



ArtComputer

DESIGNER User Manual

Automated Test with the Robot

Designer User Manual

Contents

Foreword	4
Working with Proxy	5
Login	7
User Interface	7
User interface for the editing	8
User Interface	8
Control Panel	9
User Interface	9
Scenario	10
User Interface	10
Scenario Edit	11
Scenario Import.....	12
Test.....	13
Test Edit	18
Import Test.....	20
Logfile.....	21
Gallery	22
Suite Set	23
User Interface	23
Suite Set Edit	24
Logfile.....	25
Gallery	26
Story Set	27
Story set	27
Story set Edit	28
Stories	29
Stories Edit	30
Logfile.....	31
Gallery	32

Reference	33
Reference.....	33
Reference Edit.....	34
Export Reference	35
Import Reference	36
Dictionary	37
Dictionary set	37
Dictionary set Edit.....	38
Dictionary word	39
Dictionary word Edit	40
Dictionary unused word.....	41
Rules	42
Rules set.....	42
Rules set Edit.....	43
Rules.....	44
Rules Edit.....	45
Import rule set (including rules)	46
Loop	47
Principle	47
Break a loop	48
End of a loop	49
Examples of loop.....	50
Performance.....	52
Principle	52
Define the measure	53
Mechanism.....	54
Performance with the Robot	56
Performance with Admin right	60
Example of graphs of performance designed in Excel.....	61

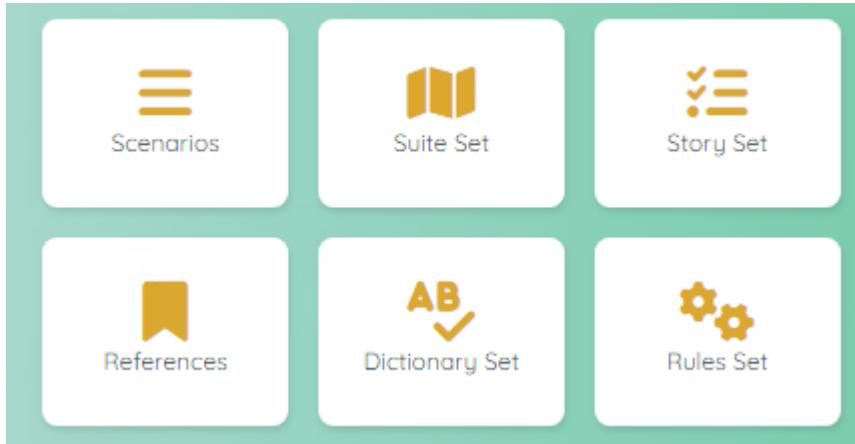
Foreword

This user manual will give you information on how to use the robot as a Designer.

Designer:

The Designer is responsible for:

- The design of the Scenarios
- The design of the Suites
- The design of the Stories
- The management of the References
- The management of the Dictionary
- The design of the Rules

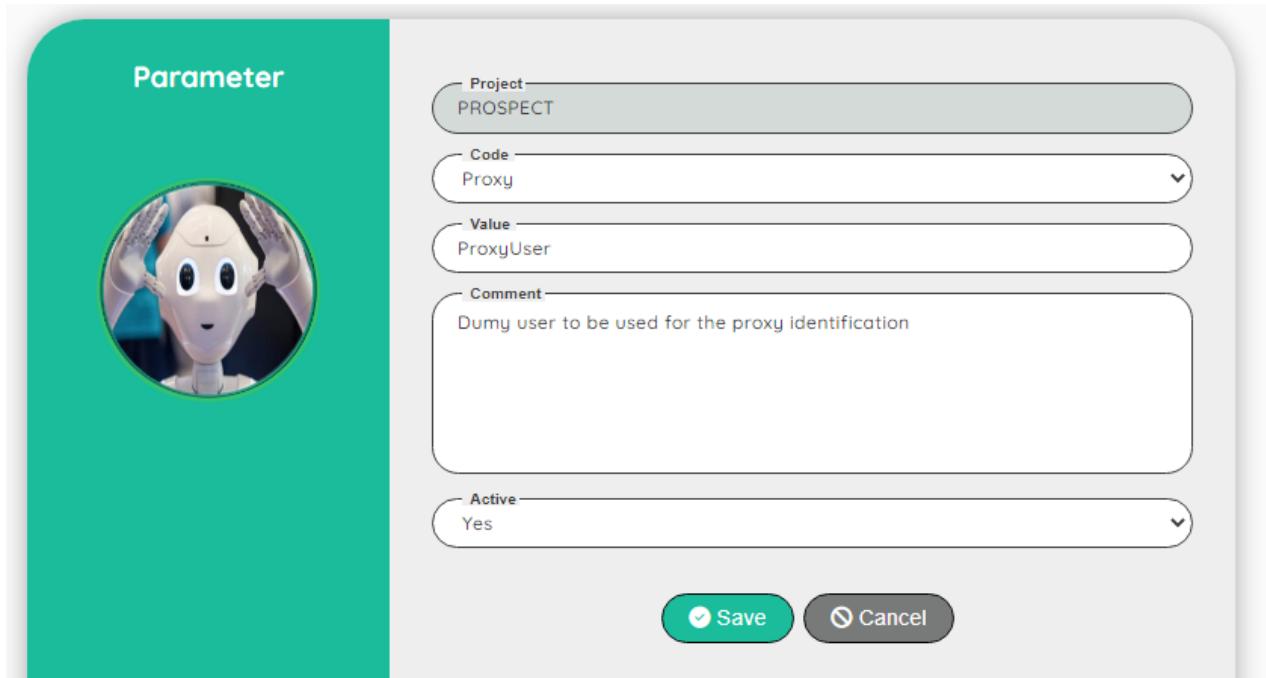


Working with Proxy

If you need to test a project that requires a proxy, please use the following method.

Job to be performed by the Administrator:

In the parameters of the project, create a new entry for the parameter Proxy

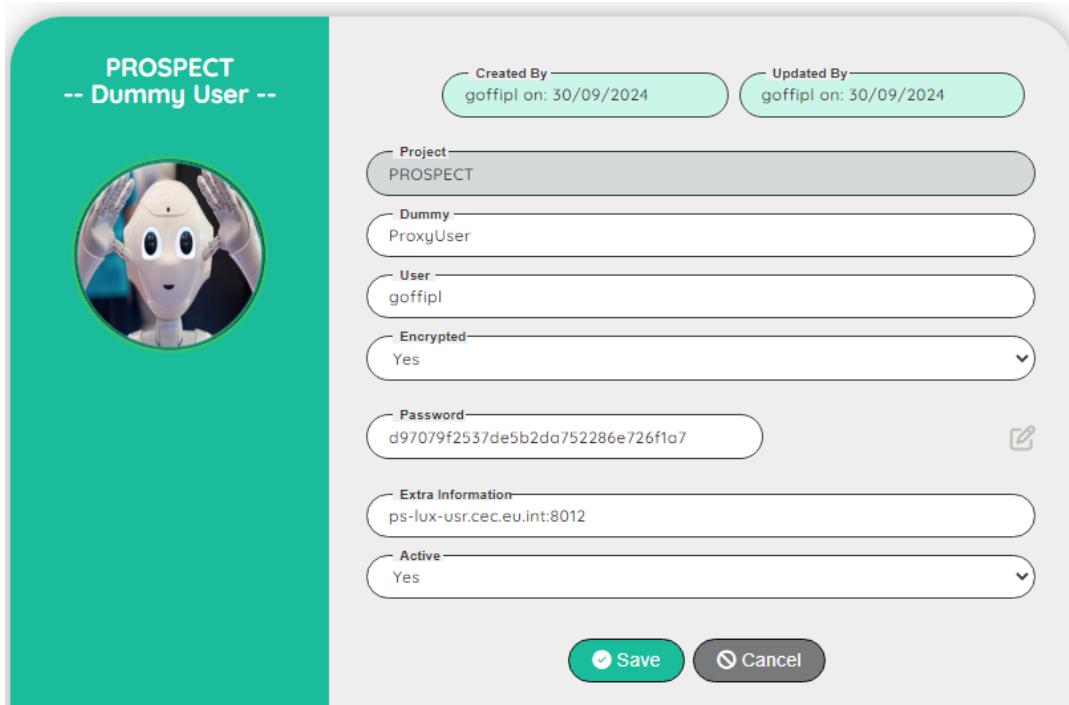


The value must be a valid dummy user

Job to be performed by the Tester (or Designer, Administrator):

Go to the dummy users and add a new entry for the value defined in the parameter (here: ProxyUser)

In the field Extra information, fill the proxy link including the port number



PROSPECT
-- Dummy User --

Created By goffipl on: 30/09/2024

Updated By goffipl on: 30/09/2024

Project PROSPECT

Dummy ProxyUser

User goffipl

Encrypted Yes

Password d97079f2537de5b2da752286e726f1a7

Extra Information ps-lux-usr.cec.eu.int:8012

Active Yes

Save Cancel

Note: You can encrypt the password

If you don't need a proxy login, you don't need to define a specific parameter and dummy user.

If you try to connect to a proxy website without defining the parameter and the dummy user, the application will display a window to ask you to enter a proxy login/password.

This popup window cannot be manipulated by normal functions!

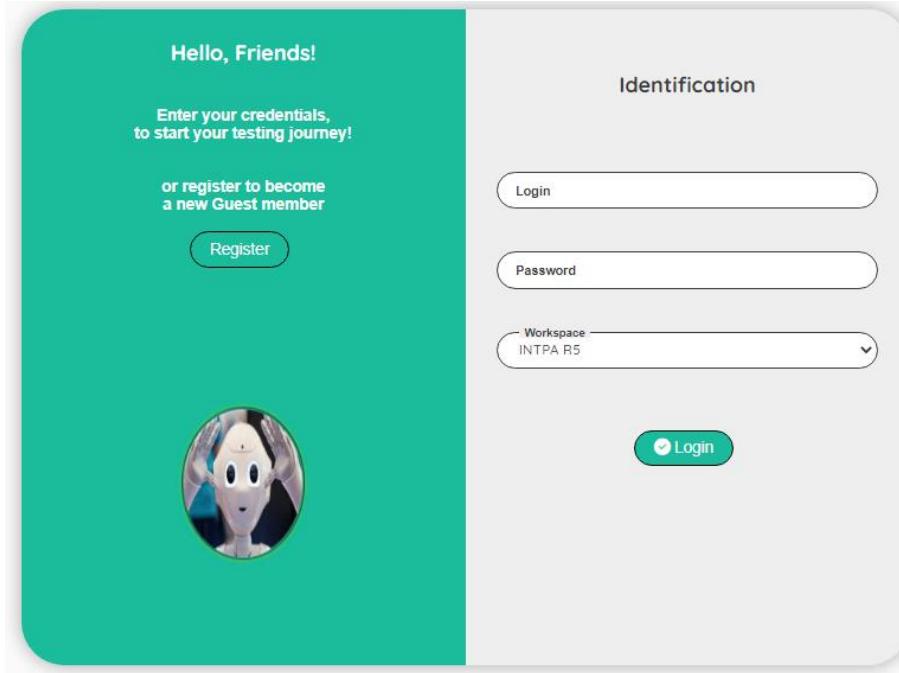
If you define a proxy, the Robot will be able to login to a normal website or a website with proxy.
For information: the time to launch the script will be shorter with a proxy defined!

Login

User Interface

To access to the Robot, you need to have a login.

After you successfully register to the application as a new user (Guest), you have to wait that the Administrator will assign you a Role to a project (or multiple projects).



Topic	Icon	Comment
Register		Click to Register as a new user
Login		Enter your login
Password		Enter your password
Workspace		Enter your workspace
Login		Click to login to the Robot

User interface for the editing

User Interface

We use the same user interface when editing a record.

To avoid repeating the same explanation all the time, you will find here a generic explanation on how to use the edit interface.

When you have the permission to edit a record, you will see the following icons:



Topic	Icon	Comment
Delete	or	Delete a record (a confirmation is required!) To delete multiple records, select the records first.
Edit		Go to a screen to update the record.
Copy		First select the record to copy and then click on the copy icon. Record(s) will be set after the copy icon.
Move		First select the record to move and then click on the move icon. Record(s) will be set after the move icon.
Select / Unselect	/	Select or unselect a record

Depending of the context, some extra icon(s) can be available.

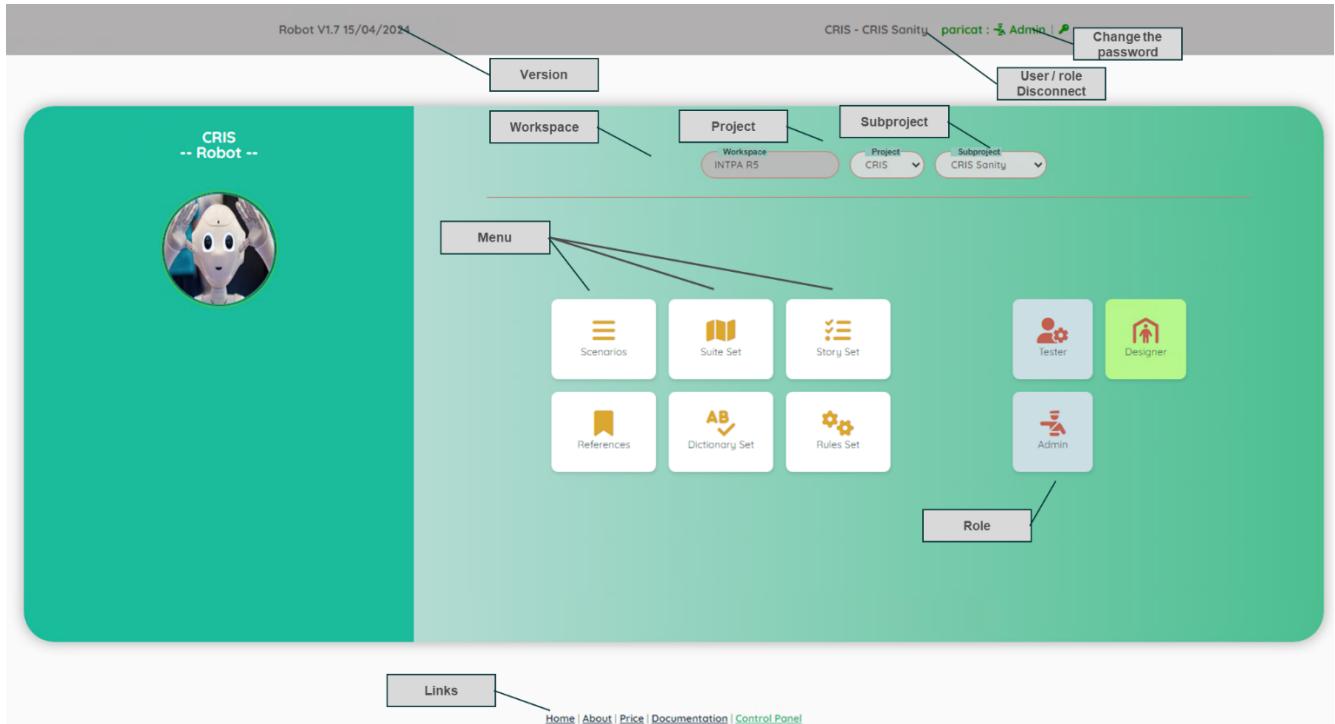
The extra icon(s) will be explain in the section dedicated to the context.

Control Panel

User Interface

The dashboard will allow you to execute pre-defined stories.

