



QAautoMATER

User Manual

CONTENTS

Page	Subpage	Page No.
Introduction		3
Login		4-5
Manual Testing	Configuration	6-9
	Test Case	10-12
	Defect	13-14
	Test Execution	15-17
	Dashboard	18-19
Automation Testing	Configuration	21-27
	Object Repository	28-28
	Test Scripts	29-34
	Test Data	35-38
	OR -Locator Parameterization	39-40
	Session Variable	41-41
	Action for similar element	42-42

INTRODUCTION

Welcome to the user manual for our automation testing software product QAautoMATER! Our software is designed to make your software testing process faster, more efficient, and more accurate. With our software, you can automate your tests using ML algorithm, reducing the time and resources required for manual testing, while ensuring that your software performs as expected. In this manual, we will provide you with a comprehensive guide to using our software, including installation instructions, basic and advanced usage, and troubleshooting tips. Whether you are a seasoned software developer or a newcomer to the field, this manual will help you to harness the power of our automation testing software and take your software testing process to the next level.


QAautoMATER is a complete codeless, ML Integrated, end to end QA Testing solution for web applications, Rest API and Hybrid mobile applications. This tool does not require the professionals using it to have any level of coding expertise as a pre-requisite. It is a UI driven tool, which can be used by any QA professional to start from writing manual test cases, then getting the automated test scripts/ test suite.

1. LOGIN

Login to the QAautoMATER on cloud Web Interface using [QAautoMATER](#). (This tool can be access both on it is on cloud version as well the local version where it is installed in the client's environment. If you have the tool Installed in your environment, please use the respective URL to login.).

- 1.1. Provide your email Id registered with QAautoMATER. Click on Next.

https://qaautomater.vercel.app




QAautoMATER

Email

Next

- 1.2. Provide your password and choose the respective account from the drop-down list, which you are authorized to access.

https://qaautomater.vercel.app



QAautoMATER

Password

Account

Mars ▼

Next

After successful Login, the user will be presented with a beautiful welcome page, which contains QAautoMATER lifecycle, an insightful comparison chart and the Latest updates regarding our tool. On the left hand side, QAautoMATER's menu bar is present which contains paths for Manual and Automation. As an end-to-end QA solution provider we facilitate writing of manual test cases on this tool and then getting them automatically converted into automation test scripts. The steps for these will be discussed in the next few pages.

QAautoMATER

Manual >

Automation >

Manual TC Development

Automation TS Development

Continuous Integration

Test Execution

Defect Logging

One stop solution

One stop solution

QAautoMATER is end to end one stop solution. It is UI driven codeless QA solution on cloud which supports ML algorithm.

LATEST UPDATE

Upcoming release of QAautoMATER will include manual platform for QA analyst (Test case, Test Plan, Defect Logging). Expected date for this release is end of Feb 2023.

Why choose us

COMPARISON CHART

FEATURES	QAAUTOMATER	KATALON	MABL	OPKEY
No code	✓	✗	✓	✓
Coding skill required	✗	✓	✗	✗
ML Step generator	✓	✗	✗	✗
Fastest script generation	✓	✗	✗	✗
Auto healing capability	✓	✓	✓	✓
Most Granular Dynamic Dashboard	✓	✗	✗	✗
Direct Debugging on Test step level	✓	✓	✗	✗
No Hidden cost	✓	✗	✗	✗
Single click Web test script generation from Manual Test case	✓	✗	✗	✗
Bulk Test script generation on a single click	✓	✗	✗	✗
CI/CD support	✓	✓	✓	✓
Window App automation	✗	✓	✓	✓

Tool Comparison

Best Result at the most affordable price

1.1 MANUAL - CONFIGURATION

From the left navigation panel, expand the "Manual" section and click the "Configuration" link.

QAautoMATER

Manual

Dashboard

Test Cases

Test Execution

Defect

Configuration

Automation

Test Configuration

TEST CONFIGURATION

Current Test Cycle: Demo Sprint 1.1 [Save]

Add New Test Cycle: [Save]

ADD NEW ENVIRONMENT [Add] [Save] [Delete]

	#	Environment	URL
○	1	QA	https://www.demoblaze.com/index.html

Add Test Cycle

On the configuration page, you can add a new sprint cycle by entering a sprint name in the 'Add New Test Cycle' field and clicking the 'Save' button

New Test Cycle is successfully added.

Test Configuration

TEST CONFIGURATION

Current Test Cycle: Demo Sprint 1.0 [Save]

Add New Test Cycle: Demo Sprint 1.1 [Save]

ADD NEW ENVIRONMENT [Add] [Save] [Delete]

	#	Environment	URL
○	1	QA	https://www.demoblaze.com/index.html

After successfully adding the test cycle, select your current cycle from the 'Current Test Cycle' dropdown and click 'Save'.

Current Test Cycle is successfully added.

Test Configuration

TEST CONFIGURATION

Current Test Cycle: Demo Sprint 1.1 [Save]

Add New Test Cycle: [Save]

ADD NEW ENVIRONMENT [Add] [Save] [Delete]

	#	Environment	URL
○	1	QA	https://www.demoblaze.com/index.html

Add/Delete Environment

For adding a new Environment, click on the Add button, provide Environment name and URL details, and click the Save button.

RENAME/DELETE TEST COMPONENT

Rename

Delete

New Name*

NewName

-

Spacecraft

AgileAllLabs

OldComponent

To delete a component, select the desired component from the tree structure, click the 'Delete' button, and all associated test cases inside will be deleted.

RENAME/DELETE TEST COMPONENT

Rename

Delete

New Name*

-

Spacecraft

AgileAllLabs

OldComponent

Move your test cases

This feature is used to move test cases from component A to component B.

To move test cases, please follow these steps: Select the source component, choose the test case(s) you wish to move, select the destination component, and then click the 'Save' button

MOVE YOUR TEST CASES

Save

Component*

-

Spacecraft

AgileAllLabs

OldComponent

Select Test to move

QB-12

x

New Component*

-

Spacecraft

AgileAllLabs

OldComponent

Recommendation:

It is recommended to update the current cycle at the end of each sprint. To do this, select the most recent sprint from the 'Current Test Cycle' dropdown and click 'Save'. This will ensure that the testing team is working on the most up-to-date sprint and that everyone is aligned on what tasks need to be completed. Updating the current cycle also helps to maintain a clear and organized testing process, which can ultimately lead to more efficient testing and higher-quality software.

1.2 MANUAL – TEST CASES

From the left navigation panel, expand the "Manual" section and click the "Test Cases" link.

The screenshot shows the QAautoMATER application interface. On the left is a navigation menu with the 'Manual' section expanded, showing options like Dashboard, Test Cases (highlighted), Test Execution, Defect, Configuration, and Automation. The main area is titled 'Test Case' and contains two panels. The 'NEW TEST CASE DETAILS' panel has fields for Title*, Placeholder*, Priority* (set to Medium), Type* (set to Functional), and Automation type* (set to Not Automated), with a 'Save Test Case' button. The 'TEST COMPONENT STRUCTURE' panel shows a tree view with a root folder '+ Spacecraft' and a 'Create Placeholder' button.

Create Placeholder or Component

To save your manual test cases, please create a folder named [PLACEHOLDER NAME] within this directory and save your test cases under that folder

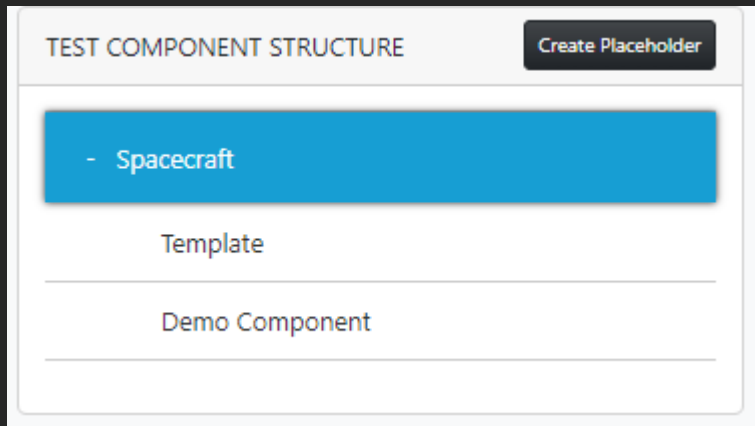
Click on Root folder (account) from the TEST COMPONENT STRUCTURE section and after that click Create Placeholder button

This screenshot shows the 'Add New Placeholder' dialog box in the foreground, which has a text input field for 'New Placeholder*' and a 'Save' button. In the background, the 'TEST COMPONENT STRUCTURE' panel is visible, showing the '+ Spacecraft' root folder highlighted with a blue box. A blue arrow points from the 'Create Placeholder' button in the background to the dialog box.

