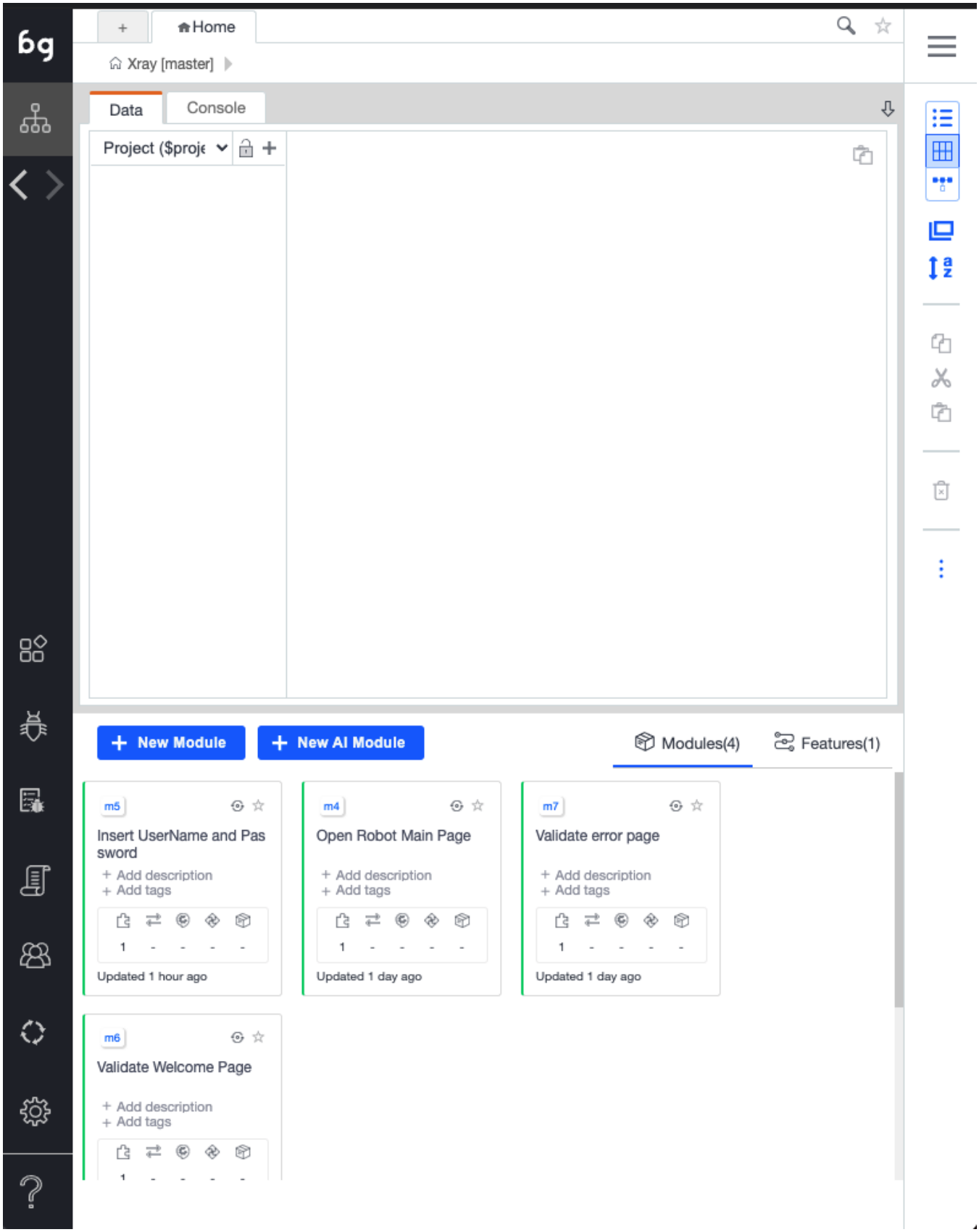
Integration with Boozang

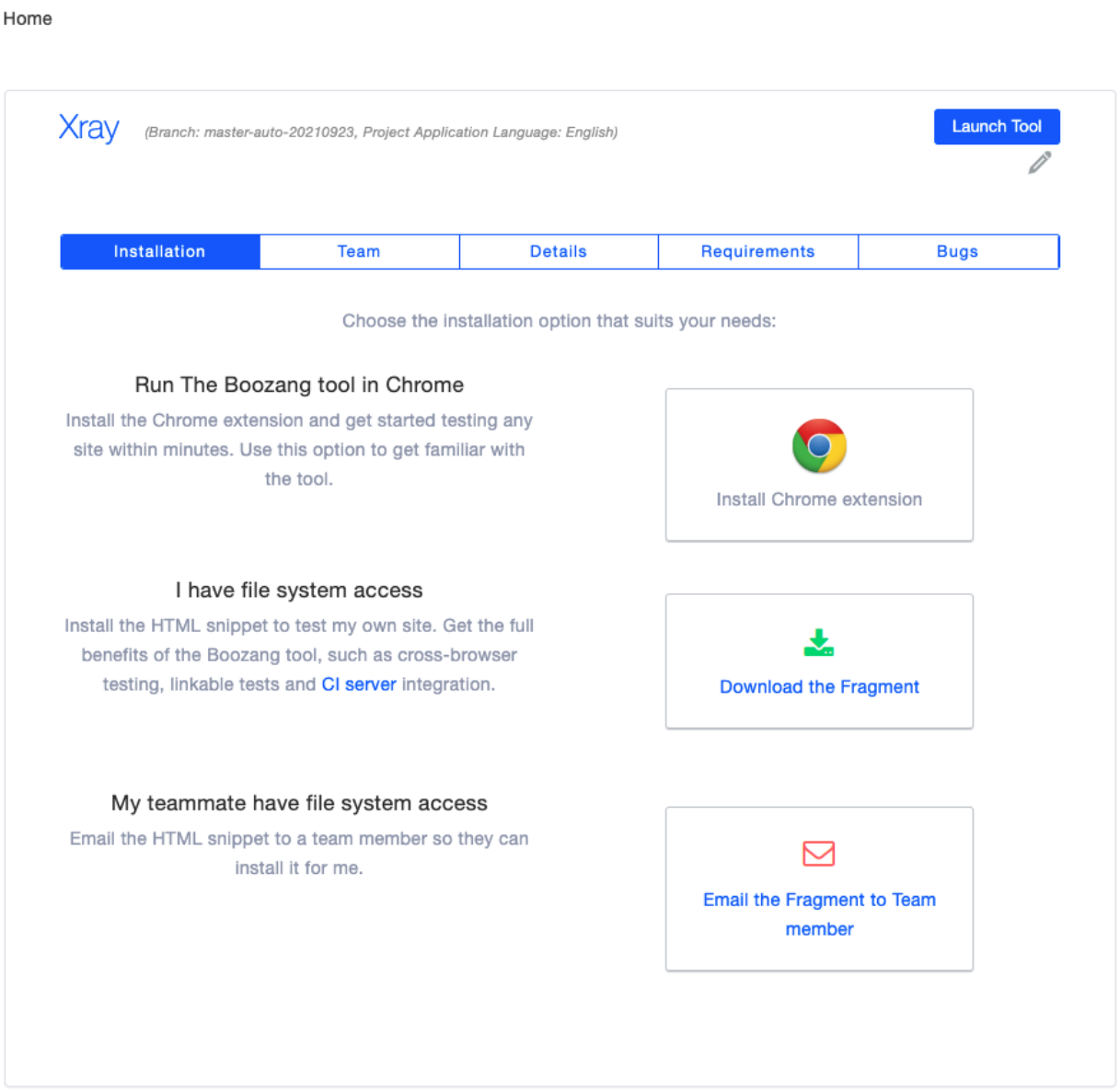
Overview

Boozang is a codeless testing tool that allows you to define and execute UI/API automated tests without the need to code them. It also supports Cucumber tests and has a modular approach towards testing.

Integration with CI/CD tools is also possible through scripts generated in Boozang to be used in your CI/CD tool.

More details about Boozang here.


Home
Xray [master]
Data
Console
Project ($proje
New Module
New AI Module
Modules(4)
Features(1)
m5
Insert UserName and Pas sword
+ Add description
+ Add tags
1 - - - -
Updated 1 hour ago
m4
Open Robot Main Page
+ Add description
+ Add tags
1 - - - -
Updated 1 day ago
m7
Validate error page
+ Add description
+ Add tags
1 - - - -
Updated 1 day ago
m6
Validate Welcome Page
+ Add description
+ Add tags



Main features

This integration provides:

- Integration with Xray cloud or Server/DC
- Ability to export Cucumber tests from Xray to Boozang
- Automate the tests in Boozang
- Import the automation results back to Xray

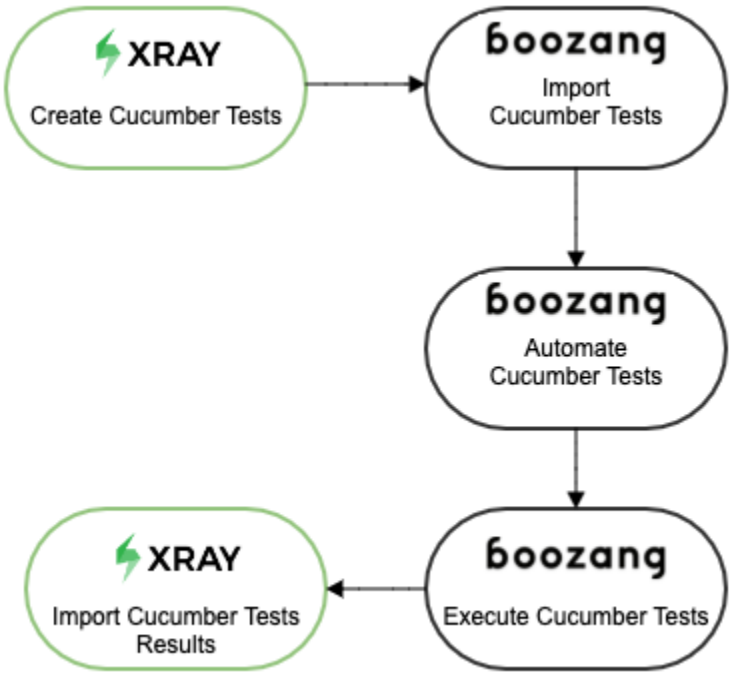
Mapping of concepts

Boozang	Xray
Test (Cucumber)	Test Case (Cucumber)
Test Execution	Test Execution

Flow

If you are using Xray as the master of information (i.e. defining your Cucumber tests and writing those in Gherkin with Xray), then you will import that specification into Boozang to automate the steps and execute them.

