệ thống cũng có thể sử dụng. Chủ yếu phục vụ mục đích thực thi Test run trong quá trình kiểm thử.

Khuyến nghị: Khi tạo Agent trên máy cá nhân, **nên sử dụng Private All-in-One Agent** để thuận tiện cho cả cấu hình và thực thi kịch bản kiểm thử

5.2 Cài đặt agent

Chi tiết cách cài đặt Agent xem chi tiết ở file “**Document_QA Platform_Hướng dẫn cài đặt và sử dụng Agent**” đính kèm