If you are assign to a **DESIGNER** or **ADMIN** role, your user interface will look like this:

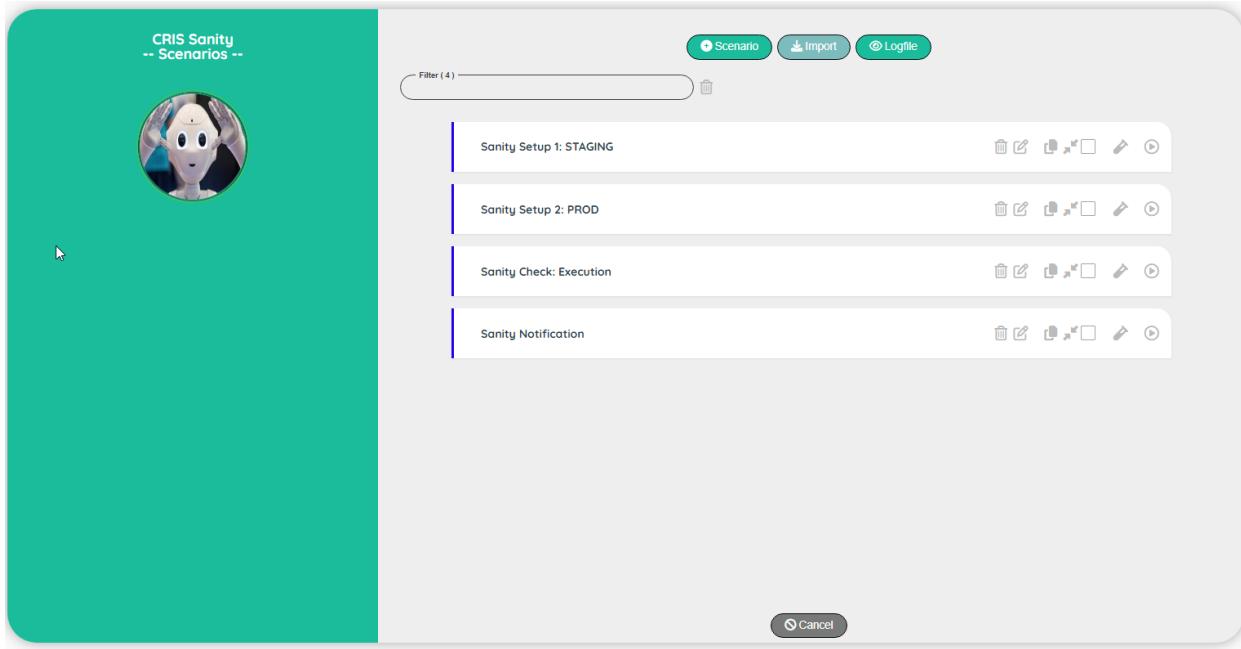


Topic	Comment
Version	Version of the Robot.
Workspace	For info: the name of the workspace of the customer.
Project	List of the project(s) that you have access (assignment to a project is done by an Administrator).
Subproject	Access to all the subproject(s) of the selected project. When executing a story, you can stop the execution.
User/Role Disconnect	For info: your login and your role. If you click on the 'User/Role', you will be disconnected from the application.
Change the password	Click on the icon to change your current password
Menu	Click on the items to open a new section of the application
Links	Useful links

Scenario

User Interface

The Scenario will allow you to manage the scenarios and the tests



Topic	Icon	Comment
Test	-pencil icon	Go to the screen Test.
Execute	play icon	Execute a scenario.
Add Scenario	+ Scenario button	Add a new scenario at the end of the list.
Cancel	Cancel button	Back to the Dashboard screen.
Logfile	@ Logfile button	Go to the screen to view the Log file.
Import	Import button	Import a scenario from any project (same workspace).

Note: if you click on the comment, you will see extra information

Sanity Setup 1: STAGING

ID: 35

Setup the sanity check on STAGING

Updated by: goffip1 on: 27/03/2024

Scenario Edit

CRIS
-- Scenario --



Created By goffipl on: 27/03/2024

Updated By goffipl on: 27/03/2024

Subproject CRIS Sanity

Scenario Sanity Setup 1: STAGING

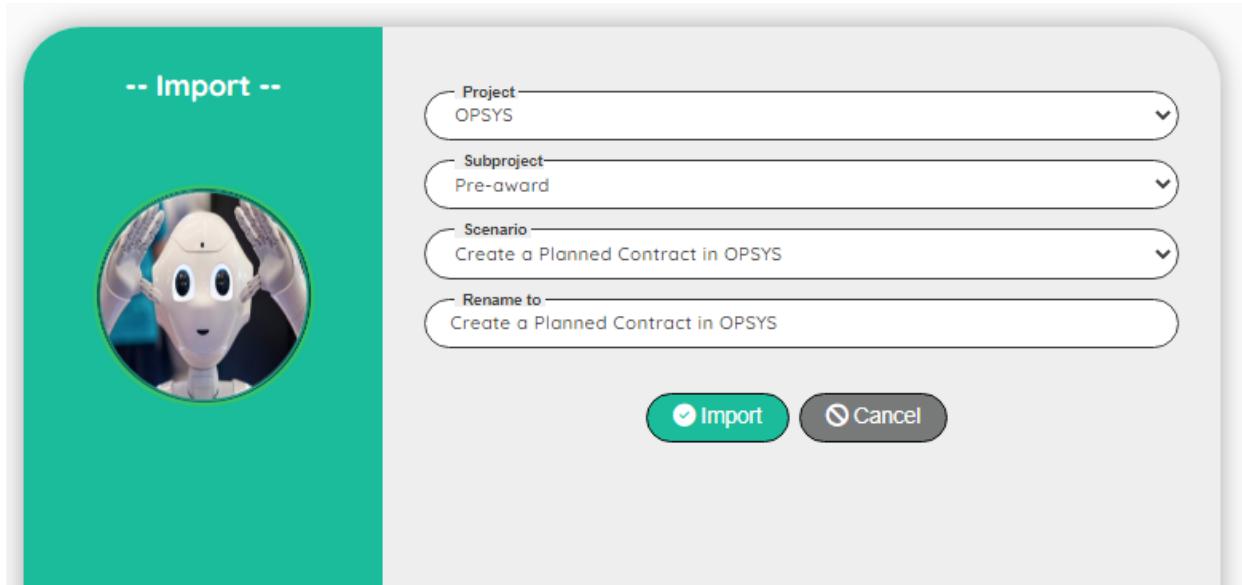
Comment Setup the sanity check on STAGING

Active Yes

Save Cancel

Topic	Icon	Comment
Subproject		For info: Name of the subproject
Scenario		Name of the scenario
Comment		Comment on the scenario
Active		Yes, No
Save	<input checked="" type="button"/> Save	Save the edit
Cancel	<input type="button"/> Cancel	Discard the edit

Scenario Import



Topic	Icon	Comment
Project		Select a project in the list
Subproject		Select a subproject in the list
Scenario		Select a scenario to import
Rename		You can update the name if necessary
Import		Import the scenario
Cancel		Discard the import

Test

The Test will allow you to manage the tests (steps)

The screenshot shows a dashboard titled "Funds Reservation -- Tests --". On the left is a circular profile picture of a robot. The main area displays a list of test steps:

- 1 > Describe: Login to OPSYS
- 2 > It: It possible to login to OPSYS
- 3 > Step: Speak about the objectives of the test (speak)
- 4 > Step: Set the debug level (debug)
- 5 > Step: Get the dataset from the reference (getReference)
- 6 > Step: Get the environment from the reference (getReference)
- 7 > Step: Get the dummy user: Super User (getData)

At the top right, there are buttons for "Test", "Import", "JSON Export", "Row(s) to insert", "Filter (65)", and checkboxes for "All" and "Business". Below the list are standard edit, delete, and refresh icons.

Topic	Icon	Comment
All / Business		Display all comments or only the ones for the Business
Add Test		Add a new test at the beginning of the list.
Active/Inactive		Set the test Inactive/Active
Business/Designer		Set the comment type to Business / Designer (Technical)
Cancel		Back to the Dashboard screen.
Import		Import a test(s) from another scenario (but in the same project)
JSON Export		Export the current test(s) into a .json file
Download	goffipl_test.json	Click on the link to download the .json file

Note: if you click on the comment, you will see extra information

This screenshot shows a detailed view of a specific test step:

3 > Step: Speak about the objectives of the test (speak)

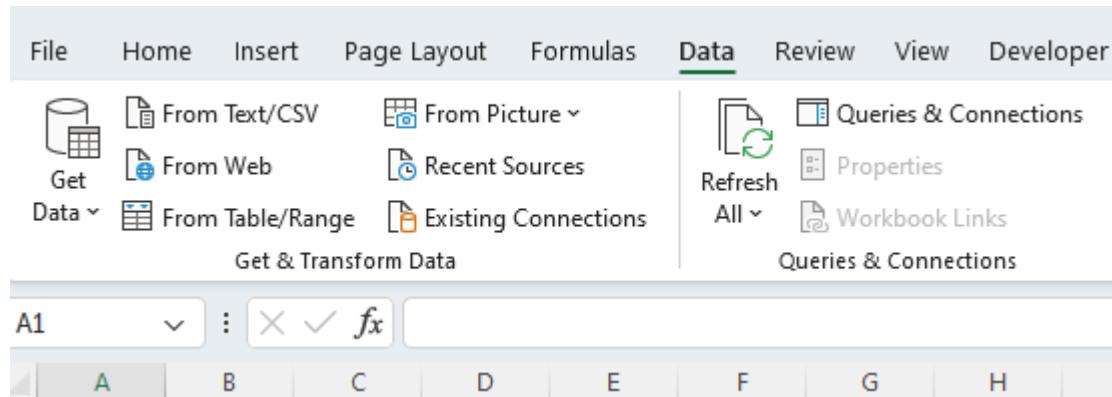
ID: 799

💡 Comment for the Busines

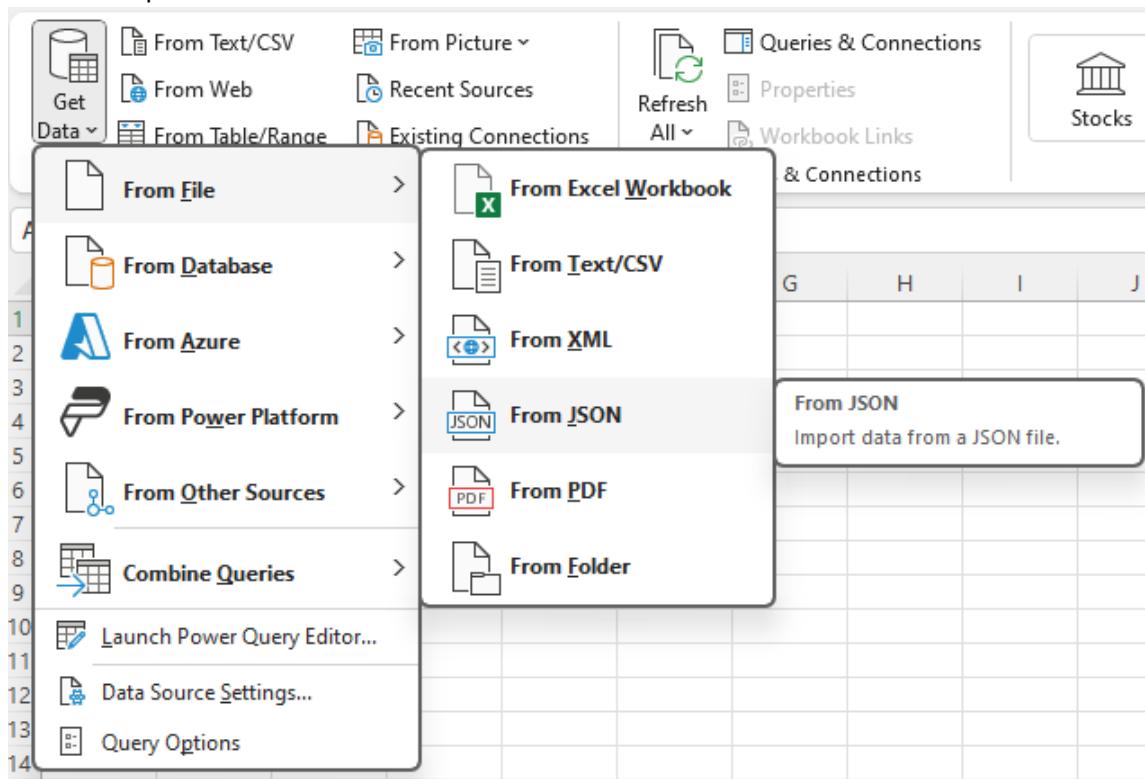
1) Text: Scenario to manage funds reservation

How to process the .json file into Excel

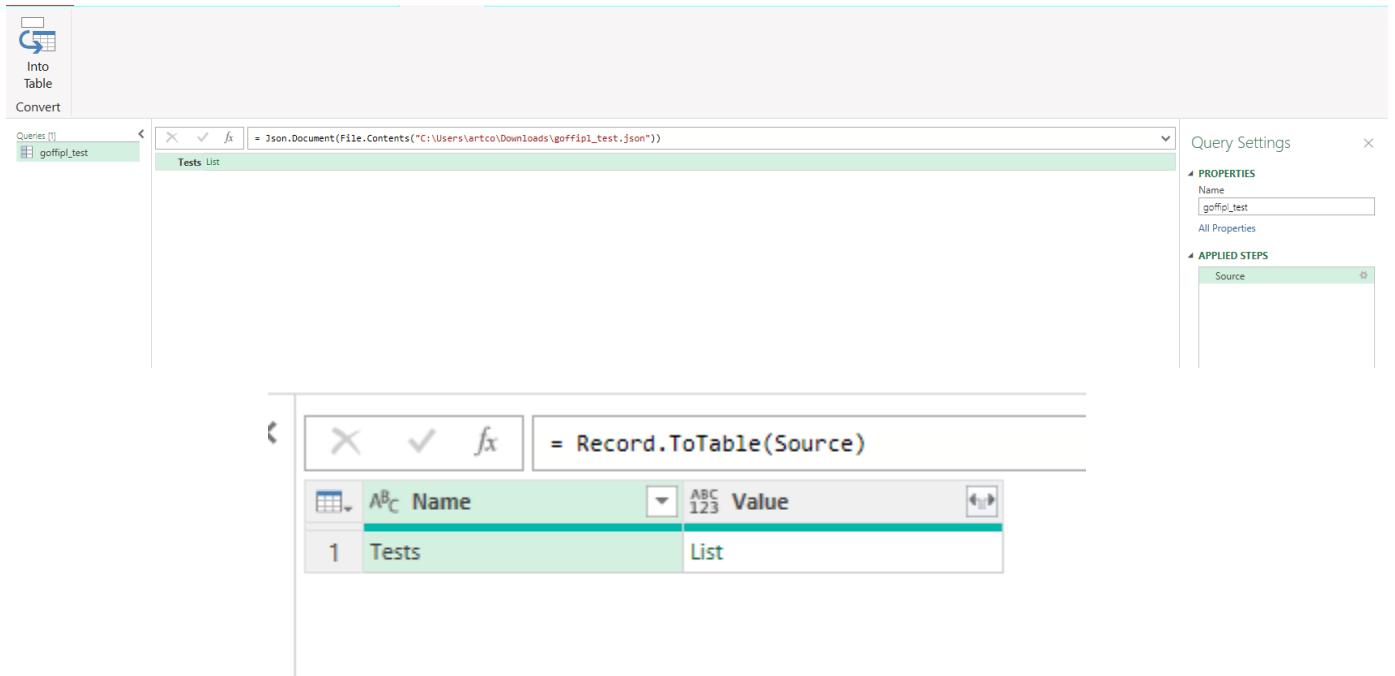
- Open a blank Excel workbook
- Go to the 'Data' section and click on 'Get Data'



- Select the option 'From JSON'



- Select the .json file from the folder 'Download' on your local driver.
a new window is now opened.
Click in the item of the menu 'Into Table Convert'

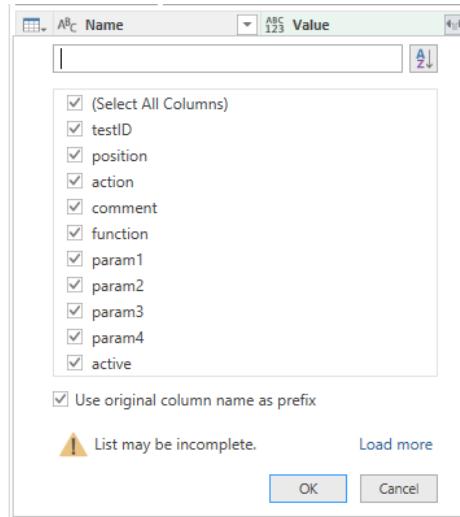


- Click on the right icon of the cell Value and select Expand to New rows

The screenshot shows a context menu for the 'Value' cell in the table, with options 'Expand to New Rows' and 'Extract Values...'. Below the menu, the table now shows six rows, each containing 'Tests' under 'Name' and 'Record' under 'Value'.