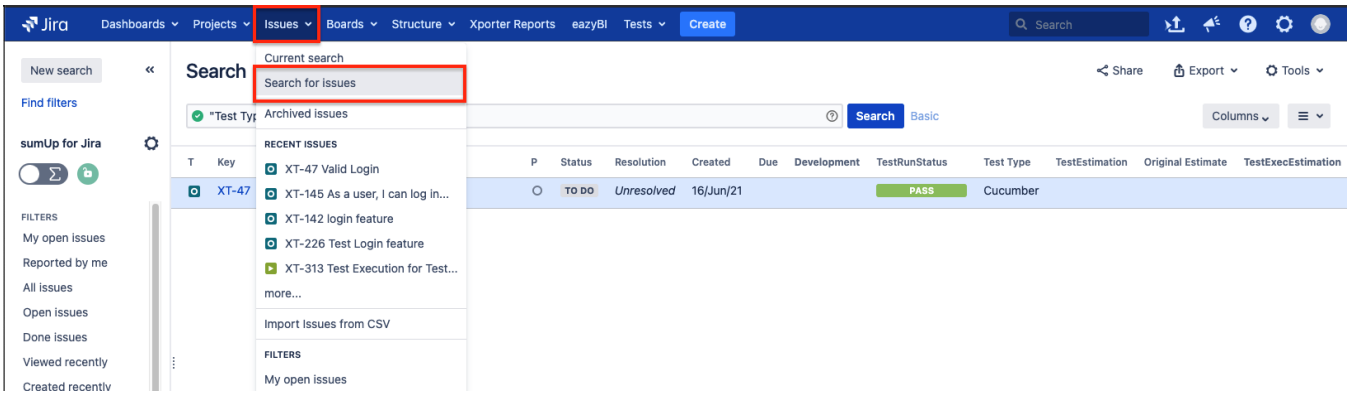
Once the execution is done you will import the test results back to Xray; so the flow will be the below one:



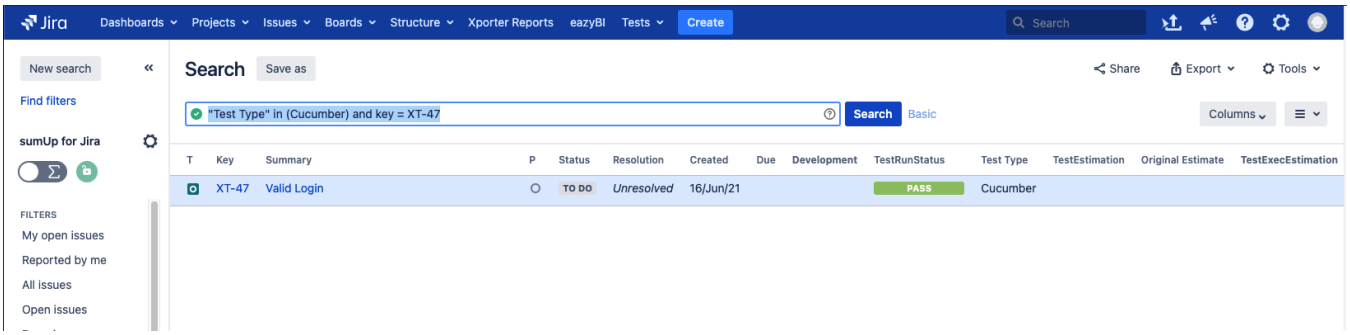
Setup

Start by having your Cucumber Tests defined in Xray. If you do not have those defined yet, you can follow this article in order to add your Cucumber tests.

The second step is to define a JQL query, in Jira/Xray, to select those tests in order to be imported into Boozang - the integration is based on this filter. To do so select "*Search for issues*" from the "*Filters*" entry in the top menu

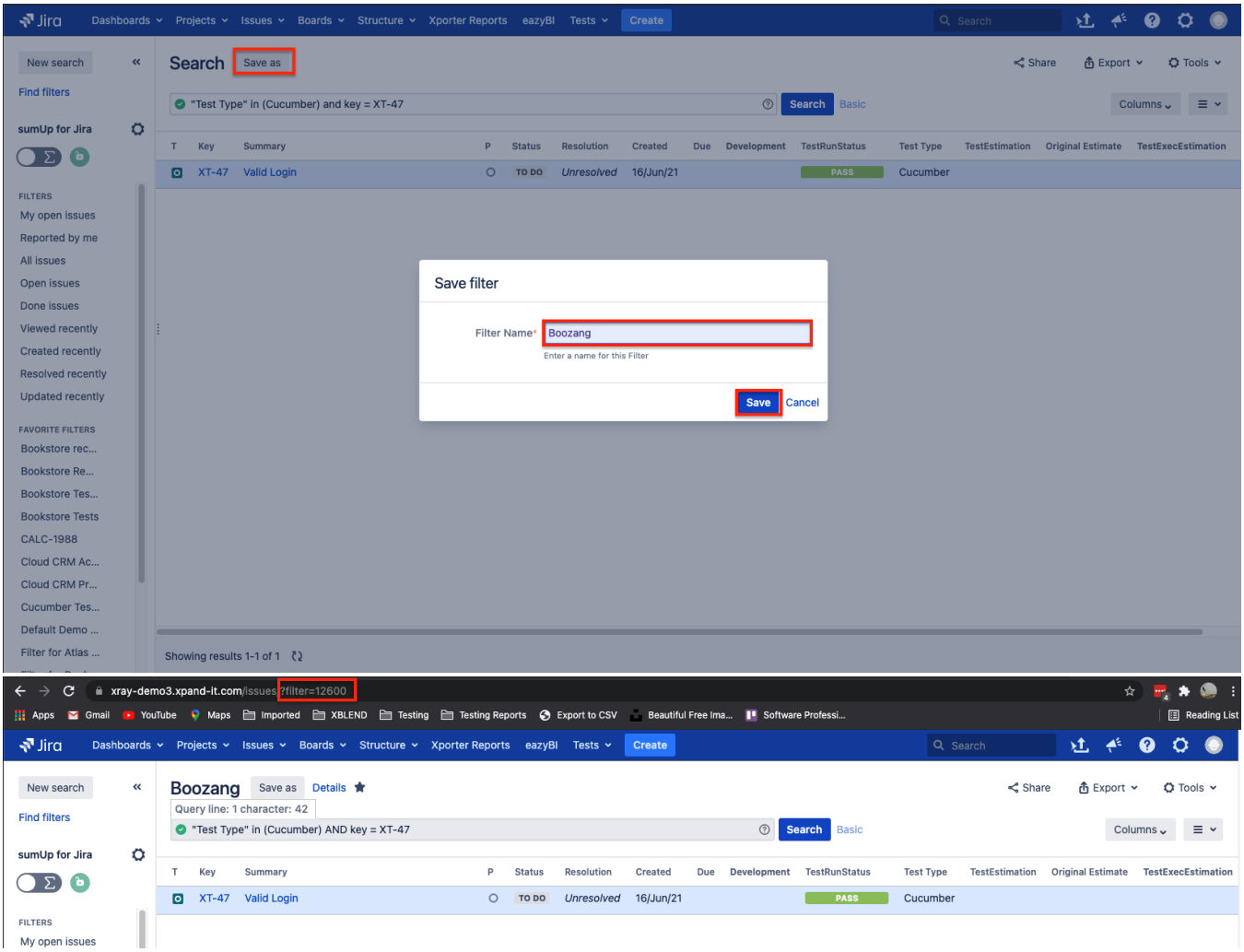


Introduce the query that will allow you to select the Cucumber Tests that you want to export to Boozang (in order to automate those), in our case it will look like this:



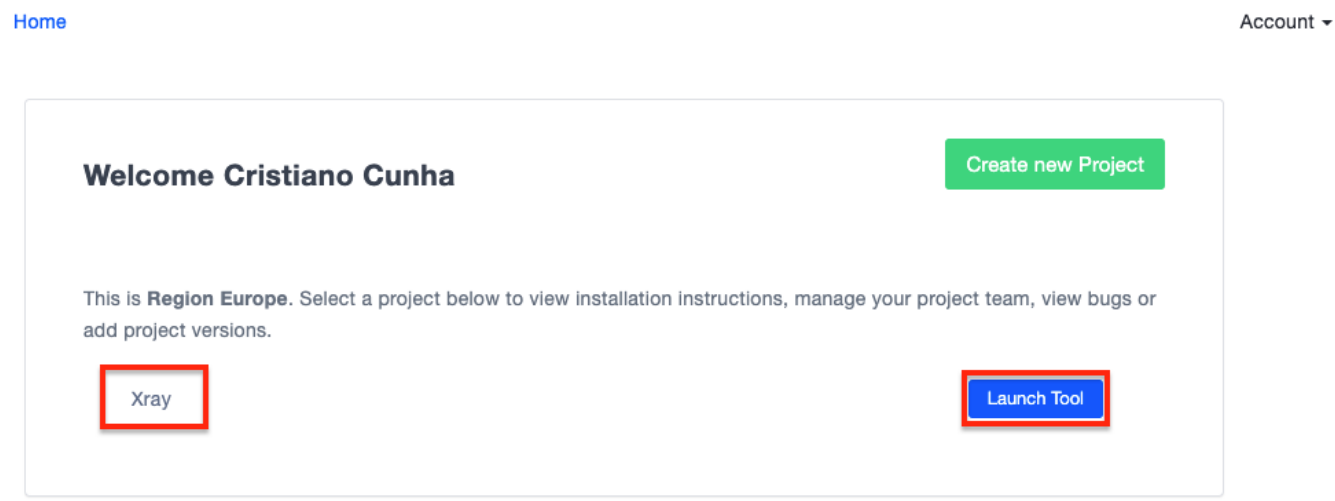
Once you have the filter defined save it using the "*Save as*" option.