Provide New Placeholder name and click save button, after successful response newly created Placeholder can be found under root folder

This screenshot shows the 'Add New Placeholder' dialog box with the text 'Demo Component' entered into the 'New Placeholder*' input field. The 'Save' button is visible at the bottom of the dialog.

After expanding the root folder, new placeholder can be found.



Please Note: We can also create folder inside the folder using same approach

Create New Manual Test Case

Select placeholder then you will get Manual test case form for development , Placeholder will automatically populated in manual test case form based on selection as well as Cycle will automatically populated based on selected current test cycle from the configuration page

A screenshot of a web application interface for creating a test case. The main heading is "Test Case". Below it is a section titled "NEW TEST CASE DETAILS" with a "Save Test Case" button. The form contains several fields: "Title*" (text input), "Placeholder*" (dropdown menu with "Demo Component" selected and highlighted by a blue box), "Priority*" (dropdown menu with "Medium" selected), "Type*" (dropdown menu with "Functional" selected), "Automation type*" (dropdown menu with "Not Automated" selected), "Cycle*" (dropdown menu with "Demo Sprint 1.1" selected and highlighted by a blue box), and "References" (text input). To the right of the form is a panel titled "TEST COMPONENT STRUCTURE" with a "Create Placeholder" button. It shows a hierarchy: "- Spacecraft", "Template", and "Demo Component" (highlighted with a blue bar).

Add all manual test case attribute from the Manual Test case form and click Save Test Case button

Test Case

NEW TEST CASE DETAILS

Save Test Case

Title*

This is new Test case

Placeholder*

Demo Component

Priority*

Medium

Type*

Functional

Automation type*

Not Automated

Cycle*

Demo Sprint 1.1

References

Preconditions

User should be on home page

Test data

NA

Steps*

1. Verify search bar is displayed on home Page
2. Verify Search icon is displayed on home page

TEST COMPONENT STRUCTURE

Create Placeholder

- Spacecraft

Template

Demo Component

Search Manual test case

Expand the Root folder and select placeholder, after loading the page, all manual test case from this placeholder will be displayed. For selecting root folder all manual test case will be displayed.

TEST CASE DETAILS									
#	Test Id	Test Name	Component	Priority	Type	Automation Type	Cycle	Reference	Created By
	Filter by Test Id	Filter by Test Name	Filter by Component	Filter by Priority	Filter by Type	Filter by Automation Type	Filter by Cycle	Filter by Reference	Filter by Created By
	Enter	Enter Test Name...	Enter Component	Enter Priority...	Enter Type...	Enter Automation Type	Enter Cycle...	Enter Reference	Enter Created By
1	QB-3	This is new manual test case	Demo Component	Medium	Functional	Not Automated	Demo Sprint 1.1		pravesh@qabunch.com
2	QB-1	Template For Pass Test Case	Template	Low	Functional	Automated	Demo sprint infoedge		demouser@qabunch.com
3	QB-2	Template For Fail Test Case	Template	Medium	Functional	Not Automated	Demo sprint infoedge		demouser@qabunch.com

10

1

User can also search test case based on filter e.g. id, name, created By etc.

TEST CASE DETAILS									
	Test Id Filter by Test Id	Test Name Filter by Test Name	Component Filter by Component	Priority Filter by Priority	Type Filter by Type	Automation Type Filter by Automation Type	Cycle Filter by Cycle	Reference Filter by Reference	Created By Filter by Created By
#	Enter	Enter Test Name...	Enter Component	Low	Enter Type...	Enter Automation	Enter Cycle...	Enter Reference	Enter Created
2	QB-1	Template For Pass Test Case	Template	Low	Functional	Automated	Demo sprint infoedge		demouser@qabunch.com

Update Manual test case

Find your manual test case from the test case details table and click the row then you will get the manual test case details dialog form, Update test attribute and click Update button.

Test Case Details

TEST ID : QB-3

UpdateAutomateDelete

Title*

This is new manual test case

Placeholder*

Demo Component

Priority*

Medium

Type*

Functional

Automation type*

Not Automated

Cycle*

Demo Sprint 1.1

References

Preconditions

User should on search result page

Test data

search Item - iPhone

Steps*

1. Verify iPhone item is listed on search Result page
 2. Verify Price is displayed on item card

Expected Result

Comments

History

#

Comment

By

TimeStamp

Comment

Add Comment

From the left navigation panel, expand the "Manual" section and click the "Defect" link.

The screenshot shows the QAautoMATER interface. On the left, the 'Manual' section is expanded, and 'Defect' is selected. The main area is titled 'Defect' and contains a form to 'ADD NEW DEFECT'. The form has the following fields:

- Title* (text input)
- Placeholder* (dropdown menu)
- Priority* (dropdown menu, set to Medium)
- Severity* (dropdown menu, set to Minor)
- Status* (dropdown menu, set to Open)
- Assigned to (dropdown menu)
- Cycle* (dropdown menu, set to Demo Sprint 1.1)

A 'Create new defect' button is located at the top right of the form. On the right side, there is a 'TEST COMPONENT' section with a dropdown menu showing '+ Spacecraft'.

Add all information on defect form and click Create new defect button

The screenshot shows the QAautoMATER interface with the 'ADD NEW DEFECT' form filled out. The form has the following fields:

- Title*: This is new Defect
- Placeholder*: Demo Component
- Priority*: Medium
- Severity*: Minor
- Status*: Open
- Assigned to: pravesh@qaautomater.com
- Cycle*: Demo Sprint 1.1
- Test Id: QB-3

The 'Steps to reproduce' section contains a list:

1. Launch app
2. Logo is missing

Below the steps, there is a 'Screenshot' section showing a thumbnail of the QAautoMATER interface. The thumbnail shows the same 'ADD NEW DEFECT' form with the 'Create new defect' button highlighted.

Search Defect

Expand the Root folder and select placeholder, after loading the page, all defect from this placeholder will be displayed. For selecting root folder all defect will be displayed.

DEFECT DETAILS										
#	Id Filter by Id	Title Filter by Title	Component Filter by Component	Priority Filter by Priority	Severity Filter by Severity	Status Filter by Status	Assign to Filter by Assign to	Cycle Filter by Cycle	Test Id Filter by Test Id	Created By Filter by Created By
	Enter	Enter Title...	Enter Compon	Enter Prior	Enter Seve	Enter Statu	Enter Assig	Enter Cycle	Enter Test I	Enter Creat
1	DF-1	This is new Defect	Demo Component	Medium	Minor	Open	pravesh@qaautomater.com	Demo Sprint 1.1	QB-3	pravesh@qaautomater.com

10

1

Update Defect

Find your Defect from the defect details table and click the row then you will get the defect details dialog form, Update defect attribute and click Update button.

Defect Details

DEFECT ID : DF-1

UpdateDelete

Title*

This is new Defect

Placeholder*

Demo Component

Priority*

Medium

Severity*

Minor

Status*

Open

Assigned to

pravesh@qaautomater.com

Cycle*

Demo Sprint 1.1

Test Id

QB 3

Steps to reproduce*

Normal

B

I

U

1. Launch app

2. Logo is missing

Screenshot

QAautoMATER

#

spacecraft

Manual

Automation

Dashboard

Defect

ADD NEW DEFECT

TEST COMPONENT

Comments

History

#

Comment

By

TimeStamp

Comment

Add Comment

From the left navigation panel, expand the "Manual" section and click the "Test Execution" link.

QAutoMATER Spacecraft

Test Execution

SELECT OR CREATE TEST PLAN New Test Plan Delete Test Plan

Test Cycle* Demo Sprint 1.1 Test Plan*

TEST EXECUTION CONFIGURATION

Create New Test Plan

From the test execution page, click on New Test plan button

Test Execution

SELECT OR CREATE TEST PLAN New Test Plan

Test Cycle* Demo Sprint 1.1 New Test Plan*

TEST EXECUTION CONFIGURATION

Testing type* Functional Screen* Desktop Browser/Device*

OS Release version

TEST CASES Load Test Cases Update Test Plan

	Component Filter by Component	Test Id Filter by Test Id	Filter by Test Name	Assigned To Filter by Assigned To	Status Filter by Status
(+) □ #	Enter Component...	Enter Test Id.	Test Name Enter Test Name...	Enter Assigned To...	Enter Status..

10 ▾

Select test cycle, provide new Test plan name, Provide all test execution configuration after that click on Load Test case button
Select the test case that you want to include in your test plan and click Update test plan button.