Name	Value
1	Tests
2	Tests
3	Tests
4	Tests
5	Tests
6	Tests

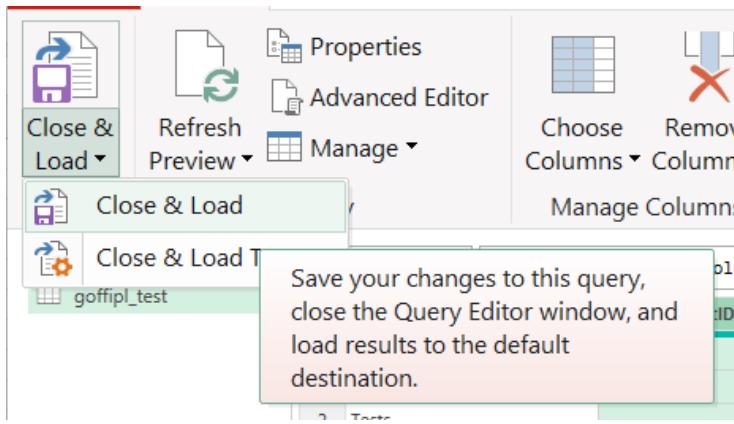
Click again on the right icon of the cell Value and click OK



The data are now displayed into a table.

Click the first item in the menu 'Close & Load'

	Name	testID	position	action	comment
1	Tests	692	1	Describe	Setup Data
2	Tests	693	2	It	It possible to setup data (Dataset)
3	Tests	695	3	Step	Get the dataset
4	Tests	696	4	Step	Copy the Dataset into the Reference
5	Tests	697	5	Step	Get the environment
6	Tests	698	6	Step	Copy the Environment into the Reference



And now, you have a nice table with the tests of a scenario.

You can now manage this table to share with the Business Team.

For instance, you can remove the name of the function and the parameters and eventually, remove the technical test (like wait a little bit or wait for the refresh of the screen...)

A	B	C	D	E	F	G	
1	Name	Value.testID	Value.position	Value.action	Value.comment	Value.function	Value.param1
2	Tests	692 1		Describe	Setup Data	Not selected	
3	Tests	693 2		It	It possible to setup data (Dataset, Environment...)	Not selected	
4	Tests	695 3		Step	Get the dataset	getData	#Data_Dataset
5	Tests	696 4		Step	Copy the Dataset into the Reference	setReference	Dataset
6	Tests	697 5		Step	Get the environment	getData	#Data_Environment
7	Tests	698 6		Step	Copy the Environment into the Reference	setReference	Environment
o							

Test Edit

**Sanity Setup 1:
STAGING
-- Test --**



Scenario — **Sanity Setup 1: STAGING**

Action — **Step**

Comment — Define the STAGING environment 💡

Technical Comment ⚙️

Condition

Function — **setReference** ▼

Code — **Sanity Environment** 🔖

Value — **STAGING**

Reference Comment — Current environment for the sanity check

Status — **Active**

Save Cancel

Topic	Icon	Comment
Scenario		For info: Name of the scenario
Comment	 Comment for the Business	Comment on the test
	 Comment for the Designer	
Technical		Text used for the natural language process
Condition		Any valid JavaScript expression. If the condition is false the test will be skipped!
Function	 / 	Select the name of the function (you can also filter by name or comment)
Parameters		Parameters depending of the selected function
Active		Yes, No
Save		Save the edit
Cancel		Discard the edit

Note: The comment can be defined as a text for the Business or for the Designer (more technical). This is a simple way to indicate to the Business that want to review the scenario which steps are useful for him.

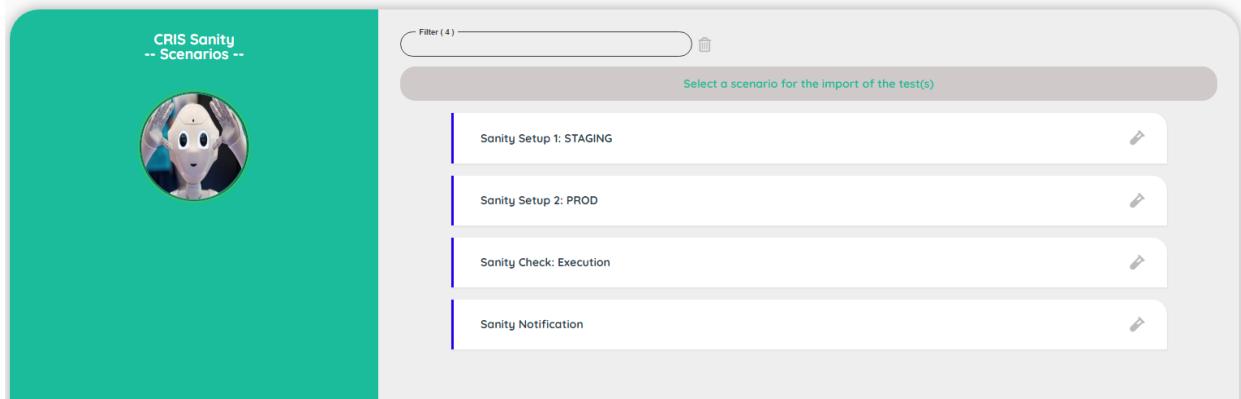
During the export, the field 'commentType' will be exported with the values:

- 1: Comment for the Business
- 0: Comment for the Designer (technical)

Import Test

The Import Test will allow you to import a test from another scenario (but from the same project).

First select a scenario

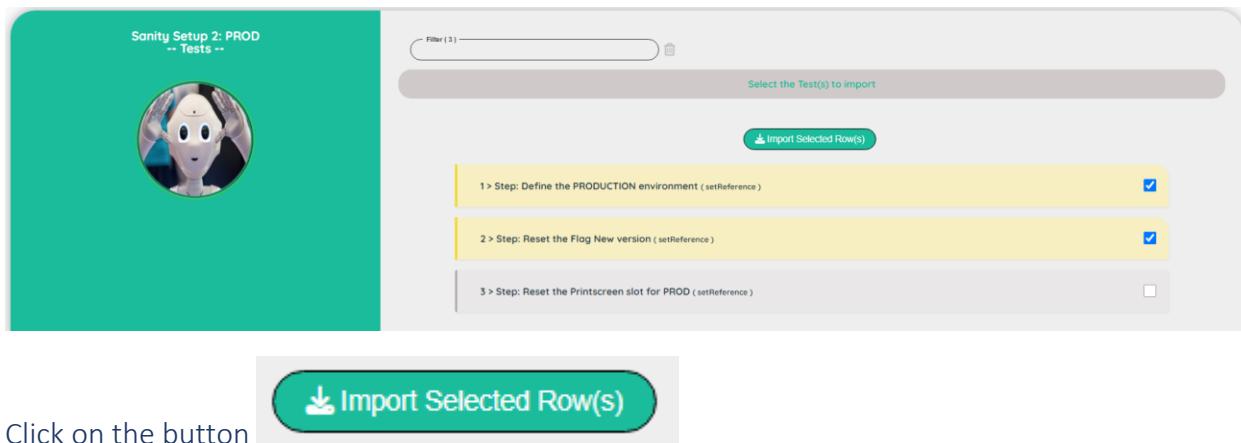


CRIS Sanity
-- Scenarios --

Filter (4) Select a scenario for the import of the test(s)

- Sanity Setup 1: STAGING
- Sanity Setup 2: PROD
- Sanity Check: Execution
- Sanity Notification

Select the test(s) that you want to import



Sanity Setup 2: PROD
-- Tests --

Filter (3) Select the Test(s) to import

Import Selected Row(s)

Step	Description	Status
1 >	Step: Define the PRODUCTION environment (setReference)	<input checked="" type="checkbox"/>
2 >	Step: Reset the Flag New version (setReference)	<input checked="" type="checkbox"/>
3 >	Step: Reset the Printscreen slot for PROD (setReference)	<input type="checkbox"/>

Import Selected Row(s)

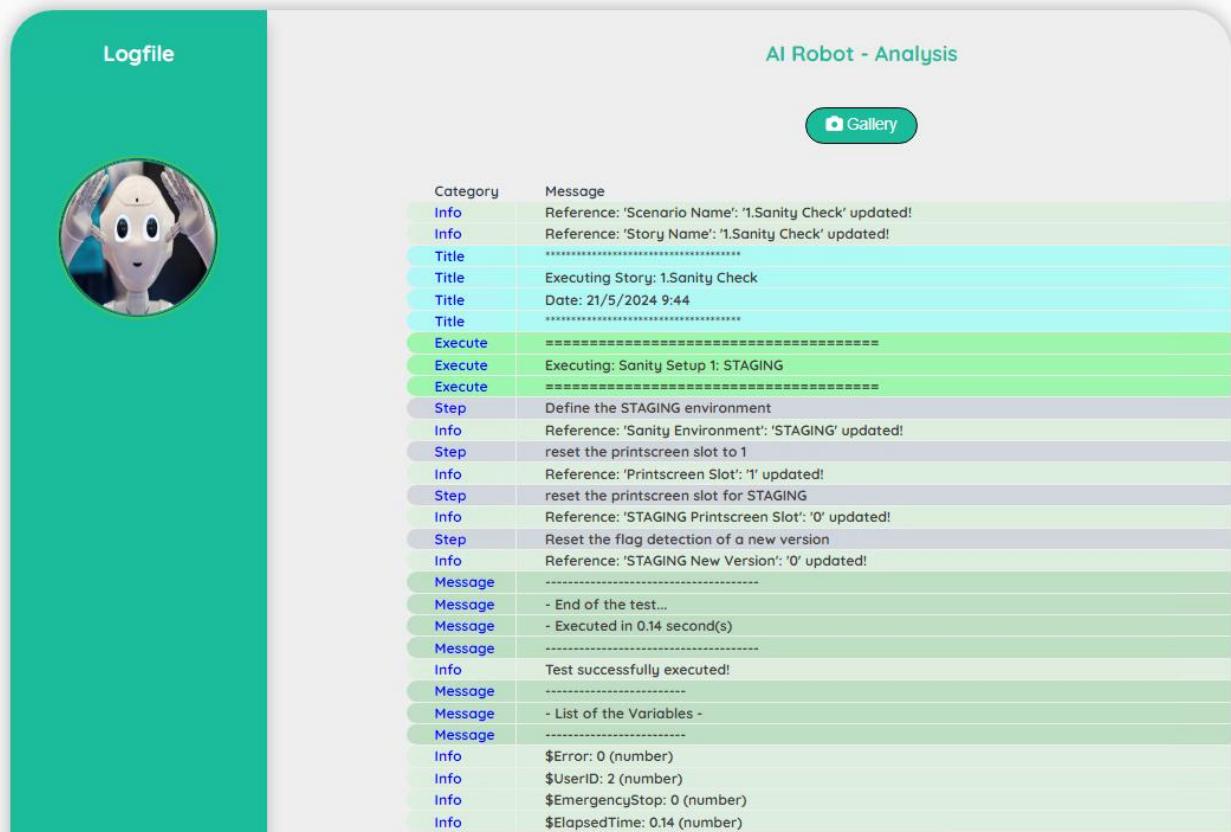
Click on the button

Logfile

The logfile will show you all the steps executed with extra information.

It helps you to understand how the test has been executed.

Note: There is one logfile by user. You see always the logfile of the last execution



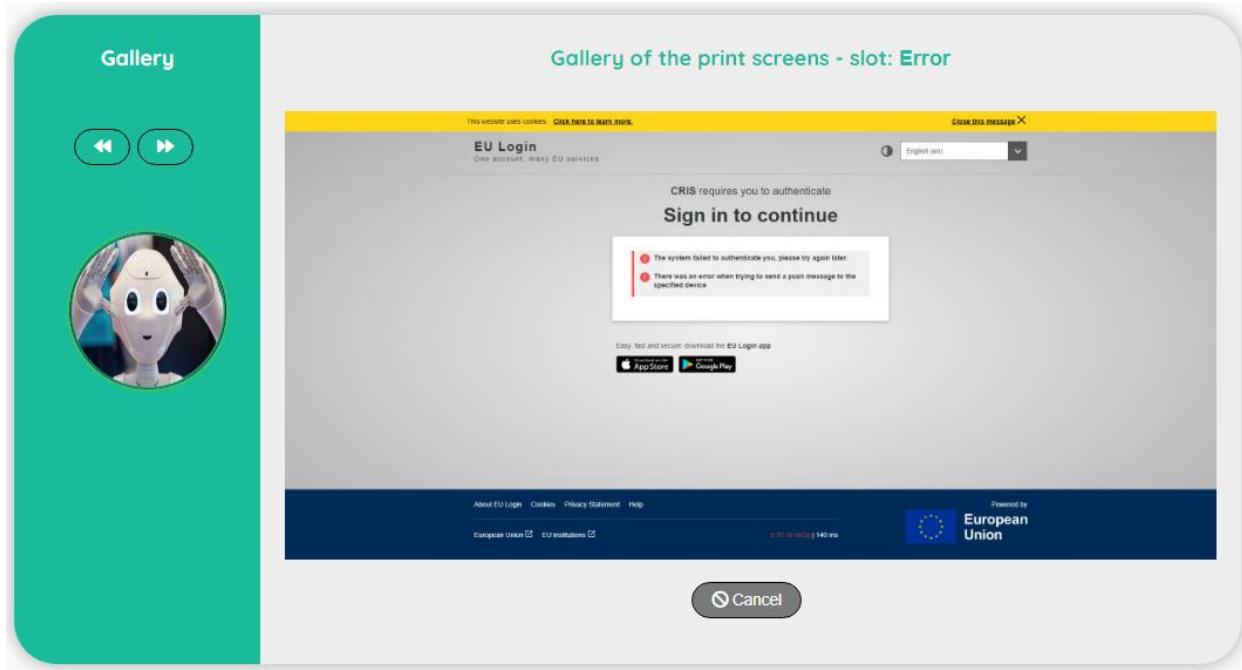
The screenshot shows the AI Robot - Analysis interface with a teal sidebar on the left containing a circular profile picture of a white robot head and the word "Logfile". The main area is titled "AI Robot - Analysis" and contains a table of log entries. A green "Gallery" button is located at the top right of the main area. The log entries are as follows:

Category	Message
Info	Reference: 'Scenario Name': '1.Sanity Check' updated!
Info	Reference: 'Story Name': '1.Sanity Check' updated!
Title	*****
Title	Executing Story: 1.Sanity Check
Title	Date: 21/5/2024 9:44
Title	*****
Execute	=====
Execute	Executing: Sanity Setup 1: STAGING
Execute	=====
Step	Define the STAGING environment
Info	Reference: 'Sanity Environment': 'STAGING' updated!
Step	reset the printscreenslot to 1
Info	Reference: 'Printscreen Slot': '1' updated!
Step	reset the printscreenslot for STAGING
Info	Reference: 'STAGING Printscreen Slot': '0' updated!
Step	Reset the flag detection of a new version
Info	Reference: 'STAGING New Version': '0' updated!
Message	-----
Message	- End of the test...
Message	- Executed in 0.14 second(s)
Message	-----
Info	Test successfully executed!
Message	-----
Message	- List of the Variables -
Message	-----
Info	\$Error: 0 (number)
Info	\$UserId: 2 (number)
Info	\$EmergencyStop: 0 (number)
Info	\$Elapsed Time: 0.14 (number)

Topic	Icon	Comment
Gallery		Access to the print screen(s)

Gallery

The gallery will show you the print screen taken during the execution of the story.
 The first print screen is done automatically by the Robot when an error occurs (it will show you the last error detected – in case of success of a story, you will see the last error detected)

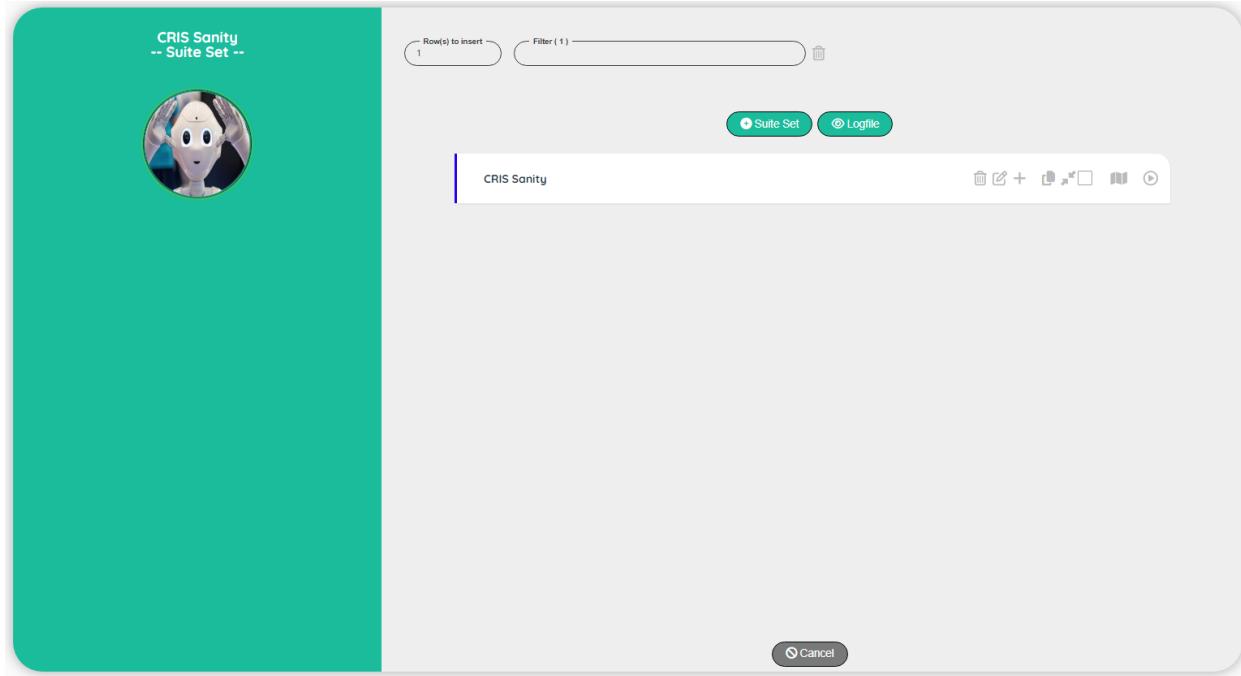


Topic	Icon	Comment
Previous		Go to the previous print screen
Next		Go to the next print screen
Cancel		Back to the Logfile

Suite Set

User Interface

The Suite will allow you to group different scenarios (you can also consult the best practices guide for more information about the suite). A suite is composed from a suite set (parent) and suites (children)



Topic	Icon	Comment
Suite	Book icon	Go to the screen suite
Execute	Play button icon	Execute a suite.
Add Suite set	+ Suite Set button	Add a new Suite at the beginning of the list.
Cancel	Cancel button	Back to the Dashboard screen.
LogFile	LogFile button	Go to the screen to view the Log file.

Suite Set Edit

CRIS Sanity
-- Suite set --



Created By goffipl on: 25/03/2024 Updated By goffipl on: 25/03/2024

Subproject CRIS Sanity

Suite set CRIS Sanity

Comment
Perform a sanity check on CRIS

Status Active

Save Cancel

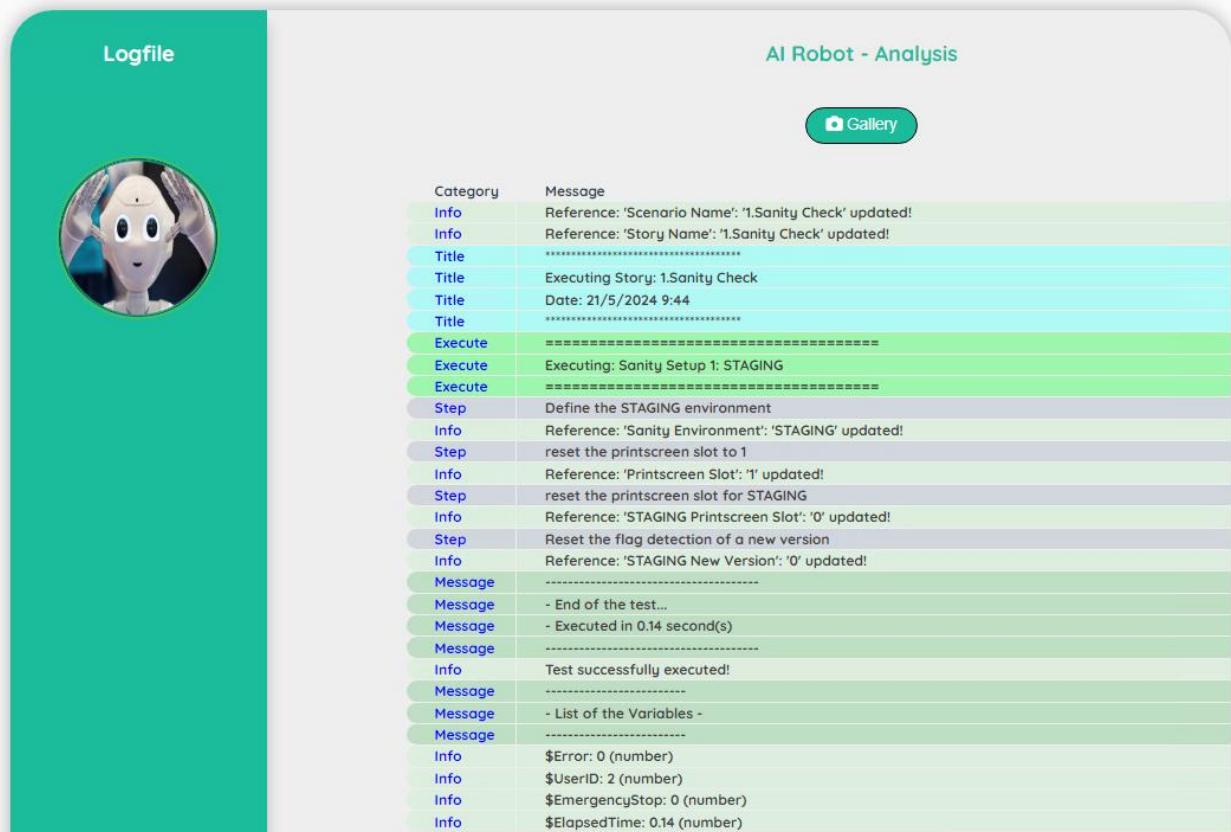
Topic	Icon	Comment
Subproject		For info: Name of the subproject
Suite set		Name of the suite set
Comment		Comment for the suite set
Active		Yes, No
Save	Save	Save the edit
Cancel	Cancel	Discard the edit

Logfile

The logfile will show you all the steps executed with extra information.

It helps you to understand how the test has been executed.

Note: There is one logfile by user. You see always the logfile of the last execution



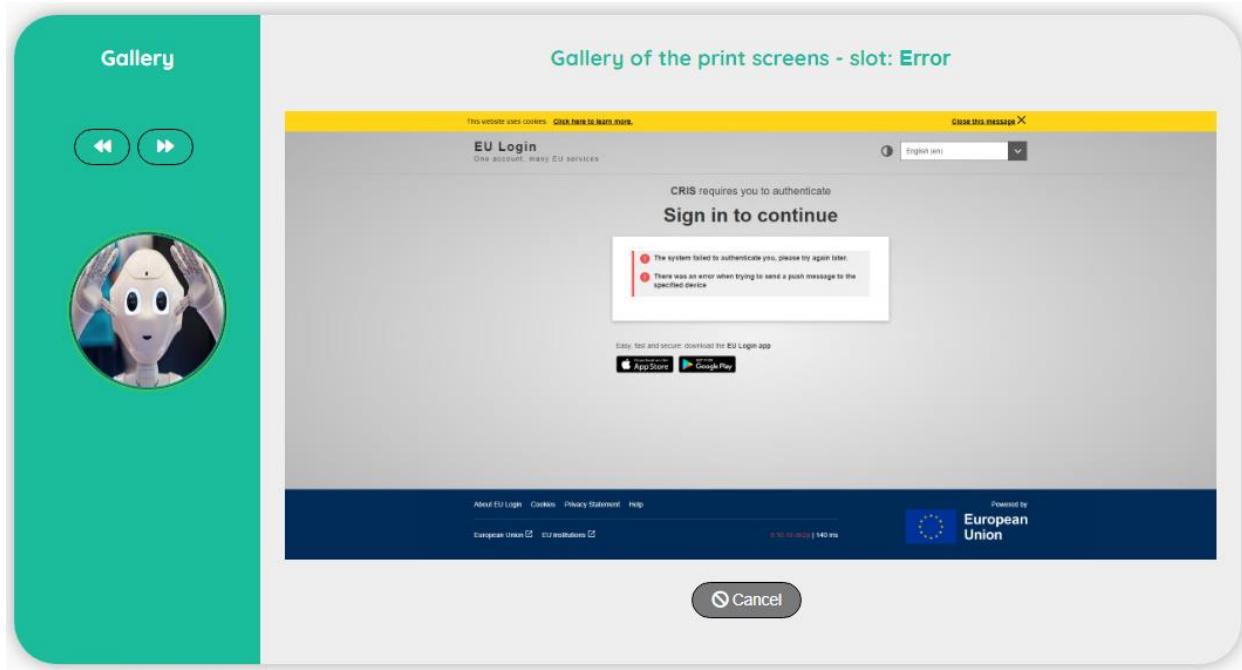
The screenshot shows the AI Robot - Analysis interface with a teal sidebar on the left containing a circular profile picture of a white robot head and the word "Logfile". The main area is titled "AI Robot - Analysis" and contains a table of log entries. A green "Gallery" button is located at the top right of the main area. The log entries are as follows:

Category	Message
Info	Reference: 'Scenario Name': '1.Sanity Check' updated!
Info	Reference: 'Story Name': '1.Sanity Check' updated!
Title	*****
Title	Executing Story: 1.Sanity Check
Title	Date: 21/5/2024 9:44
Title	*****
Execute	=====
Execute	Executing: Sanity Setup 1: STAGING
Execute	=====
Step	Define the STAGING environment
Info	Reference: 'Sanity Environment': 'STAGING' updated!
Step	reset the printscreenslot to 1
Info	Reference: 'Printscreen Slot': '1' updated!
Step	reset the printscreenslot for STAGING
Info	Reference: 'STAGING Printscreen Slot': '0' updated!
Step	Reset the flag detection of a new version
Info	Reference: 'STAGING New Version': '0' updated!
Message	-----
Message	- End of the test...
Message	- Executed in 0.14 second(s)
Message	-----
Info	Test successfully executed!
Message	-----
Message	- List of the Variables -
Message	-----
Info	\$Error: 0 (number)
Info	\$UserId: 2 (number)
Info	\$EmergencyStop: 0 (number)
Info	\$Elapsed Time: 0.14 (number)

Topic	Icon	Comment
Gallery		Access to the print screen(s)

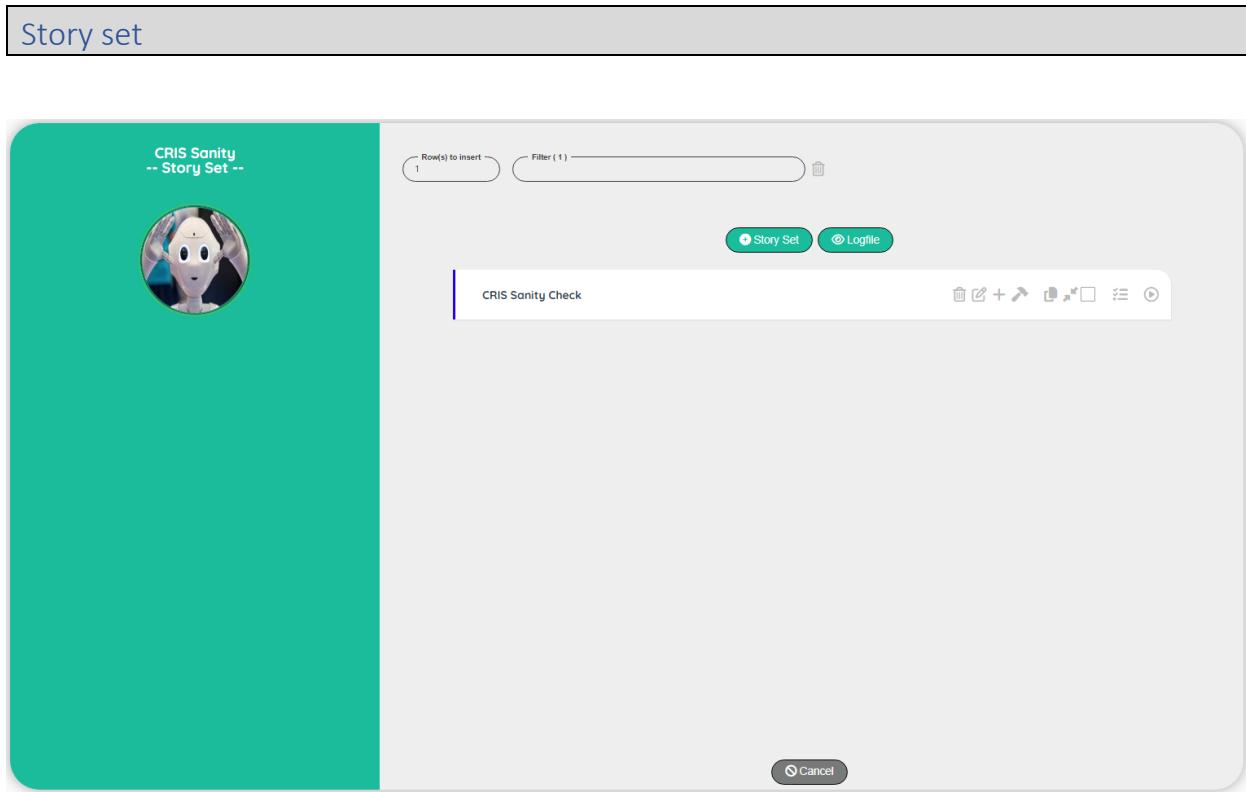
Gallery

The gallery will show you the print screen taken during the execution of the story.
 The first print screen is done automatically by the Robot when an error occurs (it will show you the last error detected – in case of success of a story, you will see the last error detected)



Topic	Icon	Comment
Previous		Go to the previous print screen
Next		Go to the next print screen
Cancel		Back to the Logfile

Story Set



Topic	Icon	Comment
Stories		Go to the screen Stories.
Execute		Execute a Story.
Parameters		Add a story at the beginning of the list
Add Story		Add a story at the beginning of the list
Cancel		Back to the control panel
Logfile		Go to the screen to view the Log file.