Provide a name and save the id that Jira will associate to your filter. This is especially important as we will use this id in the integration with Boozang to import the Cucumber Tests.

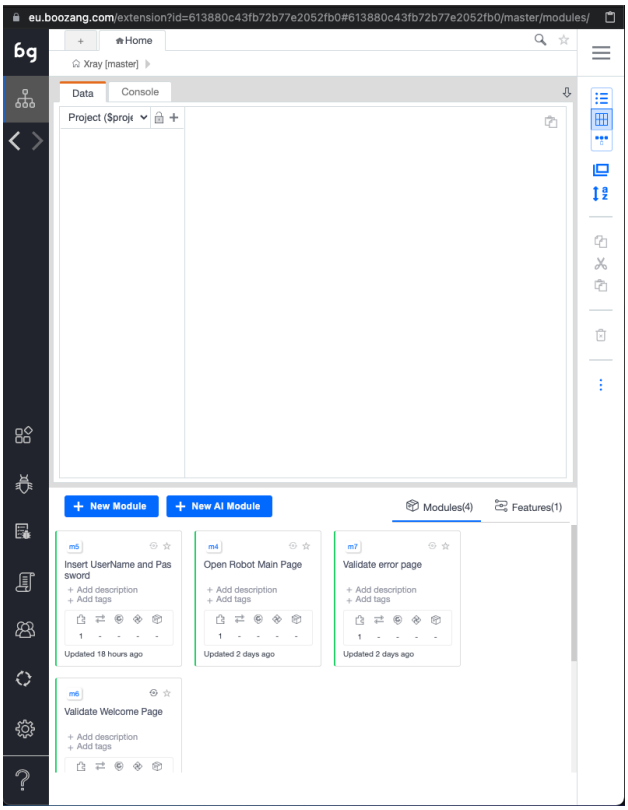


At this stage we have defined the Cucumber Tests in Xray and created a filter to extract those fields, so it is now time to switch to the Boozang application.

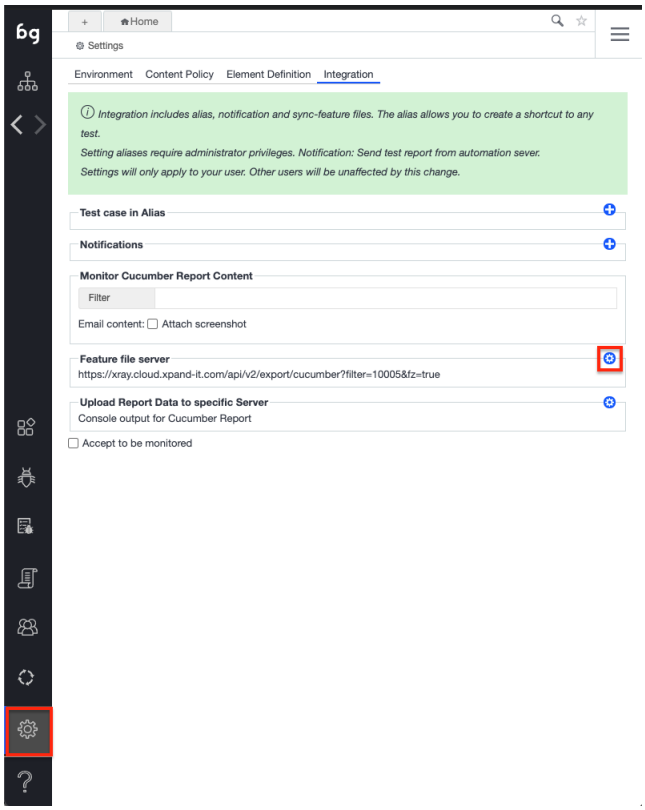
Access Boozang via your region and create a new project designated for your test automation. In the example below, an Xray project was created. Next, open the Boozang tool by selecting the "*Launch Tool*" option:



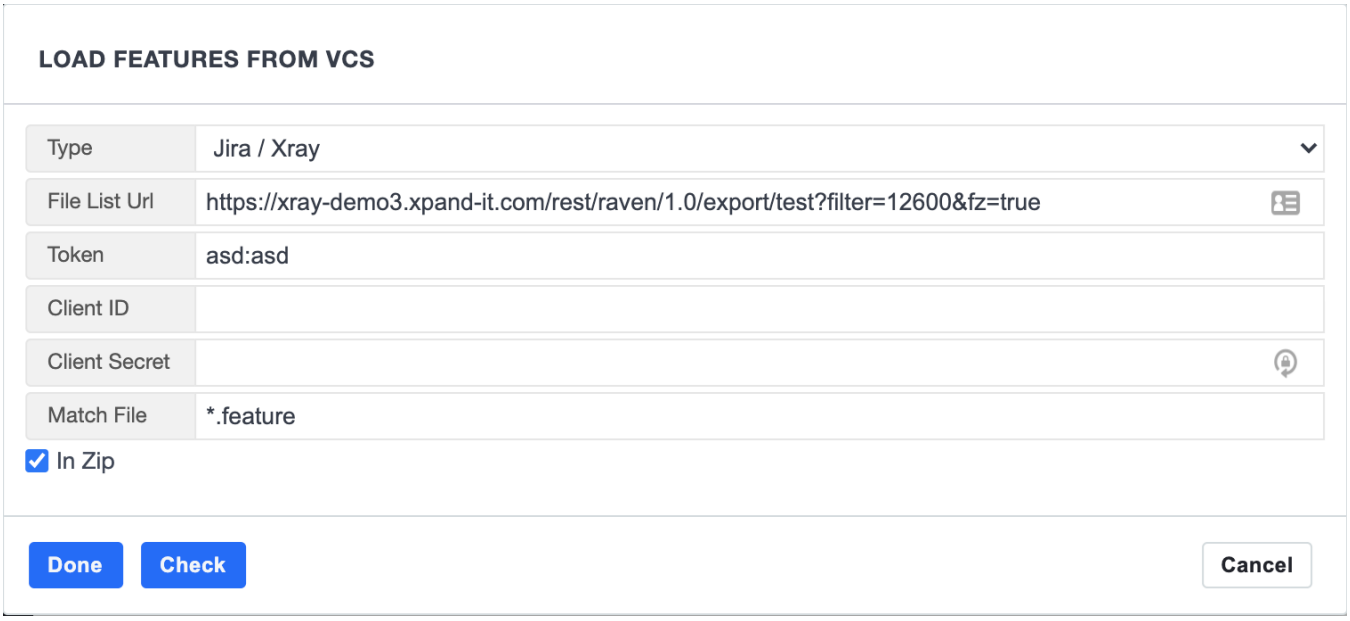
This will open the tool unless it's the first time you press this option. If so, it will take you to the installation option of the Boozang AI Chrome extension and after you've installed it you'll be taken to the tool.



In order to configure the integration with Xray, we must access the main page and select the configuration option in the bottom left-hand corner and then configure the *Feature File Server* by clicking on the configuration icon next to it



This will open a configuration pop up with the different integrations available in Boozang, in our case we are going to choose the "*Jira/Xray*" option

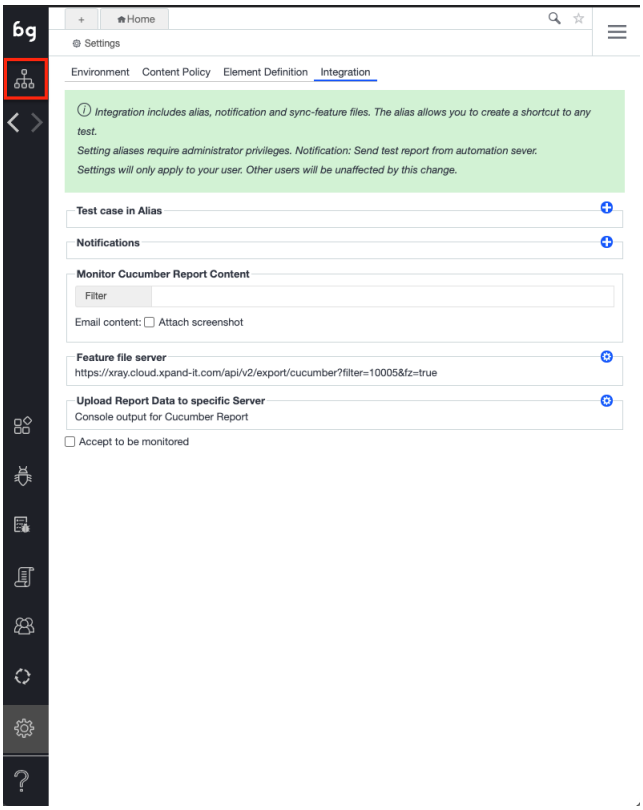


In more detail:

- Type: Choose Jira/Xray option
- File List Url: Replace by your Jira instance URL plus "/rest/raven/1.0/export/test?filter" and the JQL filter that we have saved in Jira/Xray
- Token: Insert the username and password for your Jira instance
- Client ID: Leave blank
- Client Secret: Leave blank
- Match File: Leave the "*.feature"