SELECT OR CREATE TEST PLAN

New Test Plan

Test Cycle*

Demo Sprint 1.1

New Test Plan*

Demo Test Plan

TEST EXECUTION CONFIGURATION

Testing type*

Functional

Screen*

Desktop

Browser/Device*

Chrome 111.0

OS

Windows11

Release version

1.1

TEST CASES

Load Test Cases

Update Test Plan

			Component Filter by Component	Test Id Filter by Test Id	Filter by Test Name	Assigned To Filter by Assigned To	Status Filter by Status
(+)	<input checked="" type="checkbox"/>	#	Enter Component...	Enter Test Id.	Test Name Enter Test Name...	Enter Assigned To...	Enter Status..
(+)	<input checked="" type="checkbox"/>	1	Demo Component	QB-3	This is new manual test case		Pending
(+)	<input checked="" type="checkbox"/>	2	Template	QB-1	Template For Pass Test Case		Pending
(+)	<input checked="" type="checkbox"/>	3	Template	QB-2	Template For Fail Test Case		Pending

10
1

After creation of test plan, refresh the page and select test plan from the dropdown, all selected test case will be populated

After that, you can update status for adding comment and document expand the test case, add comments and supporting document, and click Update Test Plan button

TEST CASES

Load Test Cases

Update Test Plan

			Component Filter by Component	Test Id Filter by Test Id	Filter by Test Name	Assigned To Filter by Assigned To	Status Filter by Status
(-)	<input checked="" type="checkbox"/>	#	Enter Component...	Enter Test Id.	Test Name Enter Test Name...	Enter Assigned To...	Enter Status..
(-)	<input checked="" type="checkbox"/>	1	Demo Component	QB-3	This is new manual test case		Pass

REMARKS REGARDING STATUS

Normal

Test case executed

Evidence :

Test Execution

SELECT OR CREATE TEST PLAN

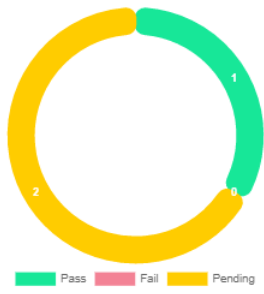
New Test Plan

Delete Test Plan

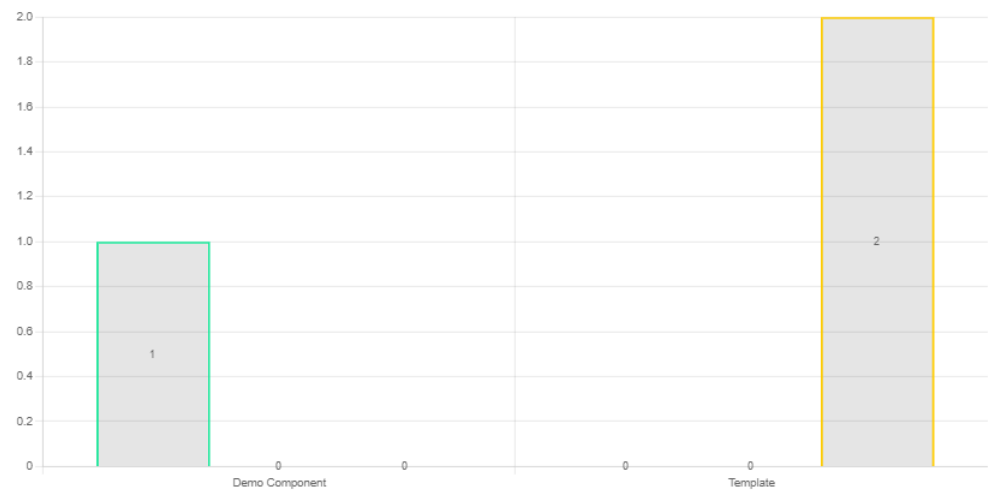
(+)	<input checked="" type="checkbox"/>	2	Template	QB-1	Template For Pass Test Case		Pending
(+)	<input checked="" type="checkbox"/>	3	Template	QB-2	Template For Fail Test Case		Pending

After updating test plan, graph will automatically be updated.

EXECUTION STATUS

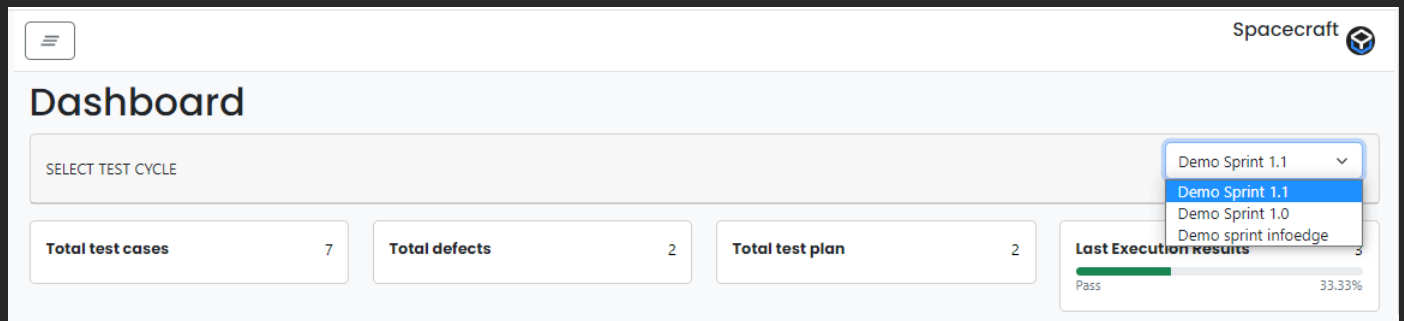


PASS FAIL COUNT MODULE WISE



By default dashboard gives the data for current test cycle, although user can view account and individual progress based on sprint cycle

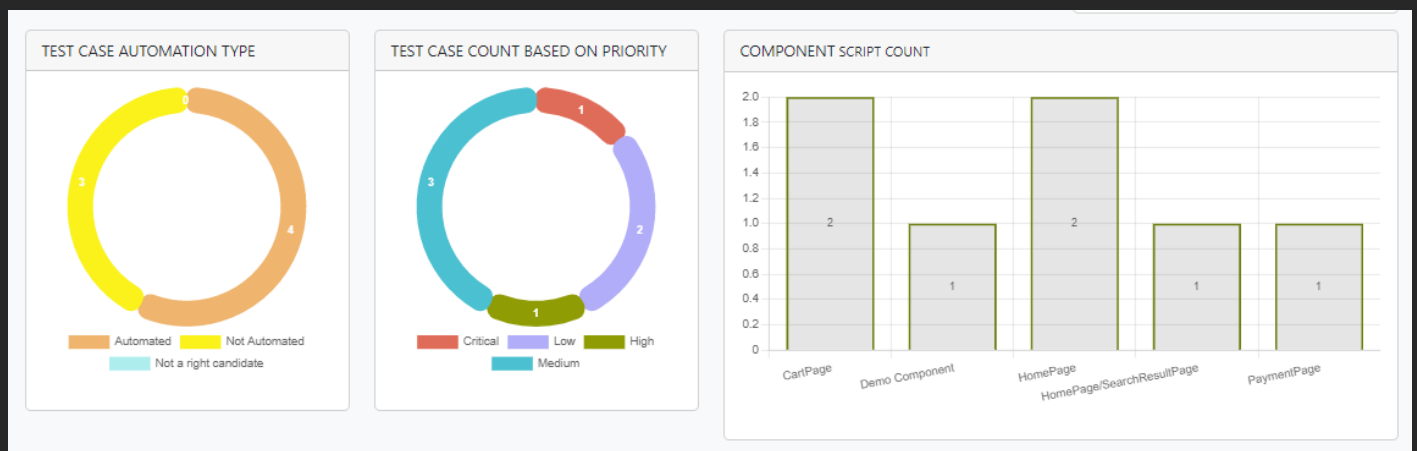
The widget displays data on the number of test cases, defects, and test plans created in the current test cycle, as well as the latest test plan results



The TEST CASE AUTOMATION TYPE section indicates the number of test cases that have been automated in the current test cycle.

The TEST CASE COUNT BASED ON PRIORITY section displays the number of test cases created in the current test cycle, categorized by their priority level.