Story set Edit

CRIS Sanity
-- Story set --

Created By
goffipl on: 27/03/2024

Updated By
goffipl on: 27/03/2024

Subproject
CRIS Sanity

Story set
CRIS Sanity Check

Comment
CRIS Sanity Check

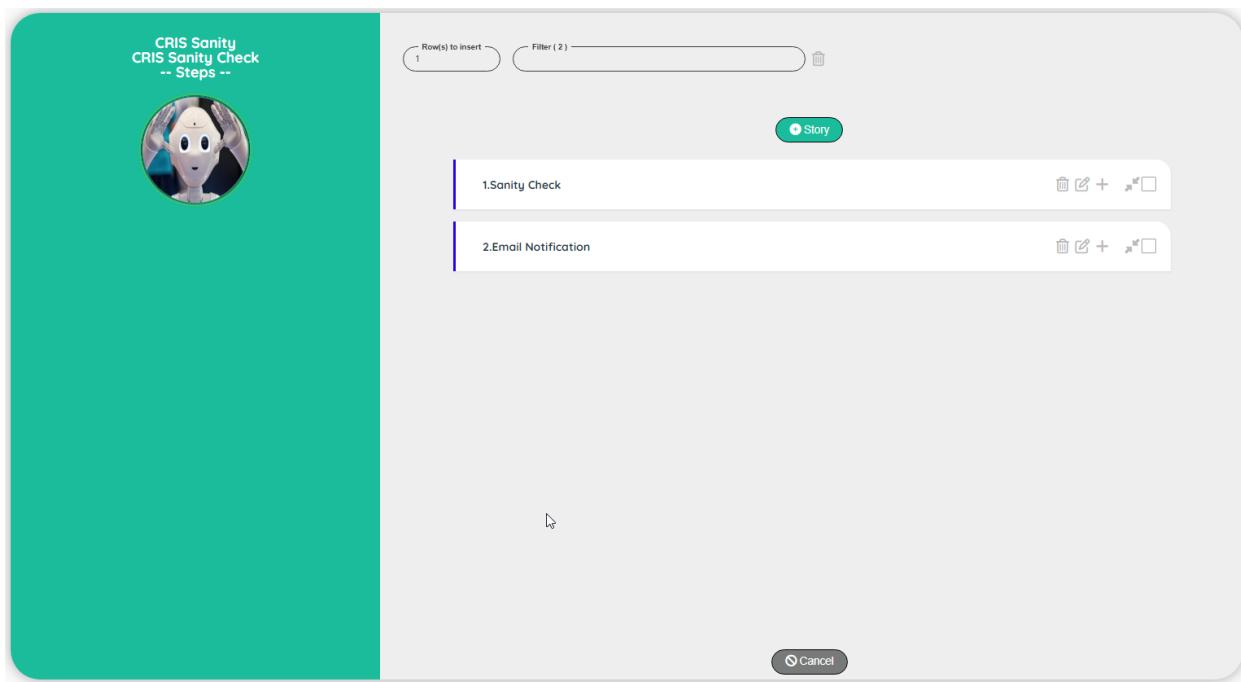
Status
Active (restricted to Designer & Admin)

Save Cancel

Topic	Icon	Comment
Subproject		For info: Name of the subproject
Story set		Name of the story set
Comment		Comment for the story set
Active		Active, Not active, Publish
Save		Save the edit
Cancel		Discard the edit

Note: The Tester can see only the Published story (Active and Published are visible by the Designer and the Administrator)

Stories



Topic	Icon	Comment
Add Story		Add a story at the beginning of the list
Cancel		Back to the Story set

Stories Edit

CRIS
-- Story --



Subproject
CRIS Sanity

Story set
CRIS Sanity Check

Story
1.Sanity Check

Graph label
Sanity Check

Selector
Suite

Suite
10

Comment
Perform a sanity check on CRIS

Status
Active

Save Cancel

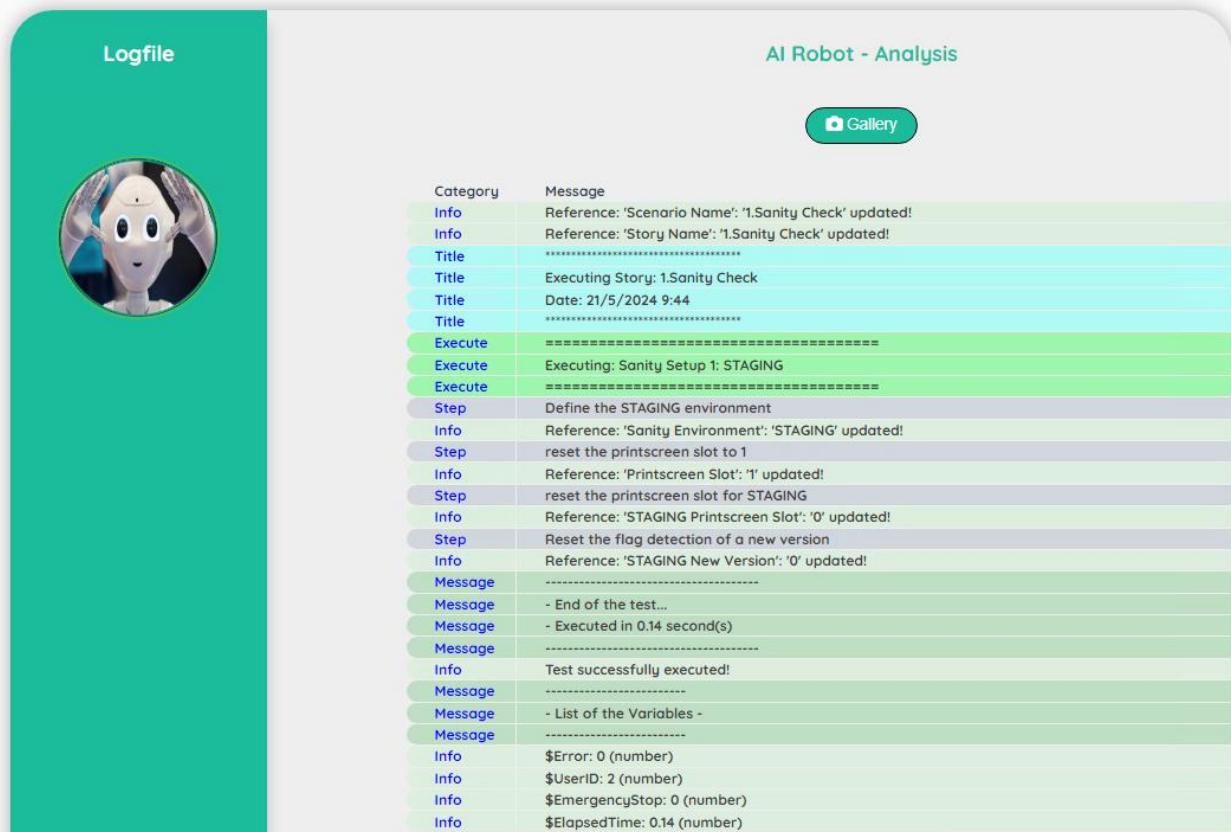
Topic	Icon	Comment
Subproject		For info: Name of the subproject
Story set		For info: Name of the story set
Story		Name of the story
Graph label		Scenario or Suite
Selector		Comment for the story set
Selector Info		ID and Comment on the selected selector
Active		Active, Not active, Publish
Save	Save	Save the edit
Cancel	Cancel	Discard the edit

Logfile

The logfile will show you all the steps executed with extra information.

It helps you to understand how the test has been executed.

Note: There is one logfile by user. You see always the logfile of the last execution



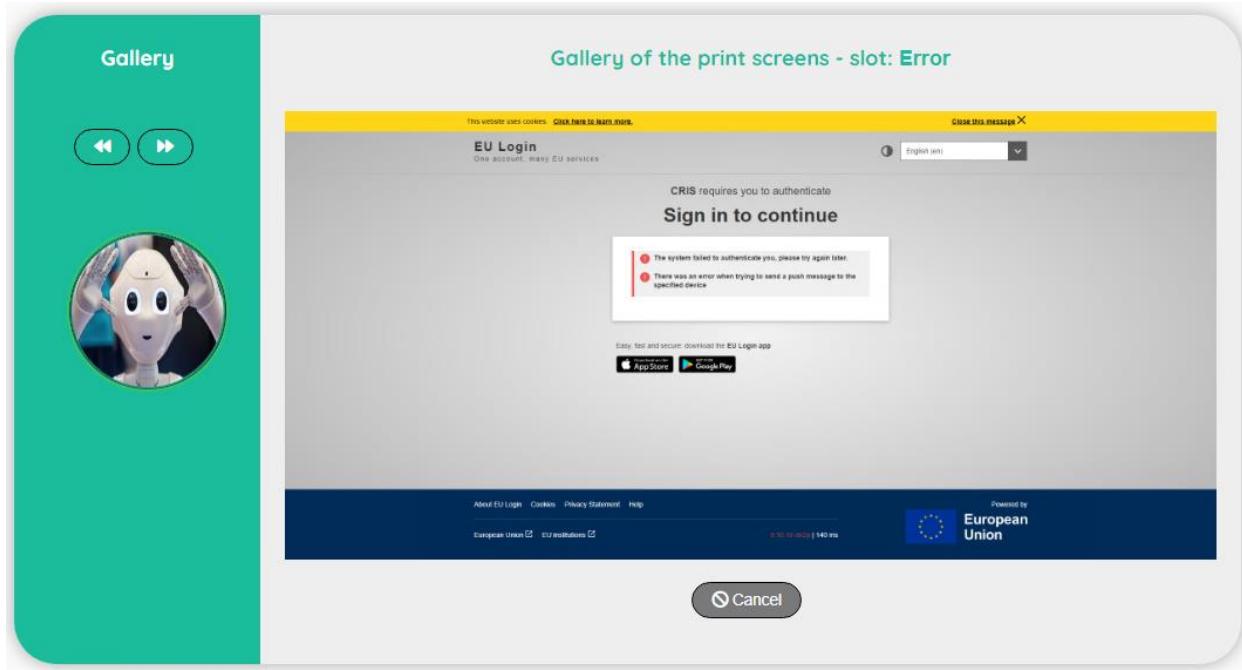
The screenshot shows the AI Robot - Analysis interface with a teal sidebar on the left containing a circular profile picture of a white robot head and the word "Logfile". The main area is titled "AI Robot - Analysis" and contains a table of log entries. At the top right of the main area is a green button labeled "Gallery" with a camera icon. The log entries are color-coded by category:

Category	Message
Info	Reference: 'Scenario Name': '1.Sanity Check' updated!
Info	Reference: 'Story Name': '1.Sanity Check' updated!
Title	*****
Title	Executing Story: 1.Sanity Check
Title	Date: 21/5/2024 9:44
Title	*****
Execute	=====
Execute	Executing: Sanity Setup 1: STAGING
Execute	=====
Step	Define the STAGING environment
Info	Reference: 'Sanity Environment': 'STAGING' updated!
Step	reset the printscreenslot to 1
Info	Reference: 'Printscreen Slot': '1' updated!
Step	reset the printscreenslot for STAGING
Info	Reference: 'STAGING Printscreen Slot': '0' updated!
Step	Reset the flag detection of a new version
Info	Reference: 'STAGING New Version': '0' updated!
Message	-----
Message	- End of the test...
Message	- Executed in 0.14 second(s)
Message	-----
Info	Test successfully executed!
Message	-----
Message	- List of the Variables -
Message	-----
Info	\$Error: 0 (number)
Info	\$UserId: 2 (number)
Info	\$EmergencyStop: 0 (number)
Info	\$Elapsed Time: 0.14 (number)

Topic	Icon	Comment
Gallery		Access to the print screen(s)

Gallery

The gallery will show you the print screen taken during the execution of the story.
 The first print screen is done automatically by the Robot when an error occurs (it will show you the last error detected – in case of success of a story, you will see the last error detected)



Topic	Icon	Comment
Previous		Go to the previous print screen
Next		Go to the next print screen
Cancel		Back to the Logfile

Reference

Reference

The reference will show all the values stored by the scenarios in order to exchange data. A reference is own by a specific user for a project.

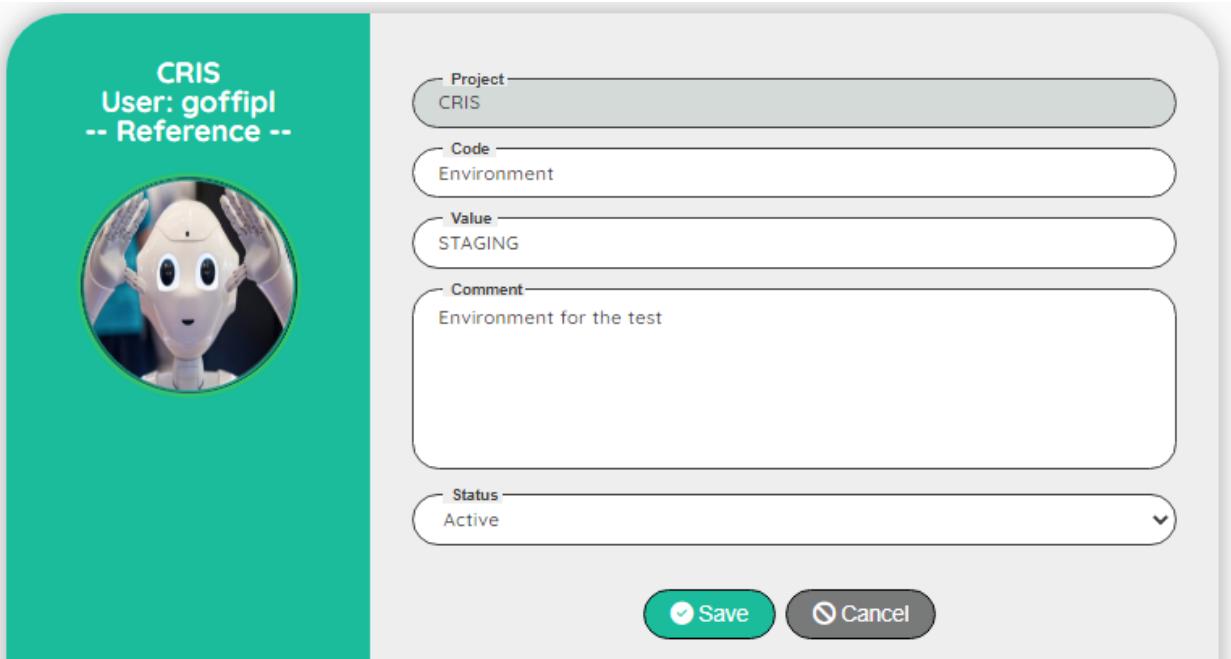
The screenshot shows the 'CRIS User: goffipl References' screen. At the top, there is a header with the user's name and a small profile picture of a robot. Below the header, there are input fields for 'Row(s) to Insert' (set to 1) and 'Filter (21)' with a clear button. At the top right are three buttons: 'Reference' (highlighted in green), 'Export', and 'Import'. The main area contains a table with the following data:

Reference Name	Environment	Dataset	Test Status	Sanity Environment
Scenario Name - 2.Email Notification	STAGING	#CTR Budget	1	PROD
Environment - STAGING				
Dataset - #CTR Budget				
TEST Status - 1				
Sanity Environment - PROD				
STAGING Error - 0				
STAGING Printscreen Slot - 0				
STAGING New Version - 0				

At the bottom right is a 'Cancel' button.