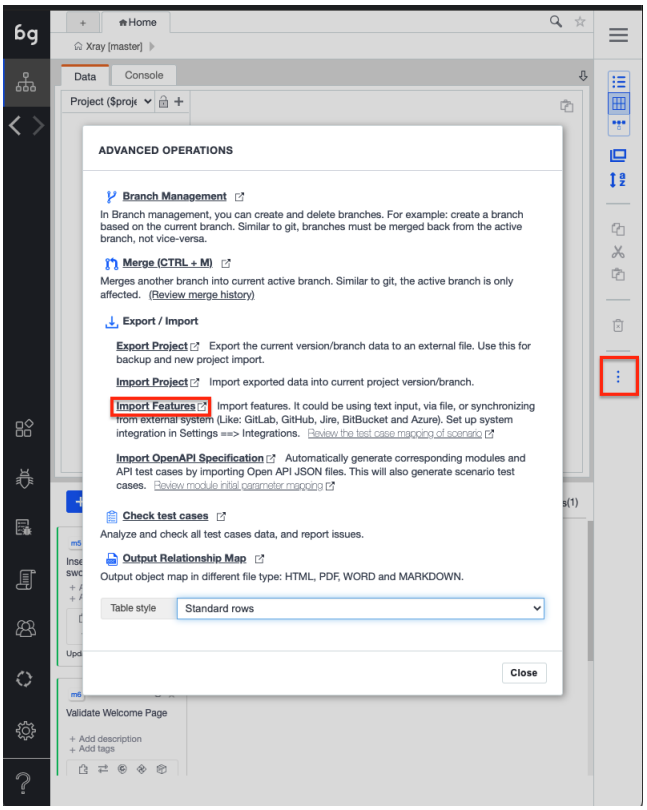
You can select Check to validate that the configuration is OK, and once everything is done click on *Done*.

Now get back to the main page of Boozang application by selecting the first option in the left-hand side menu.

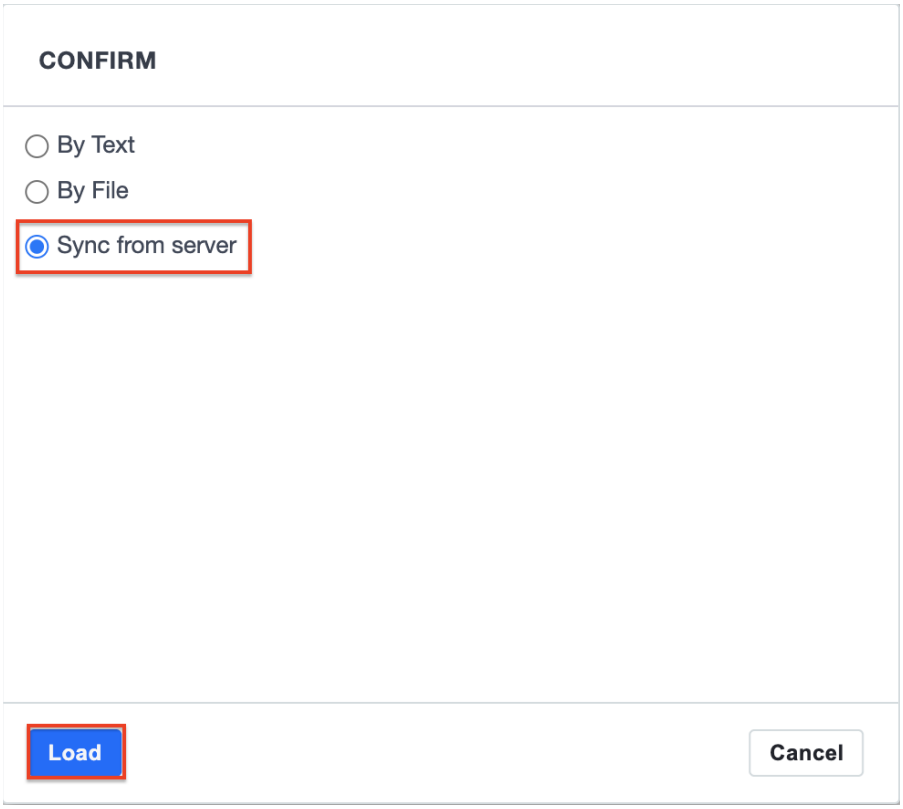


Import Xray Cucumber Tests into Boozang

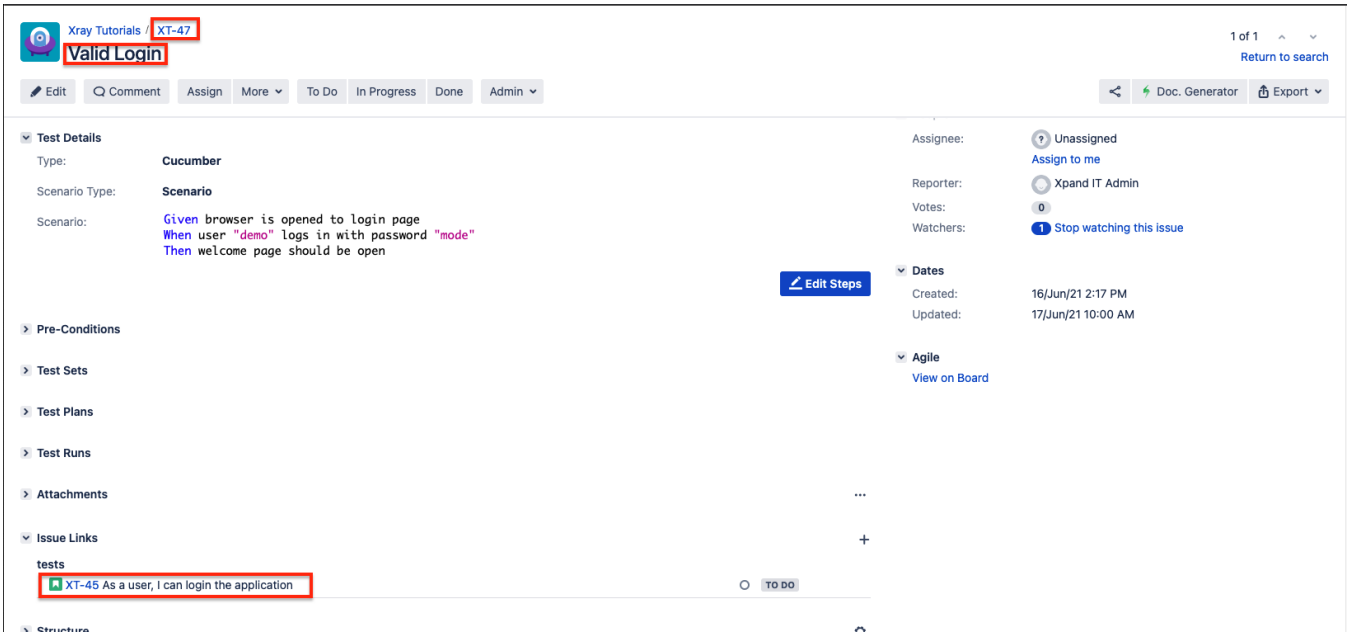
Now that the configuration is set, we can export the Cucumber tests from Xray and import them into Boozang. To do so, select the option in the right-hand side menu as shown in the screenshot below, and choose "*Import Features*" option in the pop up that comes up.



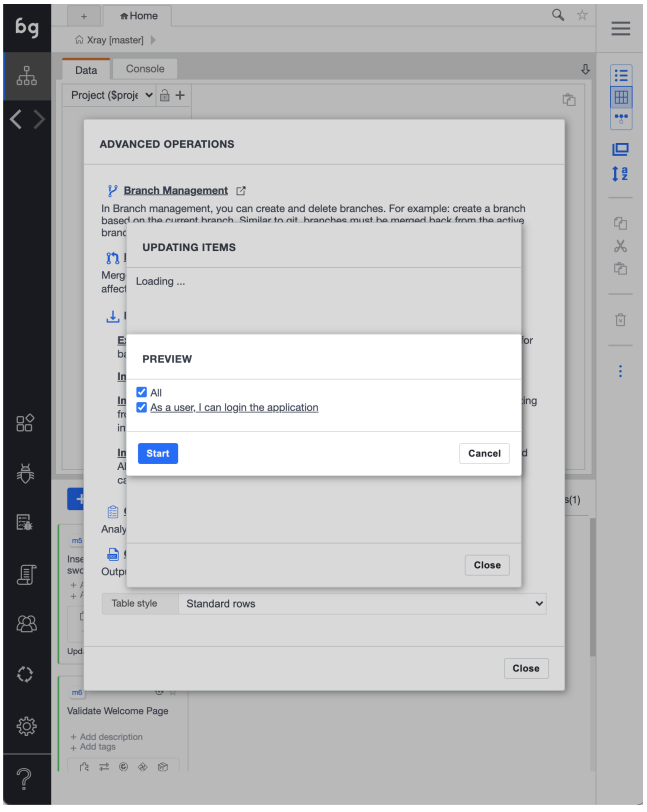
Another pop up will appear asking you to choose the import type, in our case, we will choose "*Sync from server*" and select *Load.*



At this stage Boozang will connect to Xray and list all the requirements that are covered by the Cucumber tests present in the filter you have provided. In this case, two Cucumber Tests are covering the XT-45 User Story.