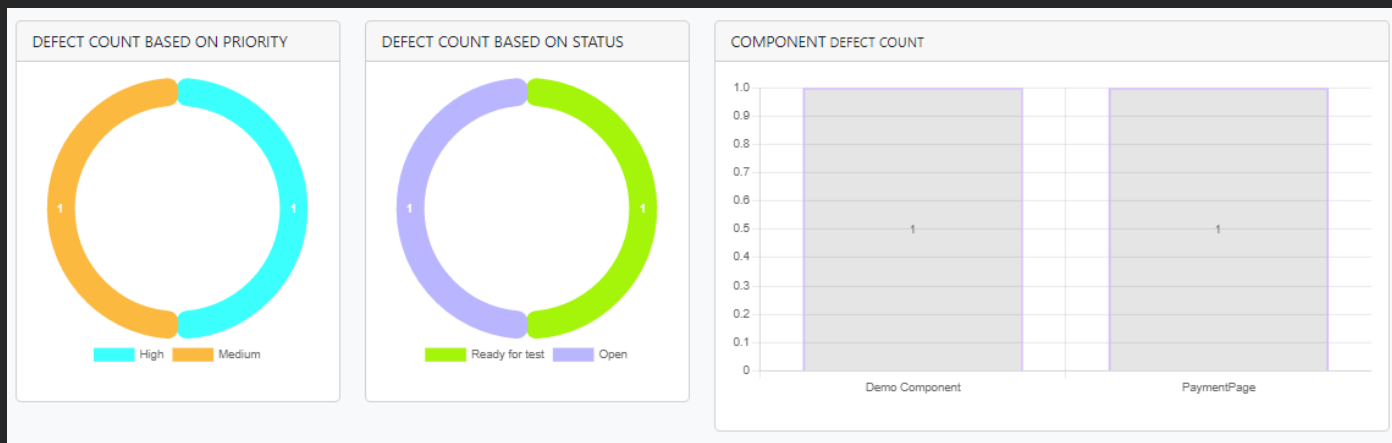
The COMPONENT SCRIPT COUNT section shows the number of test cases associated with each component.



The DEFECT COUNT BASED ON PRIORITY section shows the number of defects identified in the current test cycle, categorized by their priority level.

The DEFECT COUNT BASED ON STATUS section, which displays the status of defects in the current test cycle.

The COMPONENT Defect COUNT section shows the number of Defect associated with each component.



The TEST EXECUTION COUNT section displays the total count of test case executions for all test plans created in the current test cycle

The PASS FAIL PENDING COMPONENT WISE section indicates the pass, fail, and pending counts for each component.

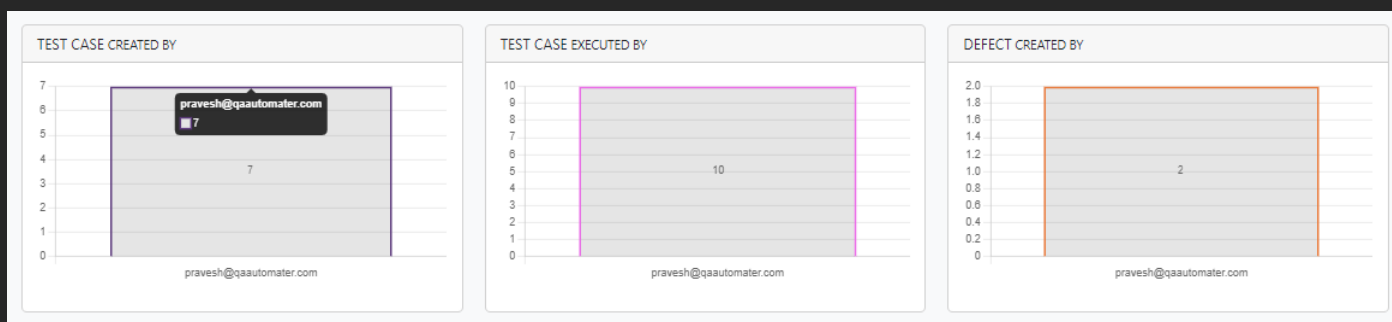
The PASS FAIL PENDING BY TEST PLAN section indicates the number of passed, failed, and pending test cases for each test plan.



The TEST CASES CREATED BY section indicates the number of test cases created by each individual in the current test cycle.

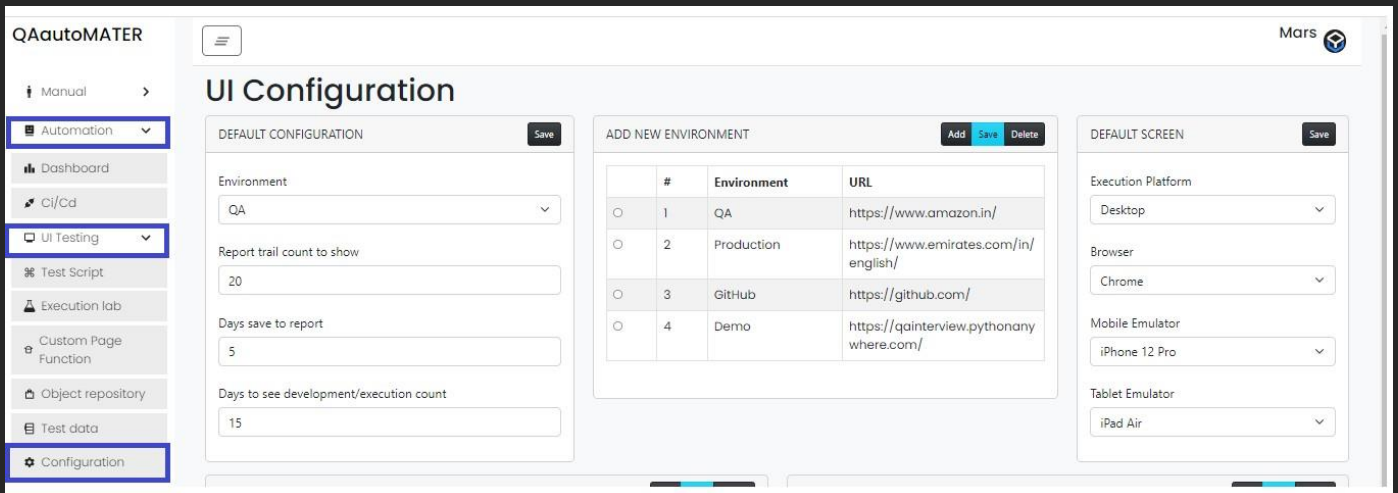
The TEST CASES EXECUTED BY section indicates the number of test cases executed by each individual in the current test cycle.

The DEFECT CREATED BY section indicates the number of defect created by each individual in the current test cycle.



2.1 AUTOMATION – CONFIGURATION

From the left navigation panel, expand the "Automation" section and click the "UI Testing>Configuration" link.



QAautoMATER Mars

UI Configuration

DEFAULT CONFIGURATION Save

Environment: QA

Report trail count to show: 20

Days save to report: 5

Days to see development/execution count: 15

ADD NEW ENVIRONMENT Add Save Delete

	#	Environment	URL
<input type="radio"/>	1	QA	https://www.amazon.in/
<input type="radio"/>	2	Production	https://www.emirates.com/in/english/
<input type="radio"/>	3	GitHub	https://github.com/
<input type="radio"/>	4	Demo	https://qainterview.pythonanywhere.com/

DEFAULT SCREEN Save

Execution Platform: Desktop

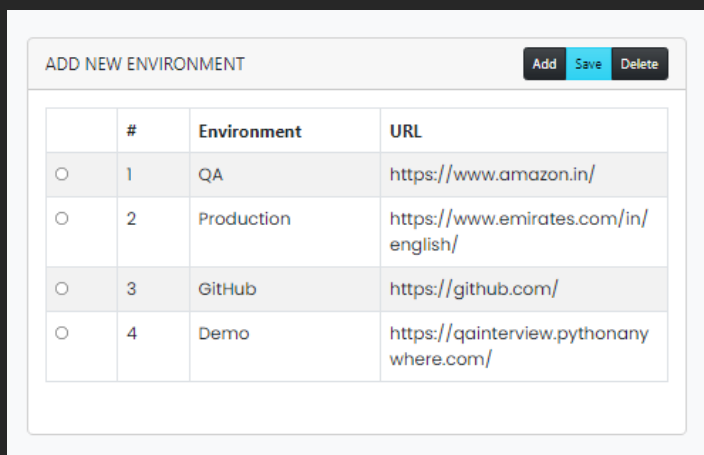
Browser: Chrome

Mobile Emulator: iPhone 12 Pro

Tablet Emulator: iPad Air

Add/Delete Environment

For adding a new Environment, click on the Add button from the 'ADD NEW ENVIRONMENT', provide Environment name and URL details, and click the Save button.



ADD NEW ENVIRONMENT Add Save Delete

	#	Environment	URL
<input type="radio"/>	1	QA	https://www.amazon.in/
<input type="radio"/>	2	Production	https://www.emirates.com/in/english/
<input type="radio"/>	3	GitHub	https://github.com/
<input type="radio"/>	4	Demo	https://qainterview.pythonanywhere.com/

To delete an environment, select the radio button in the environment column, click the delete button, and then click the Save button once the row has been deleted.

ADD NEW ENVIRONMENT

Add

Save

Delete

	#	Environment	URL
<input type="radio"/>	1	QA	https://www.amazon.in/
<input type="radio"/>	2	Production	https://www.emirates.com/in/english/
<input type="radio"/>	3	GitHub	https://github.com/
<input checked="" type="radio"/>	4	Demo	https://qainterview.pythonanywhere.com/

After Clicking Save button you will get success notification and deleted environment will not displayed under ADD NEW ENVIRIONMNE table

Environment and Uri information is successfully saved.