Topic	Icon	Comment
Add Reference		Add a reference at the beginning of the list.
Cancel		Back to the control panel.
Export		Go to the screen to export a reference.
Import		Go to the screen to import a reference.

Reference Edit



CRIS
User: goffipl
-- Reference --

Project: CRIS

Code: Environment

Value: STAGING

Comment: Environment for the test

Status: Active

Save Cancel

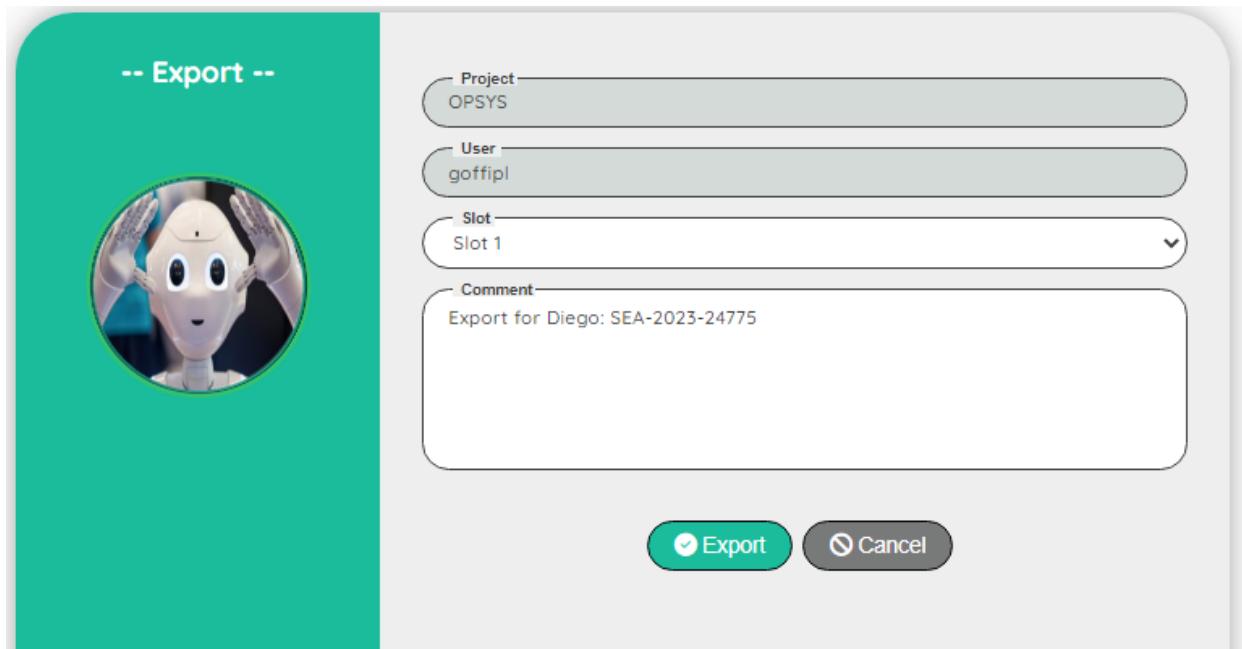
Topic	Icon	Comment
Project		For info: Name of the project
Code		Code of the reference
Value		Value of the reference
Comment		Comment for the reference
Active		Active, Not active
Save	<input checked="" type="button"/> Save	Save the edit
Cancel	<input type="button"/> Cancel	Discard the edit

Note: The reference 'Environment' is very important for the tool, be sure to create one and keep the value relevant to your tests ('Environment' is used in the Dashboard to display the graph with the performances).

Example of value for Environment could be for instance: PROD, ACC, TEST, STAGING....

Export Reference

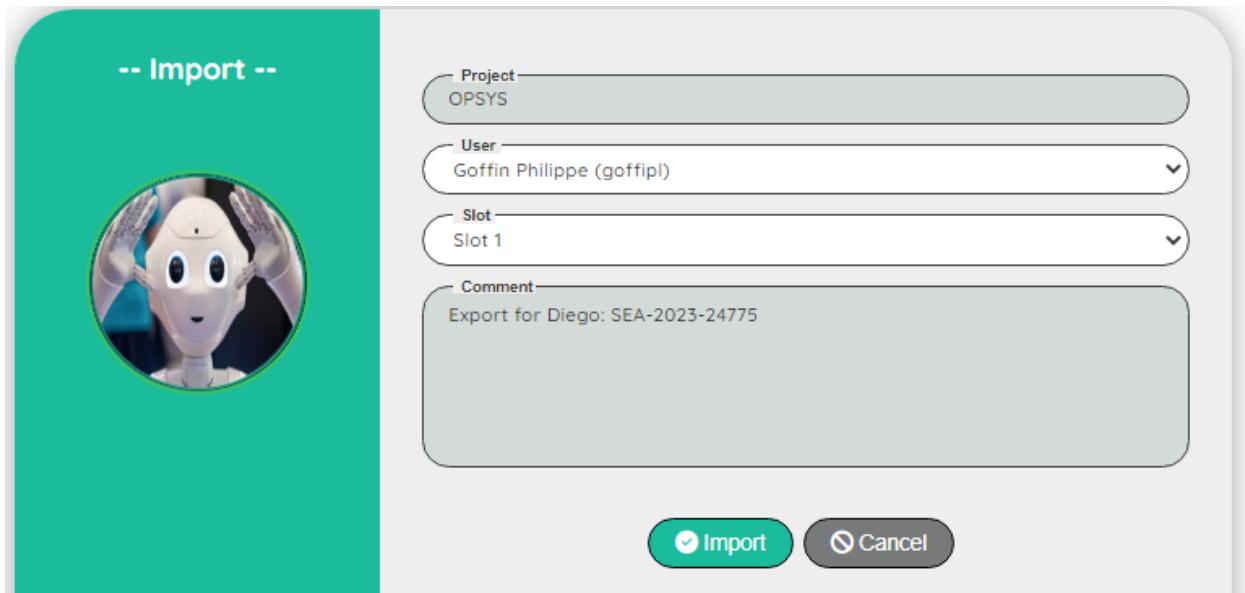
The export will allow you to take a backup or to share your reference with another colleague.



Topic	Icon	Comment
Project		For info: Name of the project
User		For info: Login of the user
Slot		10 slots are available to back up your references
Comment		Comment on the export
Export		Export the references
Cancel		Back to the control panel.

Import Reference

The import will allow you to restore a backup or get a reference from another colleague. Be careful, your current references will be erased and replaced by the import.



Topic	Icon	Comment
Project		For info: Name of the project
User		Select a user in the list
Slot		Select a slot
Comment		For info: Comment on the export
Import		Import the references
Cancel		Back to the control panel.

Dictionary

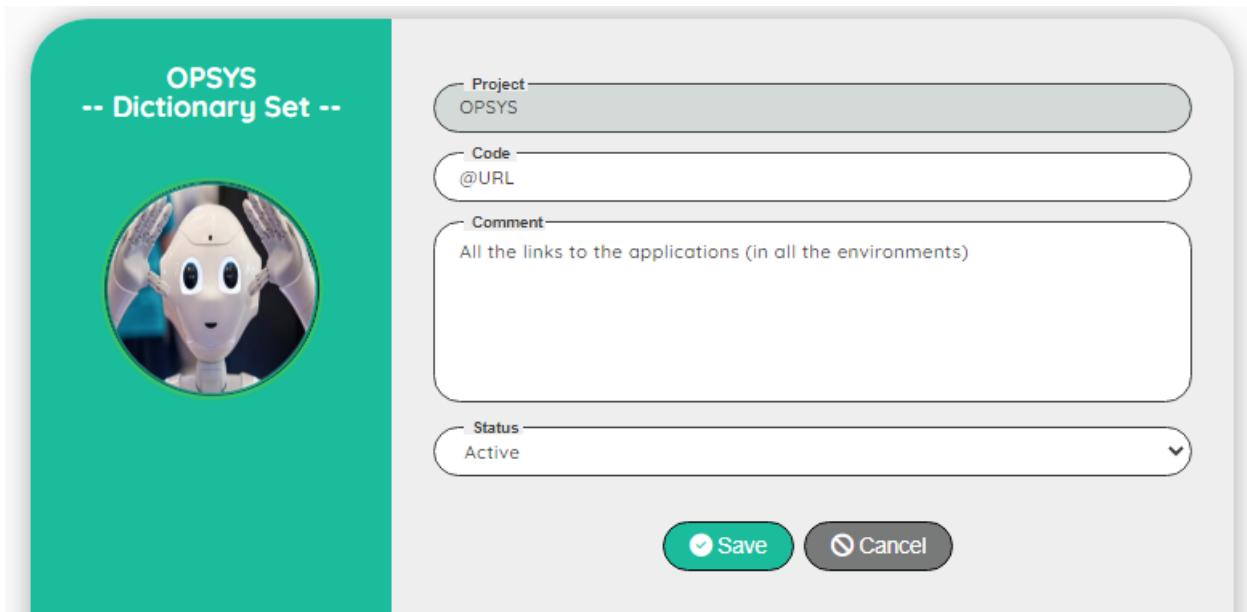
Dictionary set

The dictionary is available for a project. It is composed of a dictionary set and a dictionary word. You can store URL, xpath or even translation (there is a field language that you can use) For the translation, the logic is the following: if you ask for an unknown language, the record with the language '*' will be used. So, you can for instance, use '*' for English and if a translation in another language is missing, the user will receive the English translation.

The screenshot shows the 'Dictionary Set' screen. On the left, there is a large teal sidebar featuring a white robot icon. The main area has a header 'OPSYS -- Dictionary Set --'. Below the header are two input fields: 'Row(s) to insert' (set to 1) and 'Filter (0)'. There are two buttons at the top right: 'Dictionary Set' (highlighted in green) and 'Unused Word'. The main content area displays a list of six entries, each with a delete icon, an edit icon, a plus sign, and a language dropdown menu (set to AB). The entries are: '@URL', '@OPSYS', '@PORTFOLIO', '@PLAN CONTRACT', '@COMPASS', and '@SYGMA'. At the bottom right of the main area is a 'Cancel' button.

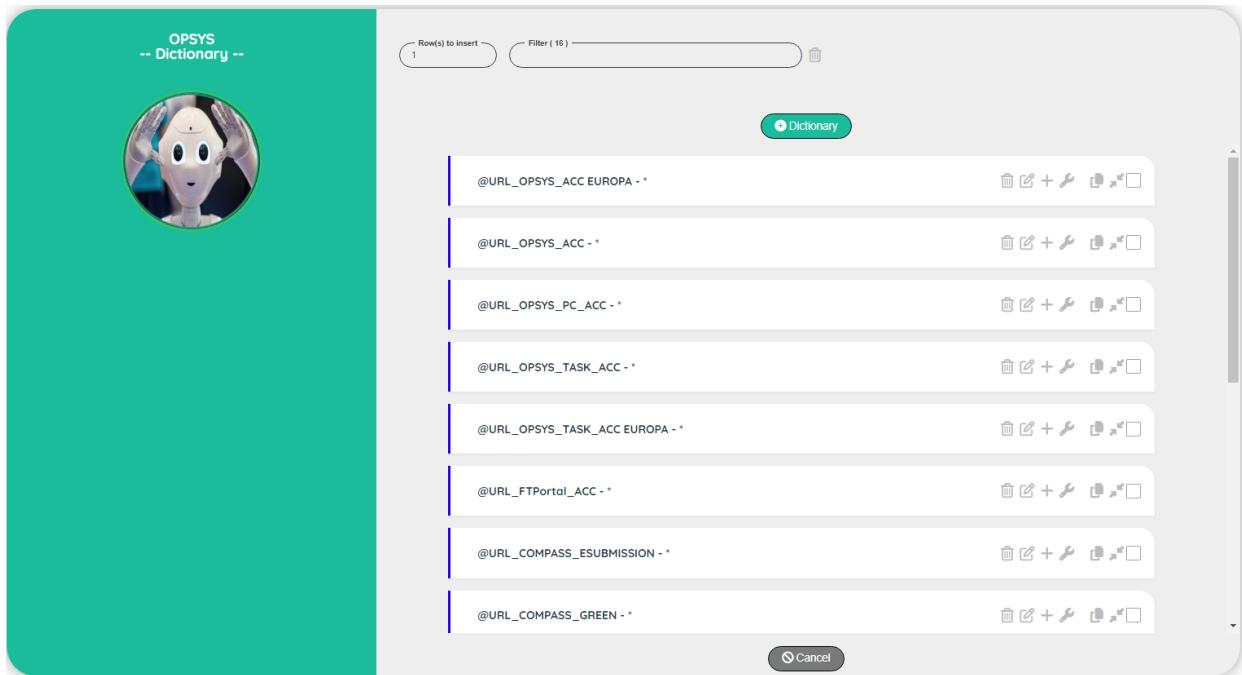
Topic	Icon	Comment
Dictionary		Go to the screen dictionary word
Add Dictionary		Add a dictionary set at the beginning of the list.
Unused word		Go to the screen to export a reference.
Cancel		Back to the control panel.

Dictionary set Edit



Topic	Icon	Comment
Project		For info: Name of the project
Code		Code of the dictionary must start by @
Comment		Comment for the dictionary
Active		Active, Not active
Save		Save the edit
Cancel		Discard the edit

Dictionary word



Topic	Icon	Comment
Add Dictionary		Add a word at the beginning of the list.
Cancel		Back to the control panel.

Dictionary word Edit

OPSYS
-- Dictionary --

Created By goffipl on: 01/03/2024 Updated By goffipl on: 02/04/2024

Project OPSYS

Header Code @URL

Code _OPSYS_ACC EUROPA

Language EN

Value <https://webgate.acceptance.ec.europa.eu/mwp/home?1fa>

Comment
Link to the OPSYS webpage in ACC (via Europa)

Status Active

Save Cancel

Topic	Icon	Comment
Project		For info: Name of the project
Dictionary set		For info: Name of the dictionary set
Code		Code of the dictionary must start by @
Language		* by default, but you can use an ISO code (EN, FR...)
Comment		Comment for the dictionary word
Active		Active, Not active
Save	<input checked="" type="button"/> Save	Save the edit
Cancel	<input type="button"/> Cancel	Discard the edit

Dictionary unused word

This screen will show you all the unused word(s).

Be careful, the detection can be fooled when you use a variable to replace a partial name of a word. For instance if you have @URL_Acceptance, it will be flagged as unused if you use the following step in your scenario: @URL_\$Environment.