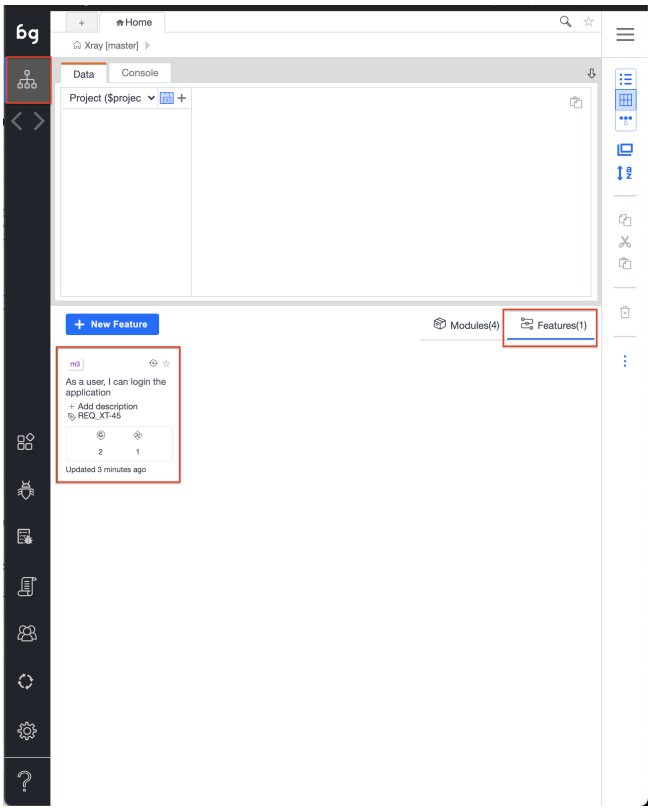


The pop up it will show a preview of the requirement and where it will import the Tests from.

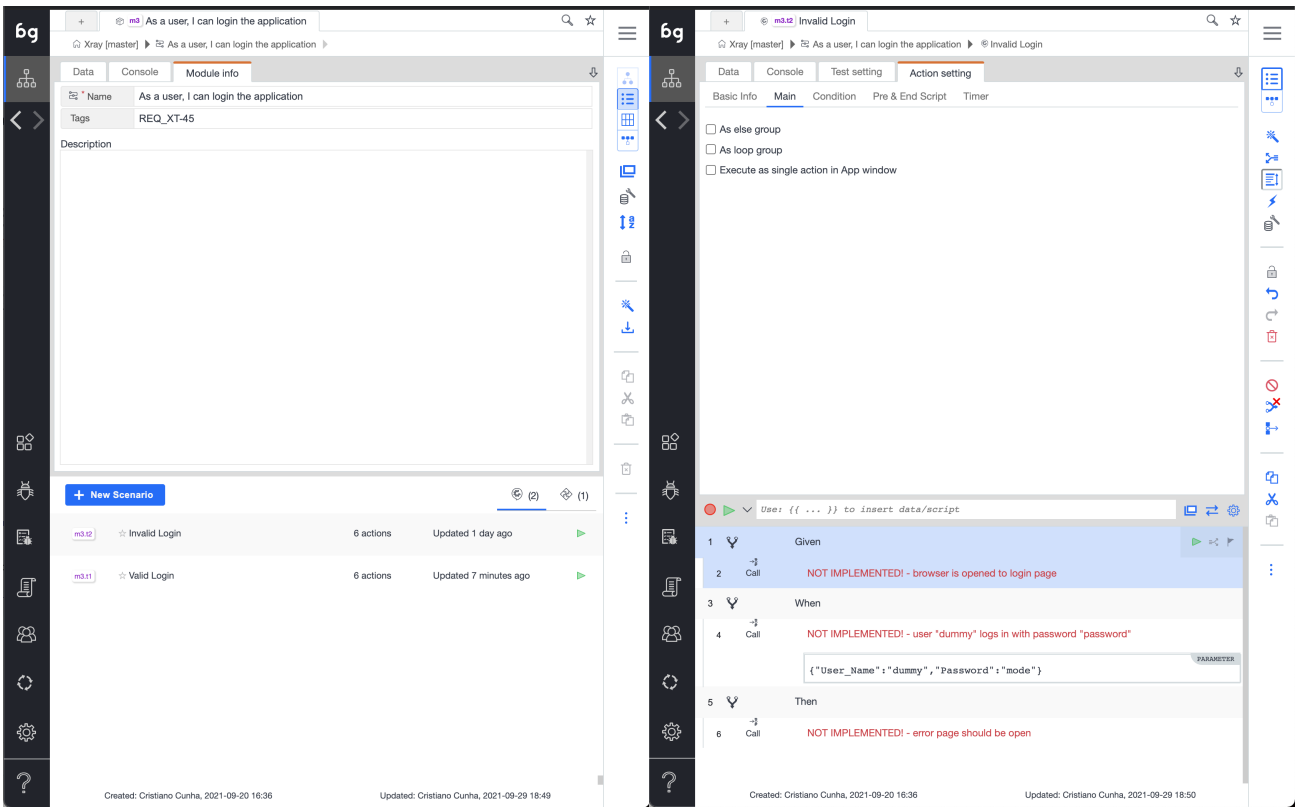


When selecting "*Start*", Boozang will show you the items it will create and proceed with the creation of the features.

Now that the features have been imported into Boozang, the next step would be to implement the code that will be executed to perform the actual validation. Head back to the main page and review the Features created.



When clicking on *Feature,* the actual Tests that were imported will come up, and selecting either of them will show the detailed view of the Tests with the steps that need to be automated - displayed in red below.



Automating Cucumber Tests in Boozang

There are several ways to automate tests and link them to steps in Boozang however for this section, we will focus on one. For other options more suited to your requirements, we suggest taking a look at Boozang's documentation.

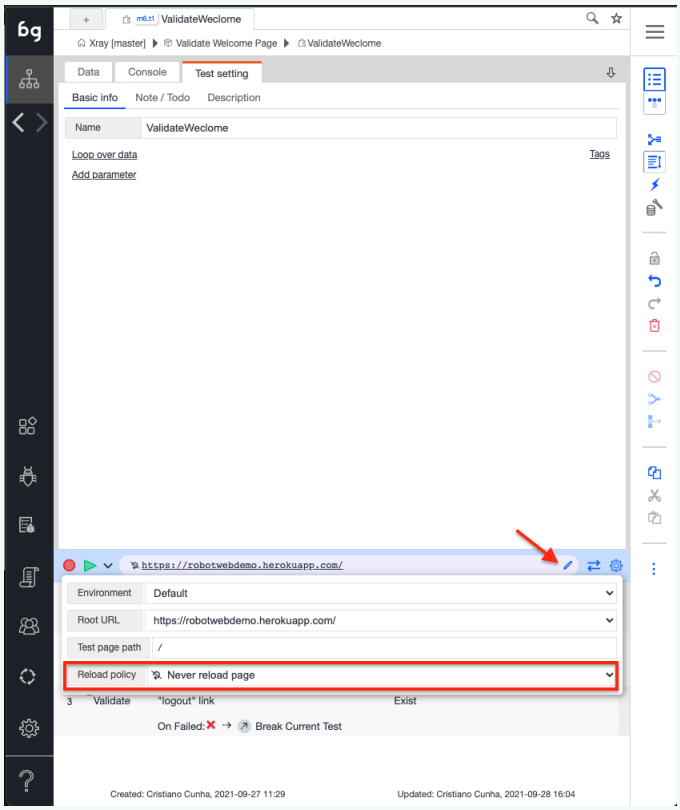
For this example, what matters is to have the steps automated so that they can be executed and produce a report of the execution which will be imported into Xray.

The approach is to create new Modules with a test in them that will represent each step in the Cucumber scenario, that way Tests can be reused in other scenarios with the same description.

You can use the recording capability of Boozang to create each Test. The recorder allows you to define the Test steps and the validations required for each Test.

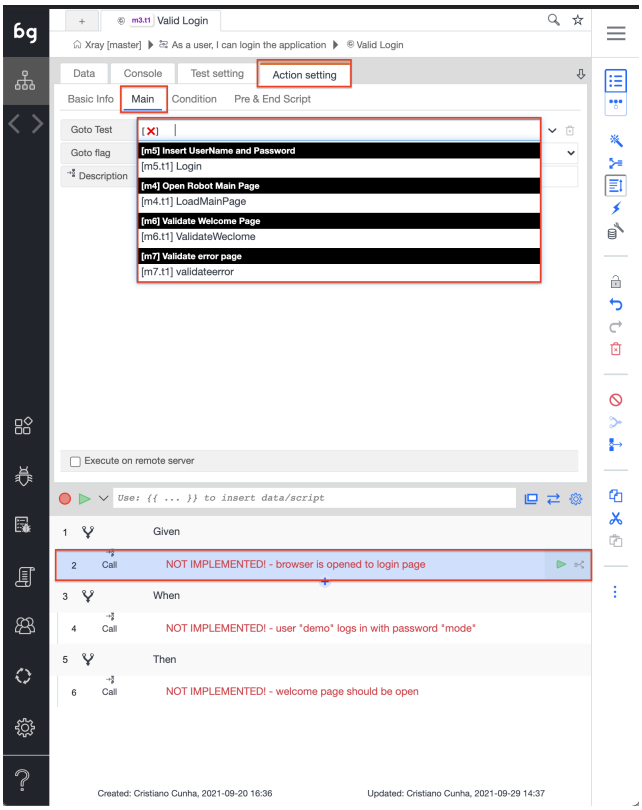
Tip

In order to reuse some Test steps, you must assure that when the Test starts it will not reload the URL but start where the last Test have ended. To do that, set the Parameter inside the "*Reload Policy*" of the detail of the Test to "*Never*".