ADD NEW ENVIRONMENT

Add

Save

Delete

	#	Environment	URL
<input type="radio"/>	1	QA	https://www.amazon.in/
<input type="radio"/>	2	Production	https://www.emirates.com/in/english/
<input type="radio"/>	3	GitHub	https://github.com/

DEFAULT SCREEN

Save

Execution Platform

Desktop

Browser

Chrome

Mobile Emulator

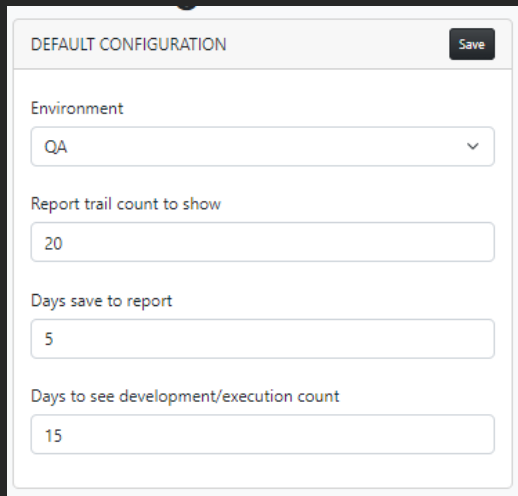
Default Configuration

Environment: You can select a default environment from the 'Default configuration' section, which will be displayed across all pages and will be pre-selected by default

Report Trail count to show: By setting the 'Report Trail' count to 10, for example, the dashboard page will display the execution history for the 10 most recent executions.

Days save to report: By setting, the 'Days to Save' report value to 10, for example, you will be able to see all the details of test executions that were performed within the last 10 days.

Days to see development/execution count: By updating the 'Days to See Development/Execution Count' value, we can view the total count of test script development and execution for a specified number of days on the dashboard page.



DEFAULT CONFIGURATION Save

Environment
QA

Report trail count to show
20

Days save to report
5

Days to see development/execution count
15

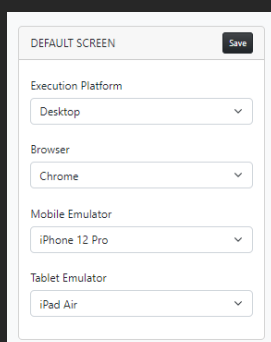
Default Screen:

Execution Platform: this tool allows for 3 execution platforms (desktop, mobile, and tablet), and the execution platform value is updated, then the selected mode for test script execution will be based on the updated platform value. For example, if the execution platform is updated to "mobile," then the test script will be executed in mobile mode. Similarly, if the execution platform is updated to "tablet," then the test script will be executed in tablet mode.

Browser: If a tool allows for selecting a browser for test execution and the browser value is updated, then the updated browser value will be preselected for the test case execution. For example, if the browser value is updated to "Chrome," then the test case will be executed in Chrome browser. Similarly, if the browser value is updated to "Firefox," then the test case will be executed in Firefox browser. The preselected browser value will be used for the test case execution until it is changed again

Mobile Emulator: when the test script is executed on a mobile platform, the mobile screen will be emulated according to the updated emulator value. For example, if the emulator value is updated to "iPhone X," then when the test script is executed on a mobile platform, the mobile screen will be emulated as an iPhone X

Tablet Emulator: when the test script is executed on a tablet platform, the mobile screen will be emulated according to the updated emulator value. For example, if the emulator value is updated to "iPhone X," then when the test script is executed on a mobile platform, the tablet screen will be emulated as an iPad air



DEFAULT SCREEN Save

Execution Platform
Desktop

Browser
Chrome

Mobile Emulator
iPhone 12 Pro

Tablet Emulator
iPad Air

Add/Delete Emulator Screen

The emulator screen in the Chrome browser is a tool that allows you to simulate different mobile and tablet devices with various screen sizes and resolutions. By using this tool, you can test how your website or application looks and functions on different mobile devices without actually owning those devices.

To access the emulator screen in Chrome, you can open the developer tools by pressing F12 or Ctrl+Shift+I (Windows, Linux) or Cmd+Option+I (Mac), then click on the "Toggle device toolbar" icon (or press Ctrl+Shift+M) to open the device mode. From there, you can choose different device presets or customize your own by adjusting the screen size, pixel density, and user agent string.

However, note that the emulator screen is just a simulation and may not fully replicate the behavior of a real mobile device. It's still recommended to test your website or application on actual mobile devices as well.

** Click on Add button select Device type and set Screen name and click Save button

EMULATOR SCREEN

Add

Save

Delete

	#	Device	Screen Name
<input type="radio"/>	1	Mobile	iPhone 12 Pro
<input type="radio"/>	2	Tablet	iPad Air

TEST MANAGEMENT TOOL SETUP

This section will be used for future prospective, if client has external test management tool then we can integrate with QAutoMATER.

TEST MANAGEMENT TOOL SETUP

Add

Save

Delete

	#	Tool	Url	Username	Password
--	---	------	-----	----------	----------

CAPABILITY SETUP:

You can configure the desired capabilities for different execution platforms such as desktop, mobile, and tablet for Selenium grid or other cloud provider server by updating the Hub URL and setting the desired capabilities for each platform. These capabilities can include the operating system, browser version, maximum parallel instances, and other settings that are required for the setup of the test environment. By configuring the desired capabilities, you can ensure that your tests run on the correct platforms with the required settings, which can help improve the reliability and efficiency of your test execution.

CAPABILITY SETUP

Save

Execution Server*

LamdaTest

Hub Url*

https://huburl

Capabilities*

⊖ "" : { 3 items

⊖ "Desktop" : { 4 items

⊖ "Chrome" : { 5 items

"setAcceptInsecureCerts" : true

"setAlertBehavior" : NULL

"setBrowserVersion" : NULL

"setPlatform" : NULL

"setProxy" : NULL

}

⊕ "Firefox" : {...} 5 items

⊕ "Edge" : {...} 5 items

⊕ "Safari" : {...} 5 items

}

⊕ "Mobile" : {...} 5 items

⊕ "Tablet" : {...} 5 items

}

Rename-Delete test component

This section can only be accessed by an admin user. Using this feature, you can delete unused components along with their associated test scripts, as well as rename components in case the wrong name was mistakenly used during the creation of a test script

To rename a component, select the desired component from the tree structure, provide the new name, and then click the Rename button

RENAME/DELETE TEST COMPONENT

RenameDelete

New Name*

newName

- Spacecraft

AgileAllLabs

OldComponent

To delete a component, select the desired component from the tree structure, click the 'Delete' button, and all associated test cases inside will be deleted.

RENAME/DELETE TEST COMPONENT

RenameDelete

New Name*

- Spacecraft

AgileAllLabs

OldComponent

Move your test cases

This feature is used to move test cases from component A to component B.

To move test cases, please follow these steps: Select the source component, choose the test case(s) you wish to move, select the destination component, and then click the 'Save' button

MOVE YOUR TEST CASES

Save

Component*

- Spacecraft

AgileAllLabs

OldComponent

Select Test to move

QB-12 x

New Component*

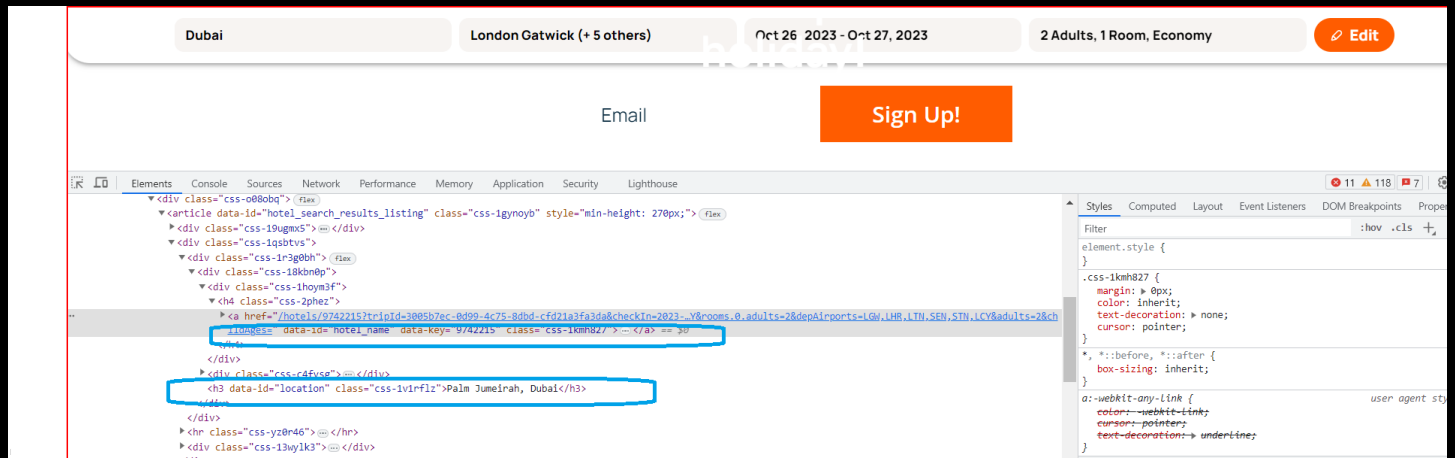
- Spacecraft

AgileAllLabs

OldComponent

Add your Custom Web locator for Web element Identification

This feature allows the addition of custom locators for web element identification. Since most applications are built using Angular and React, elements can be identified using various locators, such as data-id and data-name. We can add these strategies to identify elements.