The screenshot shows the OPSYS Dictionary interface. On the left, there is a teal sidebar with a robot icon and the text "OPSYs -- Dictionary --". The main area has a white background with a header bar containing "Row(s) to insert 1", "Filter (7)", and a trash bin icon. Below this is a green button labeled "+ Dictionary". The main content area displays a list of six items, each with a delete, edit, and search icon:

- @PORTFOLIO_Menu: Impl Plan: Contract - *
- @PLAN CONTRACT_Signature Date - *
- @PLAN CONTRACT_Contract System List - *
- @PLAN CONTRACT_Contract System Select - *
- @PLAN CONTRACT_Description - *
- @OPSYs_Title - *
- @SYGMA_Date - *

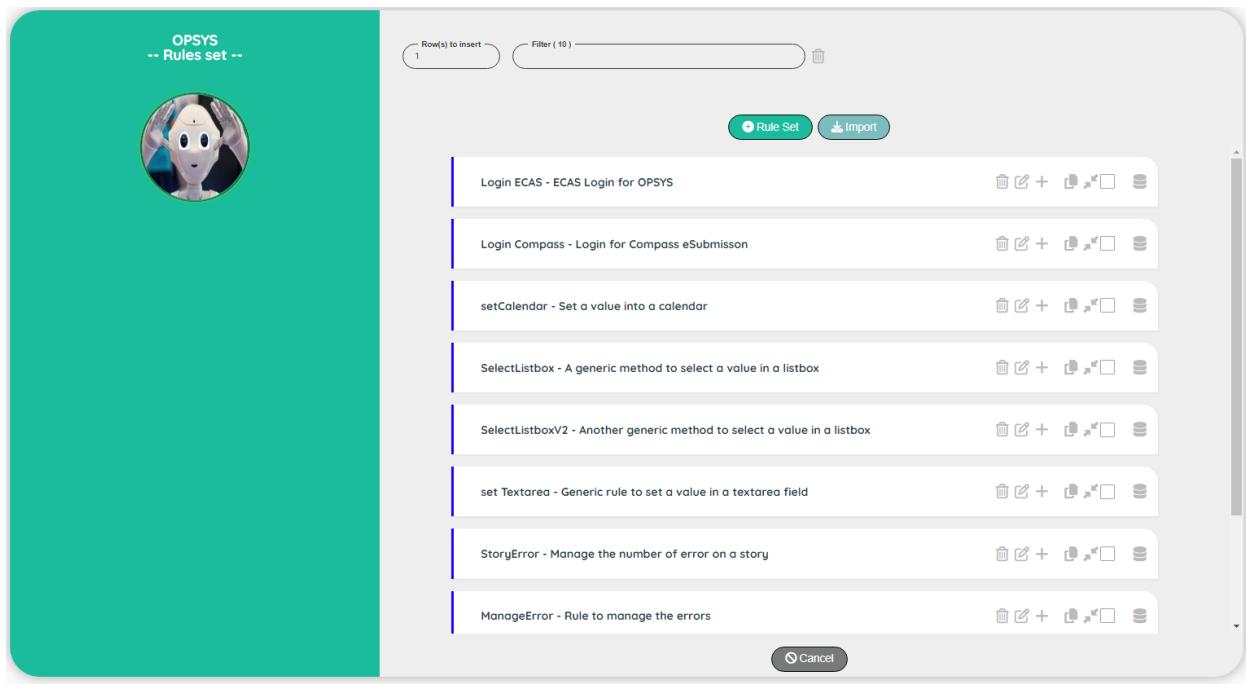
At the bottom right of the main area is a "Cancel" button.

Topic	Icon	Comment
Cancel		Discard the edit

Rules

Rules set

The rules are available for a project. It is composed of a rule set and rules. The rules can be used by the robot to make a decision or by the Designer to define subroutine that can be reused. All the functions are available also in the rules.



Topic	Icon	Comment
Rules		Go to the screen Rules
Add Rule set		Add a rule set at the beginning of the list.
Import rules		Import rules from another project
Cancel		Back to the control panel.

Rules set Edit

OPSYs
-- Rule set --



Created By goffipl on: 01/03/2024 Updated By goffipl on: 26/04/2024

Project OPSYS

Rule set Login ECAS

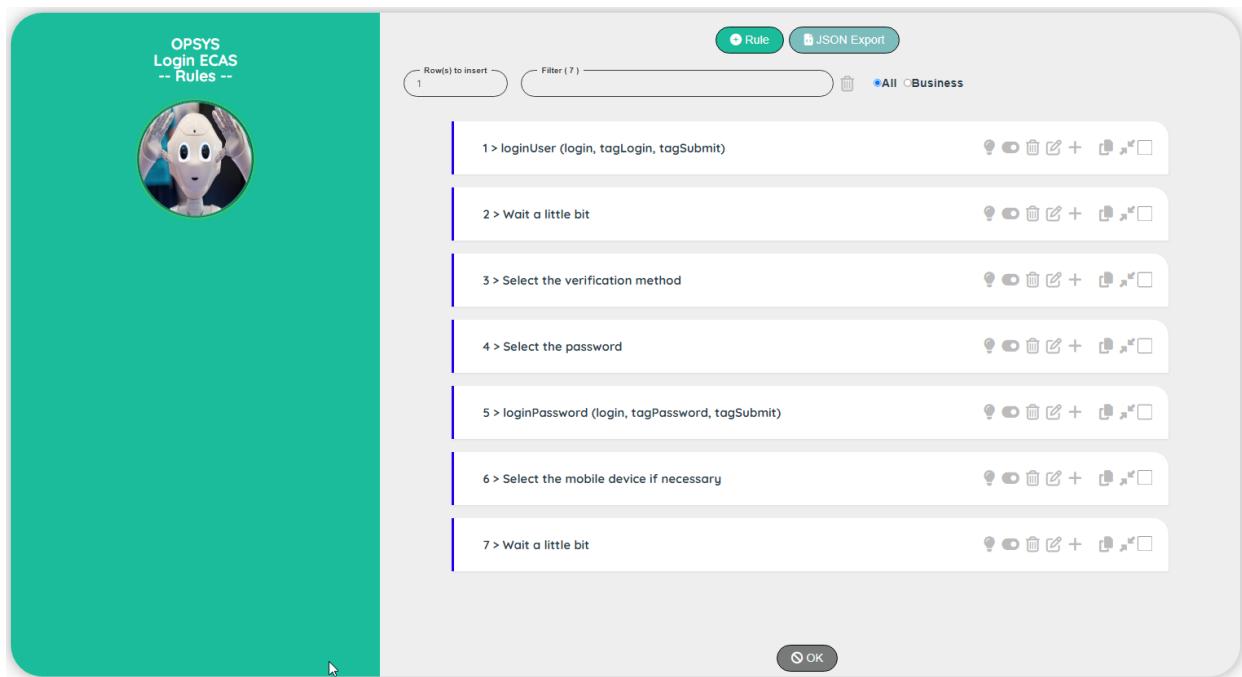
Comment ECAS Login for OPSYS

Status Active

Save **Cancel**

Topic	Icon	Comment
Project		For info: Name of the project
Rule set		Code of the rule set
Active		Active, Not active
Save	Save	Save the edit
Cancel	Cancel	Discard the edit

Rules



Topic	Icon	Comment
Add Rule		Add a rule at the beginning of the list.
JSON Export		Export rules into a .json file
Download		Download the .json file
Active/Inactive		Set the test Inactive/Active
Business/Designer		Set the comment type to Business / Designer (Technical)
Cancel		Back to the rule set.

Note: To view how to upload a '.json' file into an Excel sheet, please refer to the section: [How to process the .json file into Excel](#)

Rules Edit

OPSYS
 -- Rule --



Project OPSYS

Rule Login ECAS

Continue Yes

Condition 1 == 1

Result #loginUser: \$P1, @OPSYs_tagLogin, @OPSYs_tagSubmitLo AB ↴ ⚙️

Message -

Comment loginUser (login, tagLogin, tagSubmit) 💡

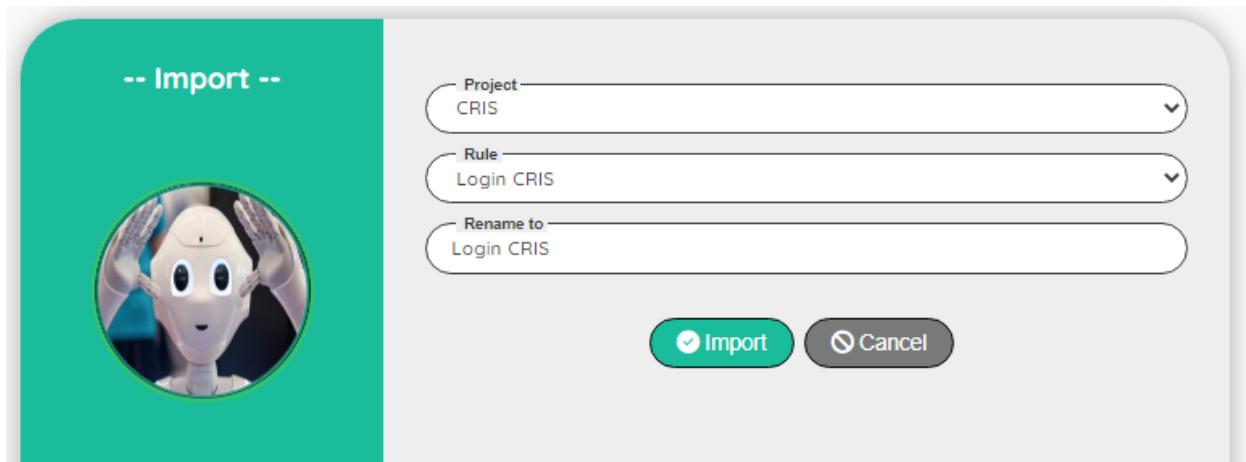
Status Active

Save Cancel

Topic	Icon	Comment
Project		For info: Name of the project
Rule set		For info: Name of the rule set
Continue		Select Yes or Skip
Condition		Any valid JavaScript expression
Result		Any valid JavaScript expression or an existing function with its parameters (function starts with #)
Comment	💡 Comment for the Business ⚙️ Comment for the Designer	Comment on the test
Active		Active, Not active
Save	Save	Save the edit
Cancel	Cancel	Discard the edit

Import rule set (including rules)

This screen will allow you to import the rules from another project



Topic	Icon	Comment
Project		Select a project
Rule set		Select a rule
Rename to		If necessary, rename the rule
Import		Import the rules
Cancel		Discard the import

Loop

Principle

The Loops are very useful when you need to repeat a process.
Example: Upload a set of documents, key a set of experts data...

To define a loop, you need to use the instruction Loop

The screenshot shows a list of steps for defining a loop, each with a set of icons for edit, delete, add, and copy. The steps are:

- 1 > Describe: Use a loop
- 2 > It: possible to use a loop
- 3 > Step: Set the variable with the maximum cycles for the loop1 (setVariable)
- 4 > Loop: through a cycle
- 5 > Step: Display Loop (message)
- 6 > Loop: A sub loop
- 7 > Step: Display a subLoop (message)

Below this list is a detailed configuration panel for a specific loop action:

- Scenario:** Test a simple Loop
- Action:** Loop
- Comment:** through a cycle
- Max Loop:** \$MaxCycle
- Status:** Active

Loop has a parameter: Max Loop to define the number of loops.

Note: If a 'It block' is defined, the loop is inside the block (and so, depends on the condition of a skipIt or a skipDescribe)

You can control the loop thanks to the variable \$Loop or \$Loop1 (it's equivalent).

For instance, you can enter a value with specific data from the dataset like this:

setValue @Portal_LumpSum, #**Dataset_Lump Sum Comment\$Loop** to key data from the dataset:

- #SEA-2023_Lump Sum Comment**1**: This is the first comment
- #SEA-2023_Lump Sum Comment**2**: This is the second comment
- #SEA-2023_Lump Sum Comment**3**: This is the third comment

You can define a nested loop if you define a second Loop inside the first one

In the image above, the line 6 define a nested loop.

If the maximum loop for the first one is 2 and 3 for the second, we will have the following sequence:

(1) → 1.1, 1.2, 1.3 (2) → 2.1, 2.2, 2.3

Break a loop

Note: Be careful, the update of the variable used in the parameter of the Loop (Max Loop) during the process has no effect as the number of loops is evaluated during the definition of the loop. If you need to break a rule in progress, you can use a skipIt (but the loop will continue its cycle without any process)

End of a loop

There are three ways to define the end of a loop:

- 1) There is no more step

As in the image above, the loop 1 and the loop 2 will be defined until the line 7

- 2) There is a Describe

When a Describe is detected, the Robot understand that the current loop is ended.

This is the most common way to indicate the end of a loop

- 3) There is an End Loop

When you have a skipt before the loop, the existence of the loop will depend of the skipt condition. In the case you have nested loop, you cannot use a Describe to end a loop.

In this case, you need to use End Loop. This instruction will be ignore if a skipt is triggered.

Examples of loop

A simple case: One loop inside a scenario

Action	Comment	Loop
Describe	Manage the deliveries	
It	Possible to manage deliveries	
Step	Get the number of deliveries from the dataset into the variable \$NbDelivery	
Loop	Through all the deliveries (max loop = \$NbDelivery)	1
Step	Add a new delivery	1
Step	Enter the name from the dataset using #\$Dataset_Delivery\$Loop	1
It	Possible to continue inside a loop	1
Step	...	1
Describe	Manage the document (End of the loop on the deliveries)	
It	

Two independent loops inside a scenario

Action	Comment	Loop
Describe	Manage the deliveries	
It	Possible to manage deliveries	
Step	Get the number of deliveries from the dataset into the variable \$NbDelivery	
Loop	Through all the deliveries (max loop = \$NbDelivery)	1
Step	Add a new delivery	1
Step	Enter the name from the dataset using #\$Dataset_Delivery\$Loop	1
It	Possible to continue inside a loop	1
Step	...	1
Describe	Manage the document (End of the loop on the deliveries)	
It	Possible to manage document	
Step	Get the number of documents from the dataset into the variable \$NbDocument	
Loop	Through all the documents (max loop = \$NbDocument)	1
Step	Add a new document	1
Step	Enter the name of the document from the dataset using #\$Dataset_Document\$Loop	1
It	Possible to continue inside a loop	1
Step	...	1
Describe	Manage the save data (End of the loop on the document)	
It	

Two nested loops inside a scenario

Action	Comment	Loop
Describe	Manage the contractors	
It	Possible to manage contractors	
Step	Get the number of contractors from the dataset into the variable \$NbContractor	
Loop	Through all the contractors (max loop = \$NbContractor)	1
Step	Get the name of the contractor in the variable \$Name	1
It	Possible to continue inside a loop	1
Step	...	1
It	Possible to manage Experts of the contractor	1
Step	Get the number of experts from the dataset into the variable \$NbExpert	1
Loop	Through all the expert (max loop = \$NbExpert)	2
Step	Add a new Expert	2
Step	Enter the name of the expert from the dataset using #\\$Dataset_Expert\$Loop	2
It	Possible to continue inside a loop	2
Step	...	2
Describe	Save the contractor (end of the loop 2, back to loop 1)	1
It	Possible to save the contractor	1
Step	...	1
Describe	Finalise the test (end of the loop 1)	
It	Possible to finalise the test	

Two nested loops inside a scenario with a conditional skip It (we need to use End Loop)

Action	Comment	Loop
Describe	Manage the contractors	
It	Possible to manage contractors	
Step	Get the number of contractors from the dataset into the variable \$NbContractor	
Loop	Through all the contractors (max loop = \$NbContractor)	1
Step	Get the name of the contractor in the variable \$Name	1
It	Possible to continue inside a loop	1
Step	...	1
It	Possible to manage Experts of the contractor	1
Step	Skip the test if the contractor is 'ArtComputer' (using \$Name)	1
Step	Get the number of experts from the dataset into the variable \$NbExpert	1
Loop	Through all the expert (max loop = \$NbExpert)	2
Step	Add a new Expert	2
Step	Enter the name of the expert from the dataset using #\\$Dataset_Expert\$Loop	2
End Loop	End of the loop 2, back to loop 1	1
It	Possible to save the contractor	1
Step	...	1
Describe	Finalise the test (end of the loop 1)	
It	Possible to finalise the test	

Performance

Principle

The Robot is not designed to measure the performance of an application (like load runner for instance) however, you can measure the elapsed time between two points in the application.

You have also the possibility to compare the last measure with the average of the last 10 tests. It can be useful after a new deployment for instance.

You can set as many timers you want in a scenario by respecting the following rules:

- Each timer has a unique identifier (topic).
- The identifier used during the closure must be equal to the identifier of the start.

In the dashboard, you have the possibility to display the performance of a story or focus on the performance of a specific step in a story.

To measure elapsed time, you can use the following functions:

startTimer	topic	Start a timer to measure the
stopTimer	environment	topic

Define the measure

If your scenario is well structured, it should be very easy to identify the bloc of actions thanks to the “Describe” and the “It”.

