

Quality Assurance - Test Automation Assessment - English - Affinipay

Objective

The objective of this assessment is to validate your test automation abilities, your knowledge on programming and best practices for automation. We expect to see a full project alongside with the instructions on how to run the tests – all committed in a code repository like GitHub, Bitbucket or Gitlab.

This project should be a showcase of your test automation knowledge – having the test cases automated and working (passing or failing according to the expected result) is crucial, but other aspects are going to be considered as well: how the project is organized, quality of selectors, usage of test automation design patterns and quality of the code are some of the aspects we will evaluate. The overall quality of your project is going to tell us what is your expertise level and if you fit in the position we have interviewed you for (or may interview you in the future).

Some tips:

- Follow best practices. The entire solution can be coded without the usage of any automation anti-pattern.
- Be mindful of your time, make sure that what you deliver is complete and functional.
- Remember to have all your tests working properly. This is the most important aspect of the assessment.

Development

- The test cases should be automated using **Python** or **Ruby** as programming language along with **Playwright** or **Selenium**.
- Feel free to use any other libraries to help with the development.
- **Please do not use low code frameworks/tools (like Robot Framework)**. We want to see how you code and how you organize your code.
- Every test case has a Data Table:
 - For tests with more than one data entry, the test should be executed once for each entry.
- Make sure everything is written in **English**.
- **Important:** Not all test cases listed should pass. What really matters is that they pass or fail according to the manual test case.

Evaluation

The code will be evaluated using the following criteria:

- The automated test cases results should match what the manual test case execution results would be.
- Code quality – proper indentation, comments where needed, no duplication, etc.
- Good programming and test automation best practices.
- Project structure.

Delivery

- Commit the code into GitHub, Gitlab or any other public source code repository:
 - It can be in your personal account or a new account
 - Remember to exclude from the commits the project files of your IDE
- Send an email to the HR person that you are in contact with. Make sure to include the following items in the message:
 - Full name
 - Link for the project
 - Any other additional information that you deem necessary
- Deadline: 3 days.

Important!

In case the website to be automated goes down, please send an email to renata.rocha@e-core.com and the HR person that contacted you telling what happened. We will investigate and return to you ASAP.

Test cases

TC001 - Login (Positive)

Goal

Validate if the user can authenticate in the application with the credentials provided

Data Table

Iteration	Data	Value
1	Username	demouser
	Password	abc123

Pre-Condition

- Have the main page loaded (<https://automation-sandbox-python-mpywqjbdza-uc.a.run.app>)

Steps

Step	Action	Expected
1	Fill the following fields and click the button Login : <ul style="list-style-type: none">• Email: @Username• Password: @Password	The application should redirect the user to the page Invoice List .

TC002 - Login (Negative)

Goal

Validate that the application denies the user login with invalid credentials:

Data Table

Iteration	Data	Value
1	Username	Demouser
	Password	abc123
2	Username	demouser_
	Password	xyz
3	Username	demouser
	Password	nananana
4	Username	demouser
	Password	abc123

Pre-condition

- Have the main page loaded (<https://automation-sandbox-python-mpywqjbdza-uc.a.run.app>)

Steps

Step	Action	Expected
1	Fill the following fields and click the button Login : <ul style="list-style-type: none">• Email: @Username• Password: @Password	The application shows the message: Wrong username or password

TC003 - Validate invoice details

Objective

Validate the invoice information is presented:

Data Table

Iteration	Data	Value
1	Username	demouser
	Password	abc123
	HotelName	Rendezvous Hotel
	InvoiceDate	14/01/2018
	DueDate	15/01/2018

InvoiceNumber	110
BookingCode	0875
CustomerDetails	JOHNY SMITH R2, AVENUE DU MAROC 123456
Room	Superior Double
CheckIn	14/01/2018
CheckOut	15/01/2018
TotalStayCount	1
TotalStayAmount	\$150
DepositNow	USD \$20.90
Tax&VAT	USD \$19.00
TotalAmount	USD \$209.00

Pre-condition

- Have a successful login with the credentials provided (<https://automation-sandbox-python-mpywqjbdza-uc.a.run.app>)
- Page "Invoice List" is shown

Steps

Step	Action	Expected
1	Click the Invoice Details link for the first item presented in the screen.	The application opens the Invoice Details screen.
2	Validate the information presented: <ul style="list-style-type: none">• Hotel Name: @HotelName• Invoice Date: @InvoiceDate• Due Date: @DueDate• Invoice Number: @InvoiceNumber• Booking Code: @BookingCode• Customer Details: @CustomerDetails• Room: @Room• Check In: @CheckIn• Check Out: @CheckOut• Total Stay Count: @TotalStayCount• Total Stay Amount: @TotalStayAmount• Deposit Now: @DepositNow• Tax & VAT: @Tax&VAT• Total Amount: @TotalAmount	The information in the screen matches the one provided in the data table.