ADD YOUR LOCATOR FOR WEB ELEMENT IDENTIFICATION

Add Save Delete

#	Locator
<input type="radio"/> 1	id
<input type="radio"/> 2	name
<input type="radio"/> 3	xpath
<input type="radio"/> 4	linktext
<input type="radio"/> 5	partiallinktext
<input type="radio"/> 6	class
<input type="radio"/> 7	cssselector
<input type="radio"/> 8	data-id

2.2 AUTOMATION – Object Repository Page

The Object Repository page is used to add new web elements as well as update existing locator properties.

QAautoMATER AI predicts web element properties, and we can also customize the prediction algorithm after updating the web element tag

QAautoMATER

Manual

Automation

Dashboard

Ci/Cd

UI Testing

Test Script

Execution lab

Custom Page Function

Object repository

Test data

Configuration

Api Testing

Mobile Testing

Help

Spacecraft

Object Repository

UI ELEMENT ATTRIBUTE

Add

Save

Delete

#	Name	Locator	Locator Property
1	INPUTSEARCHBAR	id	searchUnit

10

1

SET UP TAG FOR AUTOMATIC WEB ELEMENT CREATION

Save

#	Webelement type	Tag
1	LINK	//*[@self::a/div or self::a or self::button]
2	BUTTON	//*[@self::input or self::button or self::label]
3	TEXTBOX	//input
4	IMAGE	//*[@local-name()='svg' or self::img]
5	CHECKBOX	//input
6	RADIOBUTTON	//input
7	LISTBOX	//*[@self::select or self::p/span]
8	TEXTAREA	//textarea
9	LABEL	//label
10	DEFAULT	//*

Copyright © QAautoMATER 2022 . Any illegal reproduction of this content will result in immediate legal action

2.3 AUTOMATION – HOW TO CREATE AUTOMATION TEST SCRIPT

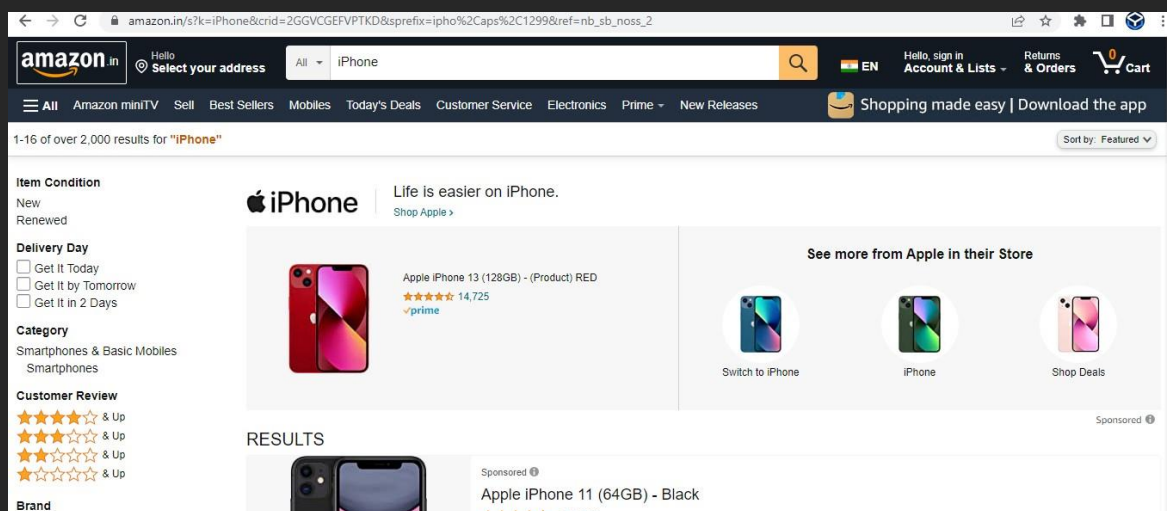
While creating new test script please focus on given below steps

1. Identifying reusable steps involves looking at the overall testing process and identifying any steps that can be reused across multiple test cases. This can help to save time and effort in creating new test cases and can ensure consistency in testing.
2. Identify the page or feature for which test script will be written.

For example, I need to write a 2-test case for the Search Result page in the Amazon application

TC-1 – for valid search result count should be greater than 0

TC-2 – For invalid test case result count should be zero



For instance, if we create a reusable method called `searchItemOnAmazon`, we can use it to test the Search Result page in the Amazon application across different test scripts, making our testing process more efficient and effective

Reusable Function:

- a. Navigate to Custom Page Function through navigation Automation-> UI Testing > Custom Page Function.
- b. Select Function from Reusable dropdown and add your method name, here I am adding method '`searchItemOnAmazon`'

- c. Click on Add Step button

ADD REUSABLE TEST STEPS						
#	Step Definition*	Action*	Web Element	Value	IsReportingRequired	
1						

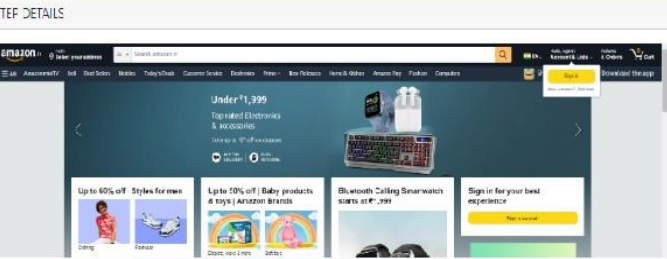
- d. Add Step definition, here I am writing 'Search item in Search Bar', once you done with Step definition write 80-90% times Action and web element will Auto generated.

ADD REUSABLE TEST STEPS						
#	Step Definition*	Action*	Web Element	Value	IsReportingRequired	
1	Search item in Search Bar	Type	INPUTSEARCH	iPhone		

- e. In value section, provide any value what we want to provide search item.
 f. To make sure your automated step is working fine, click on the Debugger window from the Debug Custom Function section

Please watch your debug action

STEP DETAILS



DEBUG CUSTOM FUNCTION

Debugger Window
Debug step
Debug page function


Application Uri*
https://www.amazon.in/

Screen*
Desktop

Device/Browser*
Chrome

- g. Select the radio button for test step and Click Debug step button

Please watch your debug action



3 items
"Step": "Search item in Search Bar"
"Status": "Pass"
"Message": ""

DEBUG CUSTOM FUNCTION

Debugger Window
Debug step
Debug page function

Application Uri*
https://www.amazon.in/

Screen*
Desktop

Device/Browser*
Chrome

Custom Page Function
Object repository
Test data
Configuration

ADD REUSABLE TEST STEPS

#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
1	Search item in Search Bar	Type	INPUTSEARCH	iPhone	

- h. If you are getting Status Fail then update the Locator, Click the Web Element column from selected Row, and update locator property. After updating the Property Click Debug Step and make sure status is PASS

Add or Select existing Element

Element Name*

Locator*

Locator Property*

Device/Browser*

Action* **Web Element**

Type

- i. Parameter the test data, in value section write ARGs.searchItem (ARGs. Is the mandatory keyword to parameterizing the value)
- j. Similarly, we can write different steps, here we are writing another steps 'Click on the Search Icon'

ADD REUSABLE TEST STEPS

Select Pre Dependent Page

#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
1	Search item in searchbar	Type	INPUTSEARCH	{ "valuetobesend": "ARGs.SEARCHITEM" }	
2	Click on the Search Icon	Click	IMAGESEARCH		

h. Close the debugger window and Click on the Save button

Coming to our example scenario 2 scenario are for search result page so if we can create Page function for search Result page so every test script can consume same page function, which is written for search result page.