If you want to measure the global performance of a test, I suggest to start the measure, just after the login.

Note: If you want to measure the performance of the login, take into account that the measure is done by scenario. You cannot display a global measure of the login of your application!

To indicate the starting point of the measure, use the function `startTimer()` with the name of the topic (a unique identifier). Keep the name short!

Note: The topic is case-insensitive.

To indicate the end of the measure, use the function `stopTimer()` with the parameters: Environment (Example: ACC, TEST or PROD) and the name of the topic used in the `startTime` function.

Warning: be careful when using the function `pause()` because you will increase artificially the performance.

Also, increasing a pause of an existing scenario will impact the global performance!

Mechanism

The stopTimer will measure the elapsed time and store the value in the database.

The Robot will perform the following tasks:

Shift the 10 existing measures up and store the elapsed time at the measure 11

ID	T1	T2	T3	T4	...	T11	T12	T13	...
1	M1	M1	M1	M1		M1	M2	M3	
2			M2	M2		M2	M3	M4	
3				M3		M3	M4	M5	
4						M4	M5	M6	
5						M5	M6	M7	
6						M6	M7	M8	
7						M7	M8	M9	
8						M8	M9	M10	
9						M9	M10	M11	
10						M10	M11	M12	
11	M1	M2	M3	M4		M11	M12	M13	

With Txx: Timing and Mxx: Measure

Note: we use a cheat with the first and the second measure, to keep the coherence of the data

Using this mechanism will reduce excessive measurement variations.

However, be careful when you need to perform a measure after a new deployment, if the elapsed time is suddenly high, the impact will not be very visible at first!

Example:

ID	T11	T12	T13	T14	T15	T16	T17	T18
1	2,0	2,1	2,2	2,1	2,2	2,3	2,0	2,1
2	2,1	2,2	2,1	2,2	2,3	2,0	2,1	2,2
3	2,2	2,1	2,2	2,3	2,0	2,1	2,2	2,1
4	2,1	2,2	2,3	2,0	2,1	2,2	2,1	2,2
5	2,2	2,3	2,0	2,1	2,2	2,1	2,2	5,0
6	2,3	2,0	2,1	2,2	2,1	2,2	5,0	5,1
7	2,0	2,1	2,2	2,1	2,2	5,0	5,1	5,0
8	2,1	2,2	2,1	2,2	5,0	5,1	5,0	5,1
9	2,2	2,1	2,2	5,0	5,1	5,0	5,1	5,1
10	2,1	2,2	5,0	5,1	5,0	5,1	5,1	5,1
11	2,2	5,0	5,1	5,0	5,1	5,1	5,1	5,1
Avg	2,1	2,2	2,4	2,7	3,2	3,5	3,7	4,0



To quickly highlight the variation, we need to show on the same graph the average and the last measure.

As you can see on the graph, at T12 we suddenly increase the elapsed time!

Performance with the Robot

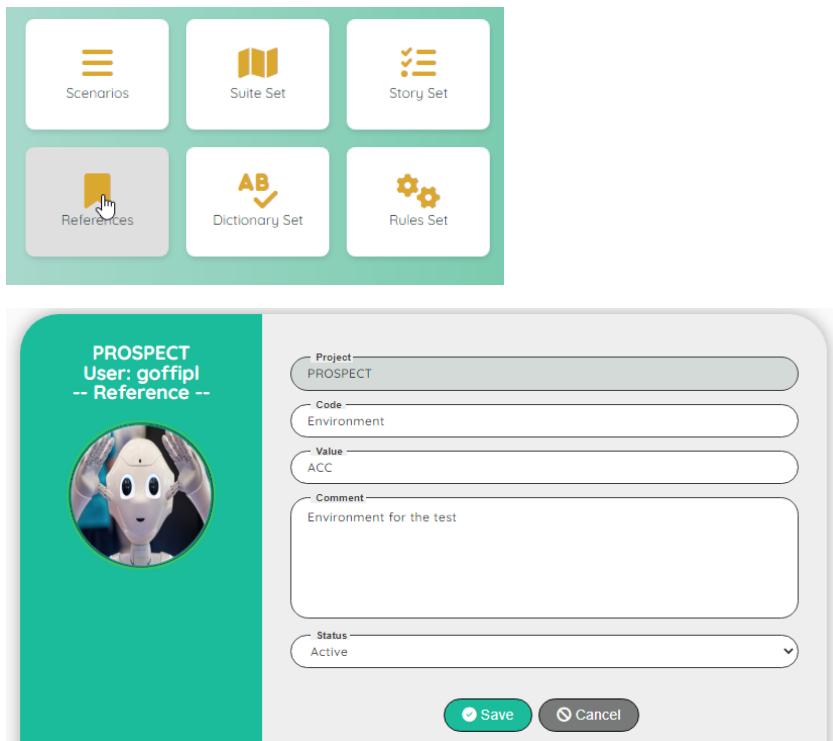
You can visualize the performances of a scenario in the Dashboard.

At the upper left, you can see an icon with a graph, each click will display a different graph.

Dashboard	 Status	Graph of the execution of the story (status)
Dashboard	 Story	Graph of the performance of all the scenarios of the story
Dashboard	 Step	Graph of the performance of all the scenarios of a specific step of a story

Note 1: The step is a kind of zoom on the graph to highlight specific scenarios.

Note 2: In the Dashboard, the Robot can use the value 'Environment' of the reference data. Be sure that you have defined the correct environment before displaying the performance! If no reference 'Environment' can be found, you will not be able to select the performance graph!

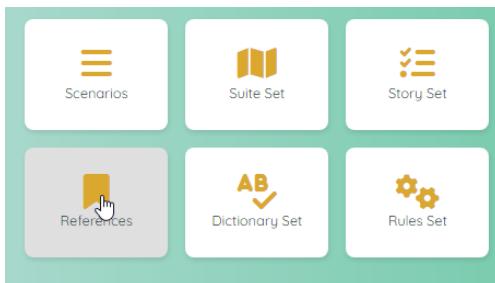


The screenshot shows the PROSPECT application interface. At the top, there is a green navigation bar with several icons: Scenarios, Suite Set, Story Set, References, Dictionary Set, and Rules Set. Below this, on the left, is a teal sidebar containing a user profile picture of a robot, the text "PROSPECT User: goffipl -- Reference --", and a "Save" button. The main area is a white card with the following fields:

- Project: PROSPECT
- Code: Environment
- Value: ACC
- Comment: Environment for the test
- Status: Active

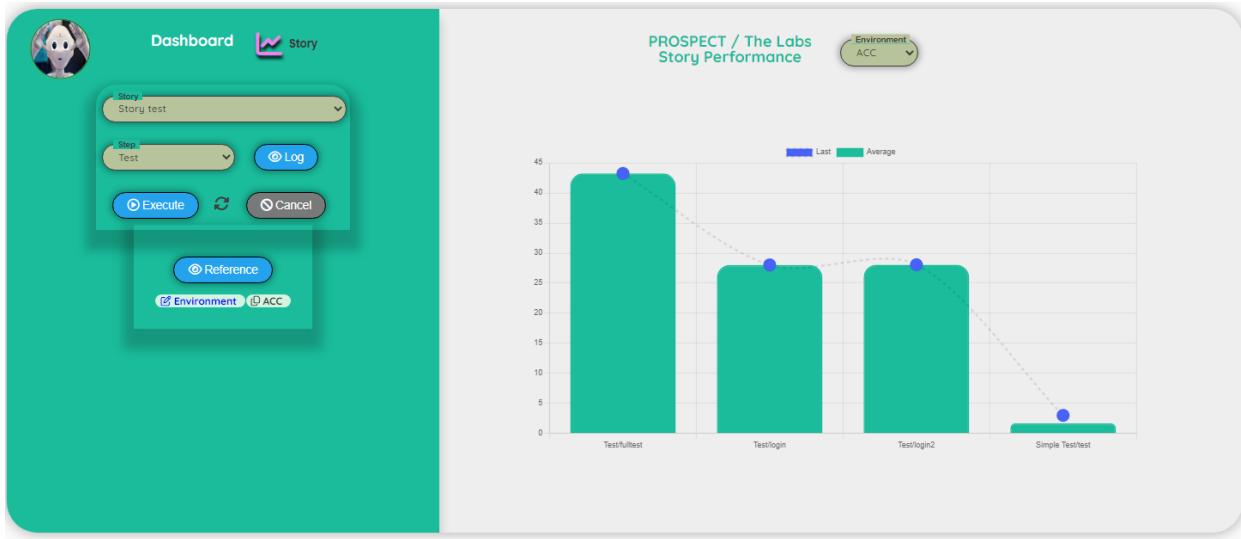
At the bottom of the card are "Save" and "Cancel" buttons.

Note 3: In the Dashboard, the Robot can use the value 'AllEnvironments' of the reference data. If the reference doesn't exist, there is no error, but the list will be limited to the current environment.



Project	PROSPECT
Code	AllEnvironments
Value	ACC,TEST,PROD
Comment	All the Environments used for the performance
Status	Active

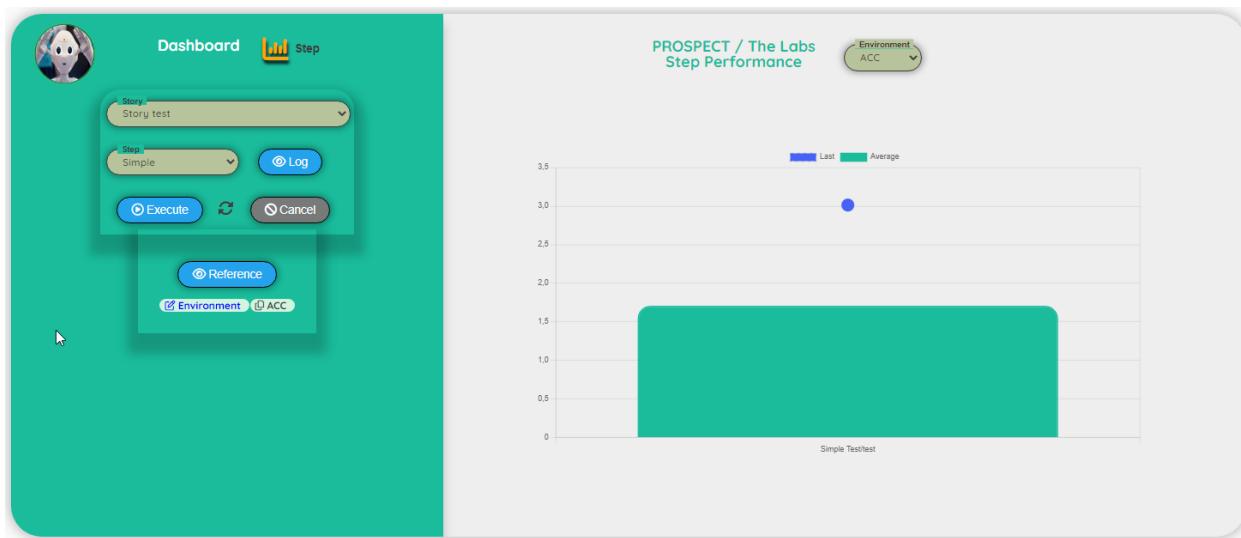
Example: Performance of the story



At this point the selection of a specific step has no impact on the graph.
The first three bars represent the step: 'Test' and the last one, the step: 'Simple'.

Example: Performance of the step.

In this case, we decided to highlight the step: Simple to have a better view of the gap between the average and the last measure.
The step graph will act as a zoom.

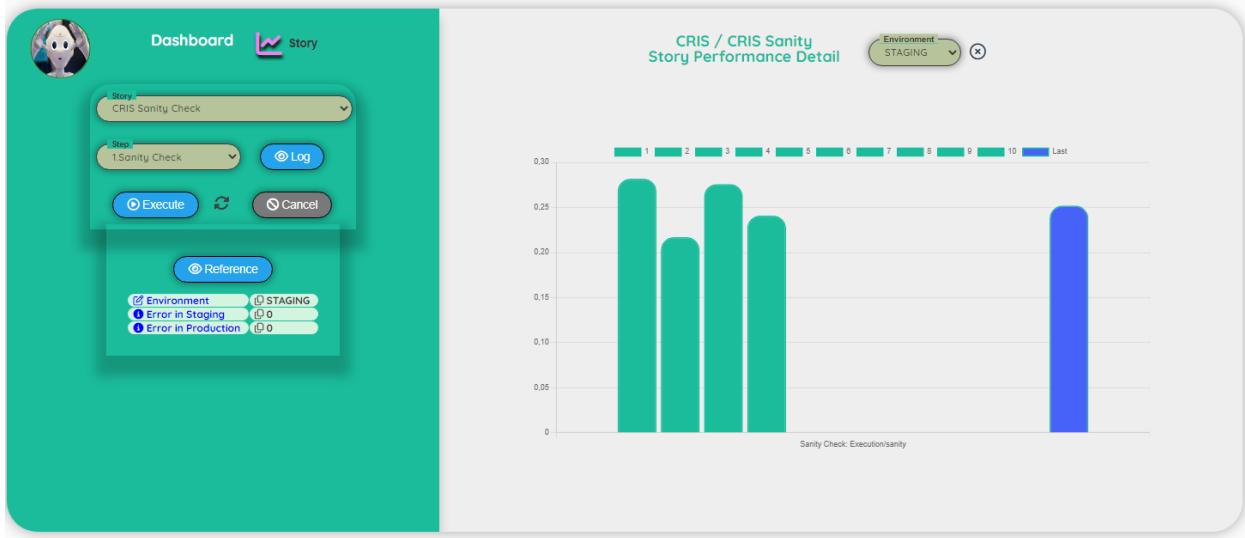


Example: Performance detail (story or step).

This graph is useful to view the variance of the measures.

As we use up to 10 measures to compute the average, it's also important to check the coherence of the measures. Although, the quality of the average will be better if you have already run the test a few time.

From 1 to 10 (in green), the data that will be used for the computation of the average.
In blue the last measure recorded.



You can change the environment to get a quick comparison of the performances.

In this case the performances in PROD



Performance with Admin right

If you are connected to the application as Admin, you can export the performance data into a Json file.

The screenshot shows a modal window titled "Playwright -- Performance --". On the left, there is a circular profile picture of a white humanoid robot. At the top right, there is a "JSON Export" button. Below it is a search bar with the placeholder "Filter (\$)" and the text "javd" entered. The main area contains a list of six test results, each with a checkbox and a trash bin icon:

- ACC > Test iFrame Java > 01-one ==> Measure: 00) 4.073 Created on : 05/06/2025
- ACC > Test iFrame Java > 01-one ==> Measure: 11) 4.073 Created on : 05/06/2025
- ACC > Test iFrame Java > 02-two ==> Measure: 00) 2.057 Created on : 05/06/2025
- ACC > Test iFrame Java > 02-two ==> Measure: 11) 2.057 Created on : 05/06/2025
- ACC > Test iFrame Java > 03-three ==> Measure: 00) 2.078 Created on : 05/06/2025
- ACC > Test iFrame Java > 03-three ==> Measure: 11) 2.078 Created on : 05/06/2025

At the bottom right of the modal is a "Cancel" button.

The .Json file has the following structure.

You can see how to import a .Json file into Excel in the Chapter: Test / How to process the .json file into Excel

```
{
  "Performance": [
    {
      "performanceID": 111,
      "projectID": 85,
      "project": "PROSPECT",
      "scenarioID": 105,
      "scenario": "Test",
      "space": "ACC",
      "topic": "fulltest",
      "created": "11/12/2024",
      "sequenceID": 1,
      "sequenceID2": "01",
      "measure": 43.236
    }
  ]
}
```

Example of graphs of performance designed in Excel

This was designed for a previous project to show the evolution of the performances in the different business categories.