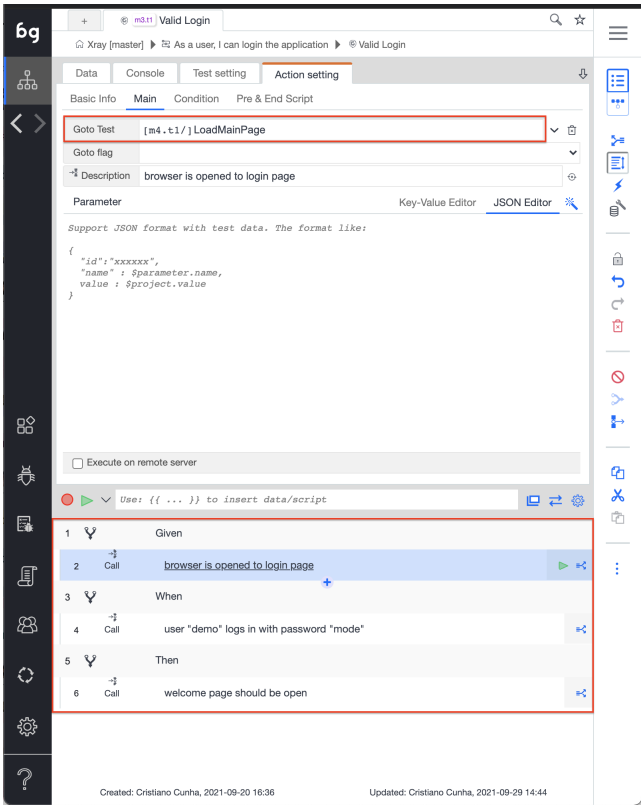


Once the Modules and Tests are ready, they'll need to be linked to each Step in the Cucumber Test. In order to achieve this, access the details of each Cucumber Test and click in the "*NOT IMPLEMENTED! -...*" which will load in the upper part of the application, under the *Main* submenu of the "*Action Setting*" tab.

Click on the "*Goto Test*" entry which Test will be associated to this step via a dropdown available with a list of tests to choose from:

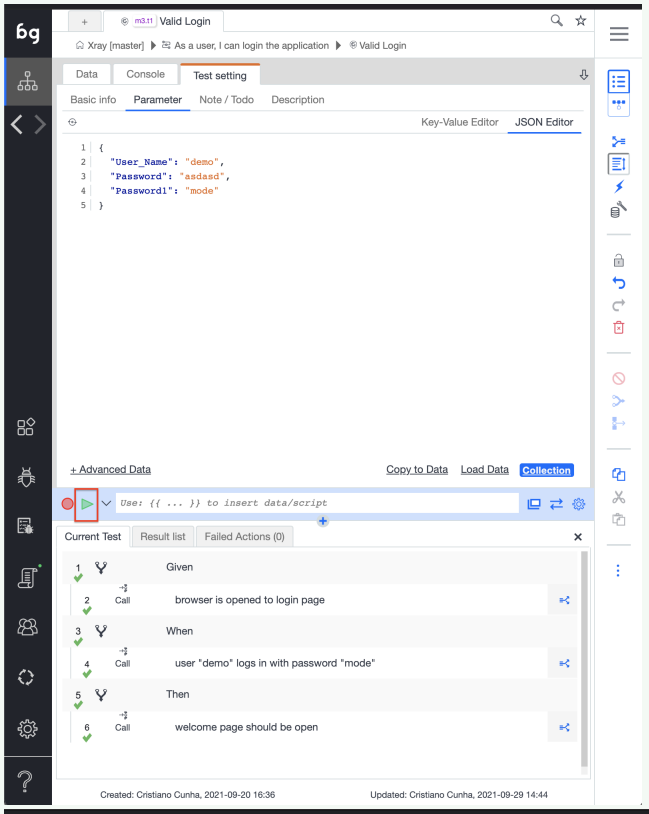


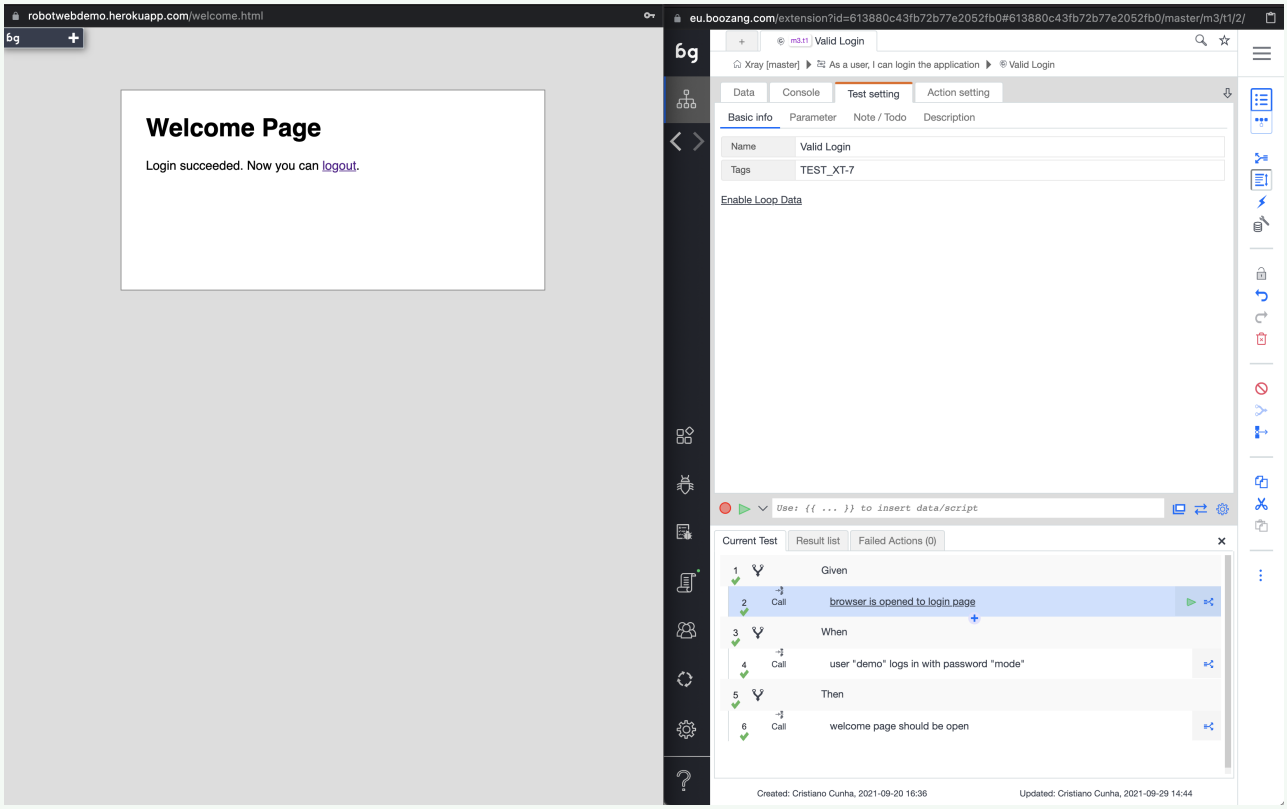
The final result looks like the following screenshot, where all the steps have automated Tests associated and are ready to be executed.



Once you have associated all steps, you can validate that everything is running as expected by clicking on the *Play* button. This option will execute the test and report the results back.

When executing the Tests, Boozang will open up a new browser window where you can follow the Test execution and report back to the Test window.

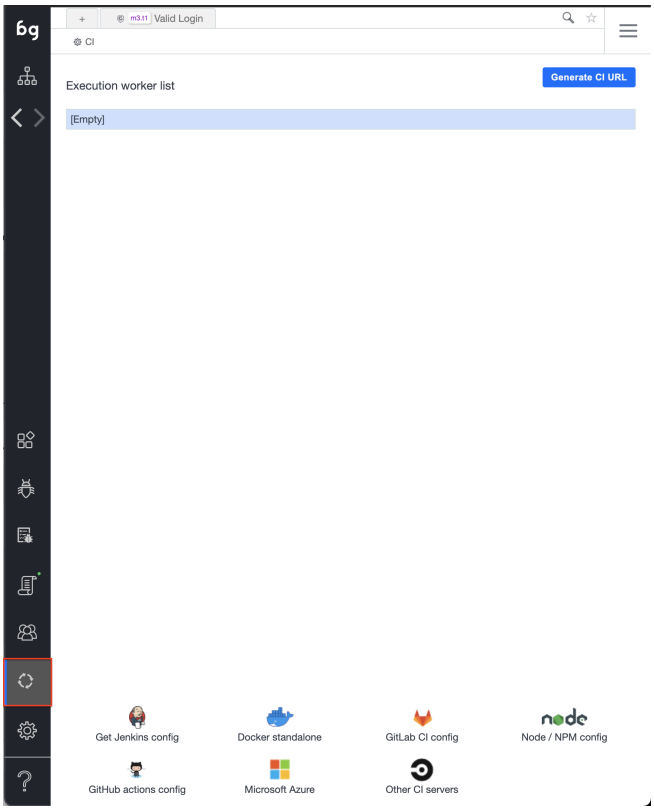




Import results to Xray

Finally, the Tests which have been exported from Xray into Boozang are automated. The results from the executions have to be pushed back into Xray. This can be achieved via Boozang's CI where the ability to generate the necessary scripts for different CI/CD tools that will execute the Boozang Test Runner is available.

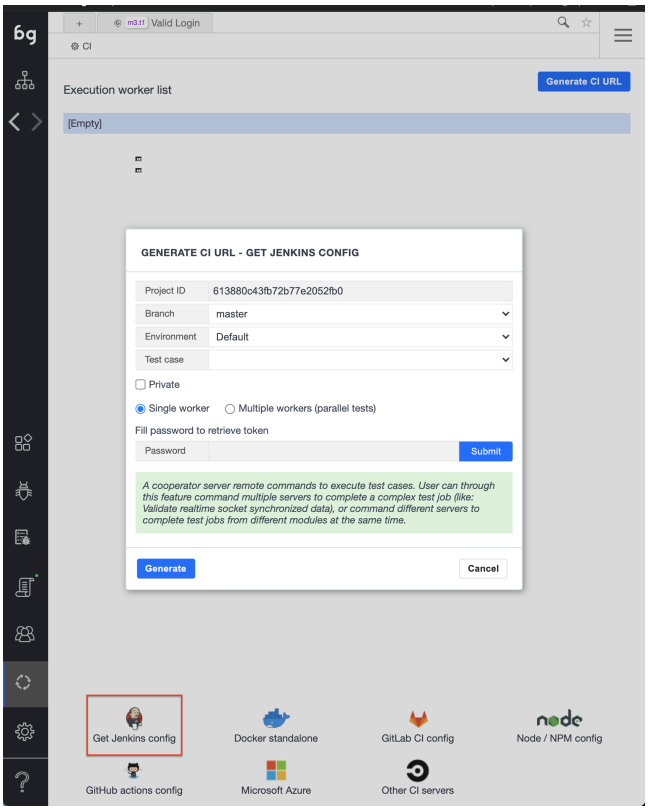
In Boozang, access the CI entry present in the bottom of the left menu.