Page Function:

- a. Navigate to Custom Page Function through navigation Automation-> UI Testing > Custom Page Function.
- b. Select Page from Reusable dropdown and add your page name, here I am adding page name 'Given I am on Search Result Page'

Custom Page Function

CUSTOM PAGE/FUNCTION

Reusable

Page/Function Name

DEBUG CUSTOM FUNCTION

Application Uri*

Screen*

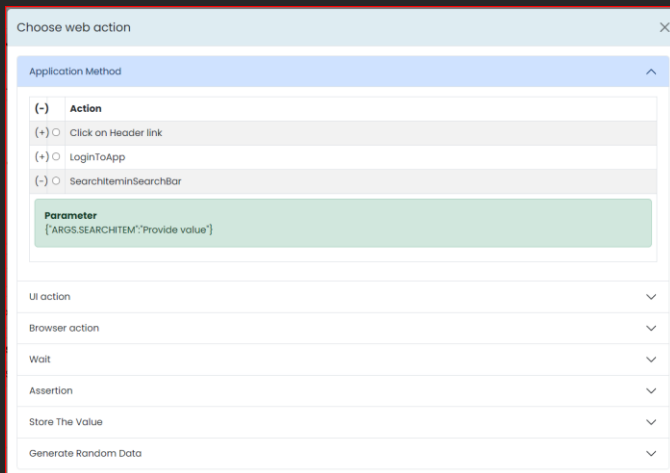
Device/Browser*

ADD REUSABLE TEST STEPS

Select Pre Dependent Page

#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
10					

- c. Click on the Add Step button and click on Action column - Choose Web Action dialog will be displayed. From this dialog window, go to application method section and find method searchItemINearchBar



After selecting radio button , method will be displayed in action section

#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
1	Search Scenario	SearchItemINearchBar		{\"ARGS.SEARCHITEM\": \"Provide value\"}	

We can also paramaterize the page label function

#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
1	Search Scenario	SearchItemINearchBar		{\"ARGS.SEARCHITEM\": \"ARGS.SEARCHITEM\"}	

- d. Add more step if required to reaching respective page, also debug your step to make sure your page function is also working.
- e. Click on Save button

New Test Script:

After creation of reusable method and page function

- a. Navigate to UI Test script page through navigation Automation-> UI Testing > Test Script

UI Test Script

Paste your manual test case

BASIC DETAILS

Save Delete Rename

Component

Test Id

Test Name

DEBUG TEST SCRIPT

Debugger Window Debug step Debug page function

Application Uri*

Screen*

Device/Browser*

AUTOMATE MANUAL TEST STEPS

Add Step Delete Step

Select Pre-Dependent Page

#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
10					

- b. Provide basic details Component, Test ID and Test name from the basic details section

BASIC DETAILS

Save

Delete

Rename

Component

SearchResultPage

Test Id

TC-1

Test Name

Search for Valid search Item

- c. Expand Pre-Dependent Page section and click Add button

Select Pre-Dependent Page

PAGE FUNCTION

Add

Delete

	#	Page Function	Parameter
<input type="radio"/>	1		

- d. Select existing Page function and update the Parameter value

Choose Manual Test Case

BASIC DETAILS

Save

Delete

Rename

Component

Search

Test Id

TC-100

Test Name

New Search

DEBUG TEST SCRIPT

Debugger Window

Debug step

Debug page function

Application Url*

https://www.dmart.in/

Screen*

Desktop

Device/Browser*

Chrome

AUTOMATE MANUAL TEST STEPS

Add Step

Delete Step

Select Pre-Dependent Page

PAGE FUNCTION

Add

Delete

	#	Page Function	Parameter
<input type="radio"/>	1	Given I am on the Search Result Page	{*ARGS.SEARCHITEM:"Provide value"}

Update the param value

PAGE FUNCTION

Add

Delete

	#	Page Function	Parameter
<input type="radio"/>	1	Given I am on the Search Result Page	{*ARGS.SEARCHITEM:"iPhone"}

- e. Now click on Add step button, Add step definition, select Appropriate action, element and Value

	#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
<input type="radio"/>	1	Verify Product Count	AssertElementCountGreater Than	LINKPRODUCT	0	

10

- f. Create all step in similar way make sure all step is working find by checking debug functionality and save the test script.

Test script is successfully saved.

UI Test Script

Paste your manual test case

BASIC DETAILS

Save

Delete

Rename

Component

SearchResultPage

Test Id

TC-1

Test Name

Search for Valid search Item

DEBUG TEST SCRIPT

Debugger Window

Debug step

Debug page function

Application Url*

https://www.amazon.in/

Screen*

Desktop

Device/Browser*

Chrome

AUTOMATE MANUAL TEST STEPS

Add Step

Delete Step

Select Pre-Dependent Page

	#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
<input type="radio"/>	1	Verify Product Count	AssertElementCountGreater Than	LINKPRODUCT	0	

Common test data (c.)

Dynamic Test Data (Utility.)

Test specific data (t.)

Common Test Data:

When multiple test scripts use the same test data, it is a good practice to add this data to the 'Test Data' page under the 'UI Testing' section.

QAautoMATER

Manual >

Automation

Dashboard

CI/Cd

UI Testing

Test Script

Execution lab

Custom Page Function

Object repository

Test data

Configuration

Common Test Data

ADD OR UPDATE COMMON TEST DATA

Add Save Delete

	#	Filter by Data Key Data Key Search data key	Filter by Value Value Search data
<input type="radio"/>	1	USERNAME	demoUser
<input type="radio"/>	2	SEARCHPRODUCT	QAautoMATER

10

How to Consume test data in Test script/Custom Function:

After adding the common test data, this test data can be directly use in test Scripts and custom Page function by calling "c.dataKeyName". Here is the example c.USERNAME

Select Pre-Dependent Page

#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
<input type="radio"/>	1	Provide userName	Type	INPUTUSERNAME	c.USERNAME

10

Add or Select Test Data

ADD OR SELECT DATA

Utility Data

COMMON DATA SET

Add Delete

#	Data Key Filter by Data Key Search data key	Filter by Value Value Search data
<input type="radio"/>	1	TEST
<input type="radio"/>	2	QA
<input checked="" type="radio"/>	3	USERNAME
<input type="radio"/>	4	SEARCHPRODUCT

Add column

Add Column Delete Column Add Row Delete Row

From the UI test script page, new Common test data can also be added from the ADD OR SELECT DATA section.

ADD OR SELECT DATA

COMMON DATA SET

AddDelete

		Data Key Filter by Data Key	Filter by Value
	#	Search data key	Value Search data
<input type="radio"/>	1	TEST	QA
<input type="radio"/>	2	QA	QA
<input type="radio"/>	3	USERNAME	demoUser
<input type="radio"/>	4	SEARCHPRODUCT	QAautoMATER
<input checked="" type="radio"/>	5	MYNEWDATAKEY	added from the Test script page

10

1

Dynamic Data:

QAAutoMATER supports dynamic test data, such as emails, random strings, and phone numbers. You can generate this test data and use it in your test scripts or custom function pages, as well as for arithmetic operations such as addition, subtraction, multiplication, and percentage calculation. Additionally, you can use string functions such as splitting, among others.

Example:

Navigate to Test Script page or Custom Function page and Expand Add or Select Test Data

AUTOMATE MANUAL TEST STEPS

Add StepDelete Step

Select Pre-Dependent Page

#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
1	Provide random data	Type	INPUTSEARCHICON		

10

1

Add or Select Test Data

ADD OR SELECT DATA

COMMON DATA SET

AddDelete

Add column

AddDeleteAddDelete

ColumnColumnRowRow

Utility Data

Select test step and click Utility Data button

Dynamic Data :

CHOOSE DATA

Evaluate Update

Data Key

Filter by Data Key

Search

Parameter

<input type="radio"/>	1	firstName	
<input type="radio"/>	2	lastName	
<input type="radio"/>	3	address	
<input type="radio"/>	4	email	
<input type="radio"/>	5	uuid	
<input type="radio"/>	6	country	
<input type="radio"/>	7	state	
<input type="radio"/>	8	city	
<input type="radio"/>	9	zipCode	
<input type="radio"/>	10	phoneNumber	

10

123>

Select data key as per test data requirement and click Evaluate button to make sure data is generated in correct way e.g.

Dynamic Data :

Brett.Hahn73@yahoo.com

CHOOSE DATA

Evaluate Update

Data Key

Filter by Data Key

Search

Parameter

<input type="radio"/>	1	firstName	
<input type="radio"/>	2	lastName	
<input type="radio"/>	3	address	
<input checked="" type="radio"/>	4	email	
<input type="radio"/>	5	uuid	

After successful evaluation Click on the Update button, Dynamic data will be added in test step, starting with Utility.

AUTOMATE MANUAL TEST STEPS

Add Step Delete Step

Select Pre-Dependent Page

#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
<input checked="" type="radio"/>	1	Provide any random data in search bar	Type	INPUTSEARCH	Utility.firstName

Please Note : you can add multiple dynamic data in single step e.g. Utility.firstname+Utility.lastName+@qabunch.com

Test Specific Data:

Test data, which is only applicable for Test script, such kind of test data can be added from the test script page. On the UI test script page, expand the section 'Add or Select Test data' section.

Add or Select Test Data

ADD OR SELECT DATA

Utility Data

COMMON DATA SET
Add
Delete

#	Data Key Filter by Data Key Search data key	Filter by Value Value Search data
<input type="radio"/> 1	USERNAME	demoUser
<input type="radio"/> 2	SEARCHPRODUCT	QAautomated

Add column

Add Column
Delete Column
Add Row
Delete Row

Provide Column or test data logical name in Add column Edit box and click Add Column button

PASSWORD
Add Column
Delete Column
Add Row
Delete Row

#	USERNAME	PASSWORD
---	----------	----------

Click on Add Row button and Provide Test data value

PASSWORD
Add Column
Delete Column
Add Row
Delete Row

#	USERNAME	PASSWORD
<input type="radio"/> 1	testuser@test.com	qaaaa

If you want to run your test script on multiple iteration, add another row with data.

PASSWORD
Add Column
Delete Column
Add Row
Delete Row

#	USERNAME	PASSWORD
<input type="radio"/> 1	testuser@test.com	qaaaa
<input type="radio"/> 2	newuser@test.com	new

2.5 OR – LOCATOR PARAMETERIZATION

Locator property can be parameterize using test data (c., t.Utility.), argument data (ARGS.). and Runtime data (Session.).

Example:

Scenario 1 : In the Amazon application, there are many web links, and you want to click on a link without having to add a locator for each one. You would like to simply provide the text of the link, and have the web element created automatically.

Solution: Go to the Custom Page function, add any reusable method name and click Add Step button

Custom Page Function

CUSTOM PAGE/FUNCTION Save Delete Rename

Reusable Function

Page/Function Name ClickOnLink

DEBUG CUSTOM FUNCTION Debugger Window Debug step Debug page function

Application Uri* https://www.amazon.in/

Screen* Desktop

Device/Browser* Chrome

ADD REUSABLE TEST STEPS Add Step Delete Step Utility Data

#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
1	Click on Amazon Web link	Click	LINKAMAZONWEB		

10

Click on Web Element Column for Step

Add or Select existing Element

Element Name* LINKAMAZONWEB

Locator* Xpath

Locator Property* //a[contains(text(),' + ARGS.LINKNAME + ')]

Add Remove

Parameterize the Xpath so here xpath is `//a[contains(text(),' + ARGS.LINKNAME + ')]`,

- Parameter will always started with + and end with +

Click on Add button and value selection Provide the Same Name which is used in Xpath `ARGS.LINKNAME`

#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
1	Click on Header link	Click	LINKHEADER	{ "ARGS.LINKNAME": "ARGS.LINKNAME" }	

After successful creation of reusable method, go to the test script page and select this reusable method from action dropdown

	#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
<input type="radio"/>	1	Click On Header Link	Click on Header link	LINKHEADER	{ "ARGS.LINKNAME": "Provide value" }	

Now call this reusable method for different link

	#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
<input type="radio"/>	1	Click On MinTV	Click on Header link	LINKHEADER	{ "ARGS.LINKNAME": "miniTV" }	
<input type="radio"/>	2	Click on MOBILE Link	Click on Header link	LINKHEADER	{ "ARGS.LINKNAME": "MOBILE" }	

Scenario 2: When you type something in the Amazon search bar, the auto-populated item with the same text must be clicked

Solution: Go to the Test script page Perform search action and provide test data

AUTOMATE MANUAL TEST STEPS

Add StepDelete Step

Select Pre-Dependent Page

	#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
<input type="radio"/>	1	Search item in searchbar	Type	TEXTBOXSEARCH	{ "valuetobesend": "t.SEARCHITEM" }	

10

1

Add or Select Test Data

ADD OR SELECT DATA

Utility Data

COMMON DATA SET

AddDelete

#	Data Key	Filter by Data Key	Filter by Value
	Search data key		Value Search data

searchitem

AddDeleteAddDelete

ColumnColumnRowRow

#	SEARCHITEM
<input type="radio"/>	1miniTV

Add Step for Auto populated selection and click Web element column and parameterize the Xpath

`//div[contains(@class,'suggestion')]/div[contains(text(),'+t.SEARCHITEM+')`

Add or Select existing Element

Element Name*

LISTSEARCHITEM

Locator*

Xpath

Locator Property*

//div[contains(@class,'suggestion

Add

Remove

2.6 SAVE SESSION VARIABLE OR HANDLE DATA ON RUN TIMES

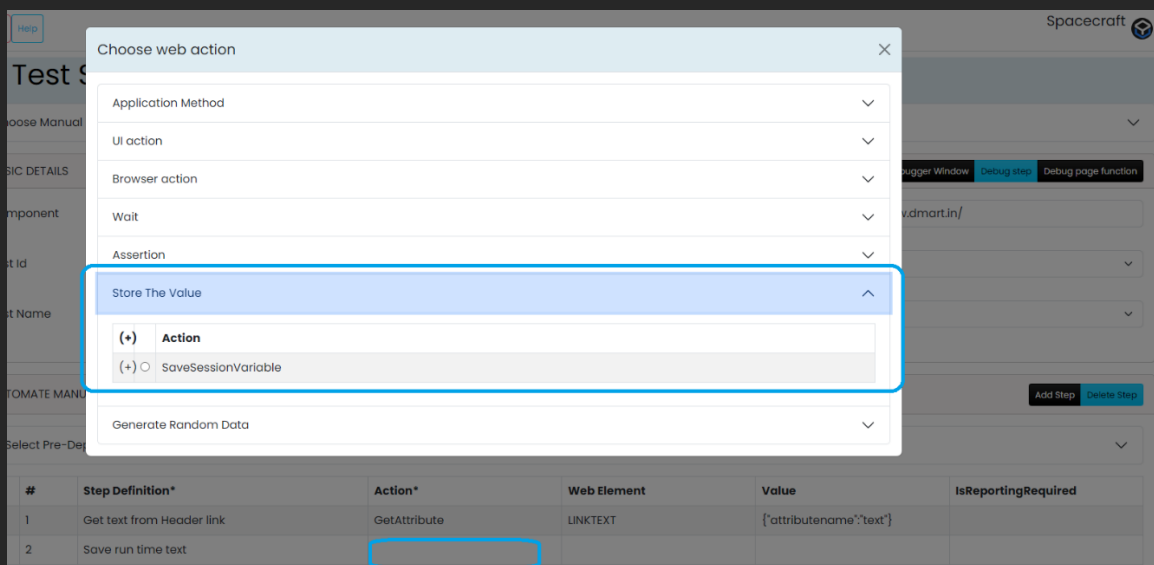
QAAutoMATER provides support for capturing runtime data, saving it, and using it across different steps. This data can also be used to parameterize locator properties.

Example:

Scenario 1 : In the Amazon web application, you have to get product price on search result page and verify same product price on product information page

Solution:

Action "SaveSessionVariable" which captures the value of the step that was just executed and saves it in a variable with a specified name



#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
1	Get text from Header link	GetAttribute	LINKTEXT	{'attributename':"text"}	
2	Save run time text	SaveSessionVariable		{'variablename':"headerlinktext"}	
3	Provide City	Type	CITY	{'valuetosend':"Session.headerlinktext" }	

Please Note: the scope of variable is on test scripts only; you cannot use this variable on another scripts.

2.7 Action for Similar Elements

Scenario:

In the header section of the HTML document, there are multiple links with identical DOM structures, except for the text content

DMart Grocery Baby & Kids Beverages Dals & Pulses Masala & Spices Detergent & Fabric Care Cleaners Freshener & Repellents Pet Supplies

Example:

Element logical Name : linkHeader (xpath = "//ul[contains(@class,'categories-header')]/a")



We can perform click or any action based on index or any other property, We have to pass attribute name or index on value section as JSON object

Example 1 : Based on index

#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
1	Click on Header link	Click	LINKHEADER	{ "index":0 }	

Example 2 : Based on attribute

#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
1	Click on Header link	Click	LINKHEADER	{ "index":0 }	
2	Click on Header link	Click	LINKHEADER	{ "text": "Baby & Kids" }	

Example 3 : combination of multiple attribute e.g. you want to click on header link which href = abc and class has =xyz

#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
1	Click on Header link	Click	LINKHEADER	{ "index":0 }	
2	Click on Header link	Click	LINKHEADER	{ "text": "Baby & Kids" }	
3	Click on Header link	Click	LINKHEADER	{ "href": "abc", "class": "xyz" }	

Example 3 : combination of multiple attribute e.g. you want to click on header link which href = abc or class has =xyz

#	Step Definition*	Action*	Web Element	Value	IsReportingRequired
1	Click on Header link	Click	LINKHEADER	{ "index":0 }	
2	Click on Header link	Click	LINKHEADER	{ "text": "Baby & Kids" }	
3	Click on Header link	Click	LINKHEADER	{ "href": "abc", "class": "xyz" }	
4	click on header link	Click	LINKHEADER	{ "href": "abc", "class": "xyz", "oroperation": true }	