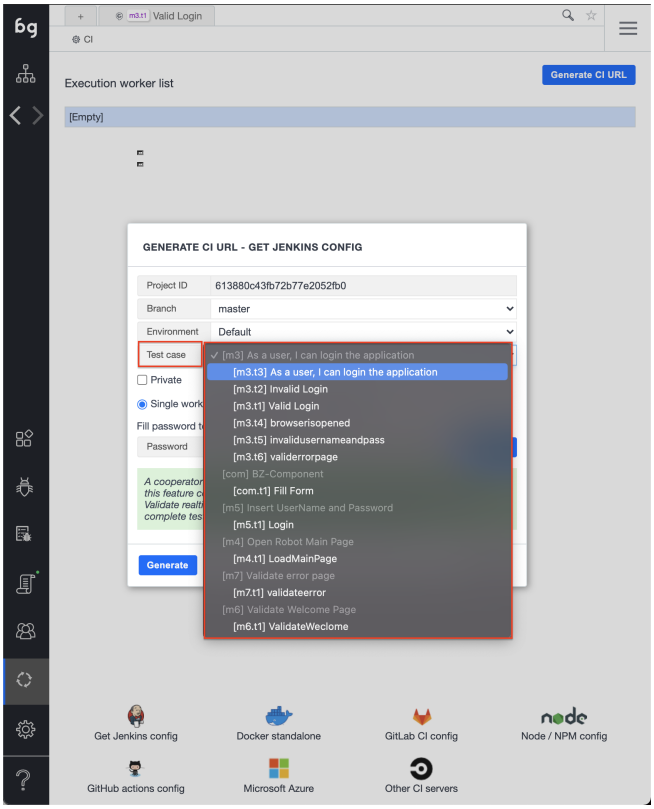
Jenkins

In this example, and to take advantage of the Xray Jenkins Plugin available, the script needed to include in Jenkins was generated by clicking the bottom link "*Get Jenkins config*".

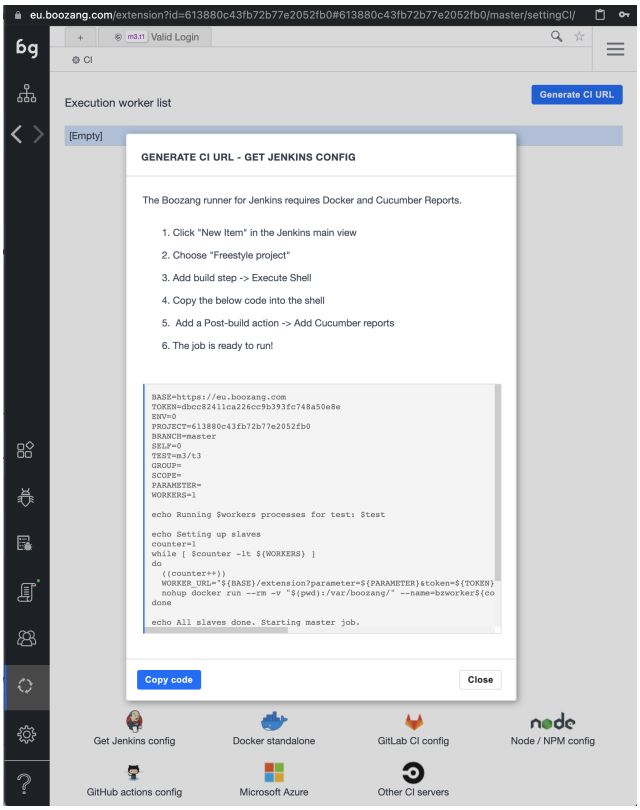
A new pop up will appear requiring details to generate the script.



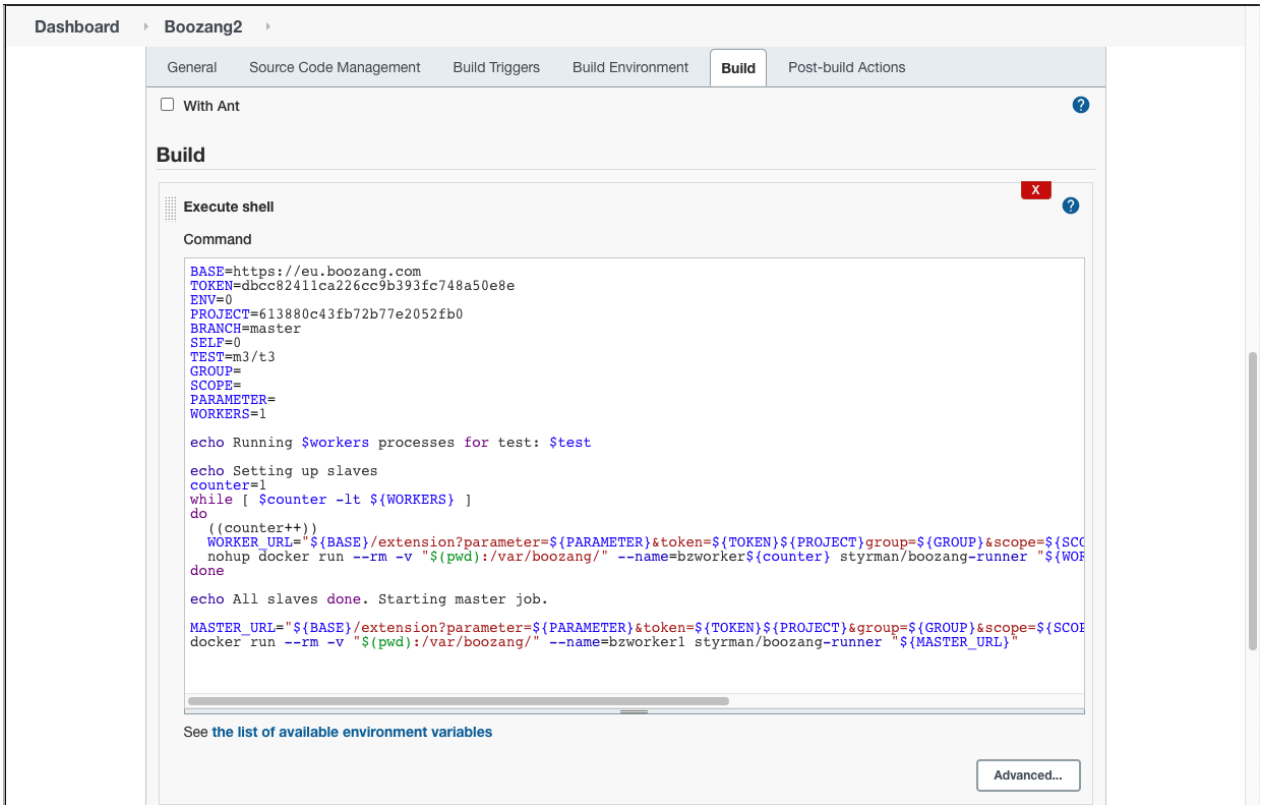
The majority of the options will be the default ones, let's choose the "*Test case*" to be the feature we have automated (from the drop down list).



Fill in the password, click "*Submit*" and click on "*Generate*". This will produce the necessary script that we will be copied and pasted into Jenkins.



In Jenkins, we created a simple Project where an additional step was added to insert the script generated. More specifically by adding a "*Execute shell*" in the build step section and inserting the generated code there.



Tip

If you have defined several Tests the output of the execution will generate one report per Test. In order to upload the results, the results need to merged into a single report. This can be achieved by using a tool that will merge two Cucumber reports into one.

You can find that tool here, the usage is simple:

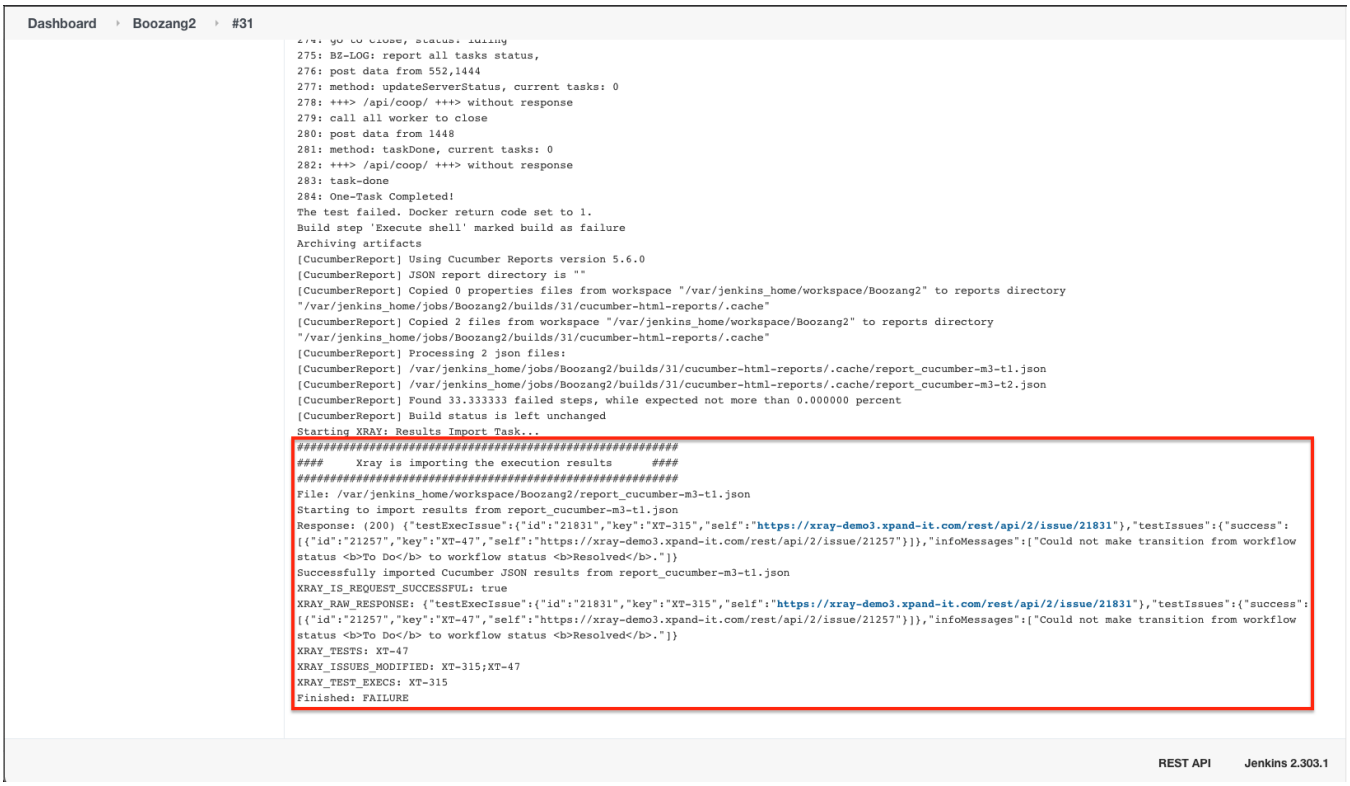
```
cucumber-json-merge-multiworker report_cucumber-m3-t1.json report_cucumber-m3-t2.json
```

After the execution, you can find the report in the default output file "*merged-test-results.json*" or specify the name of the output in the parameter "*-o*".

The last step, before executing the pipeline, is to add the Xray Plugin to import the results back to Xray. First, make sure you have previously installed the plugin and in your pipeline, just add in the Post Build Actions "*Xray: Results Import Task*" and configure properly with:

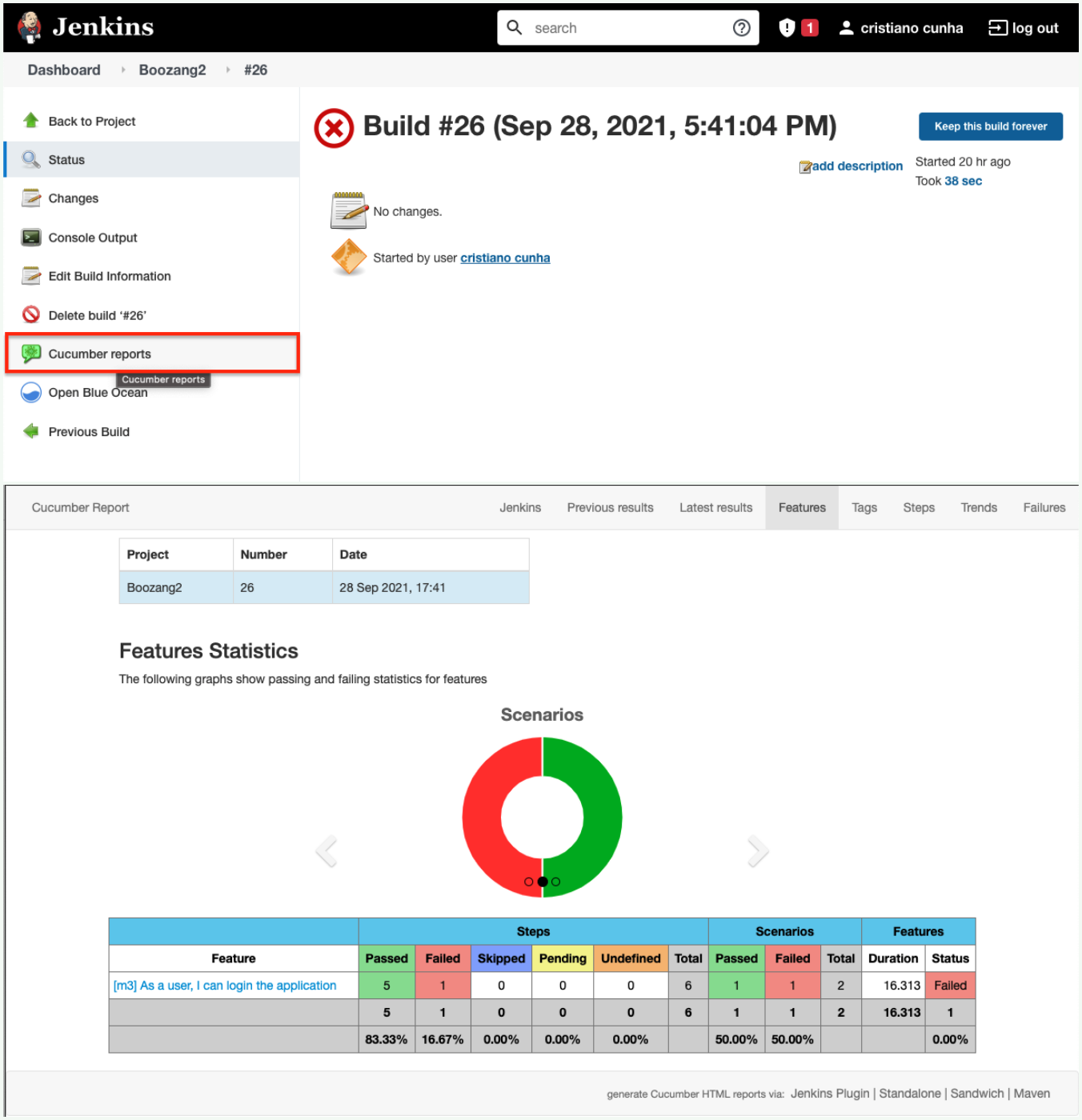
- the Jira/Xray instance that you will use to upload the results
- the Format of the results file you want to import, in our case Cucumber JSON
- The path and name of the file to be uploaded

Once this pipeline is built, it will execute the tests in Boozang and import the results into Xray as you can verify in the "*Console Output*" of the build:

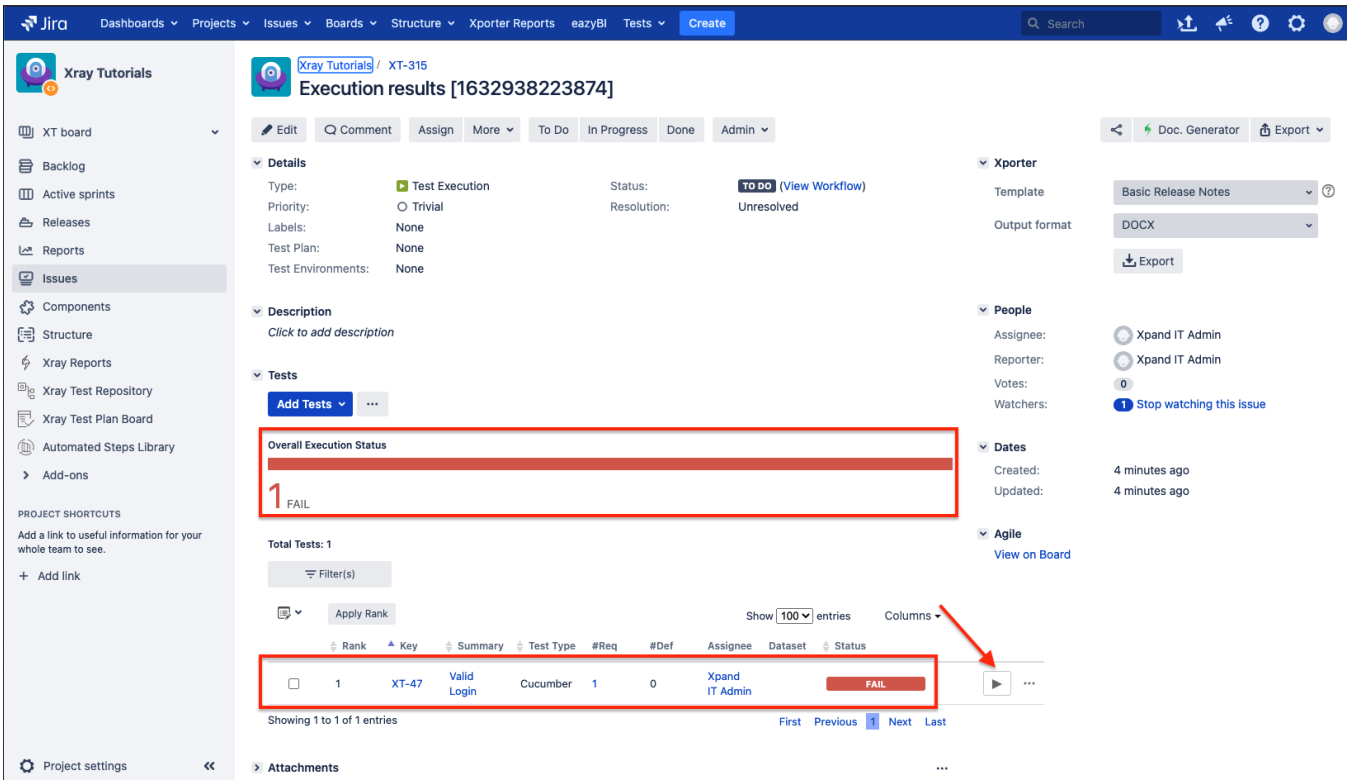


We can see the response from Xray where we can find the Test Execution created for this execution.

If you want to see the results in Jenkins, install the Cucumber Reports Plugin that will automatically process and generate a view in Jenkins for you to go over the results.

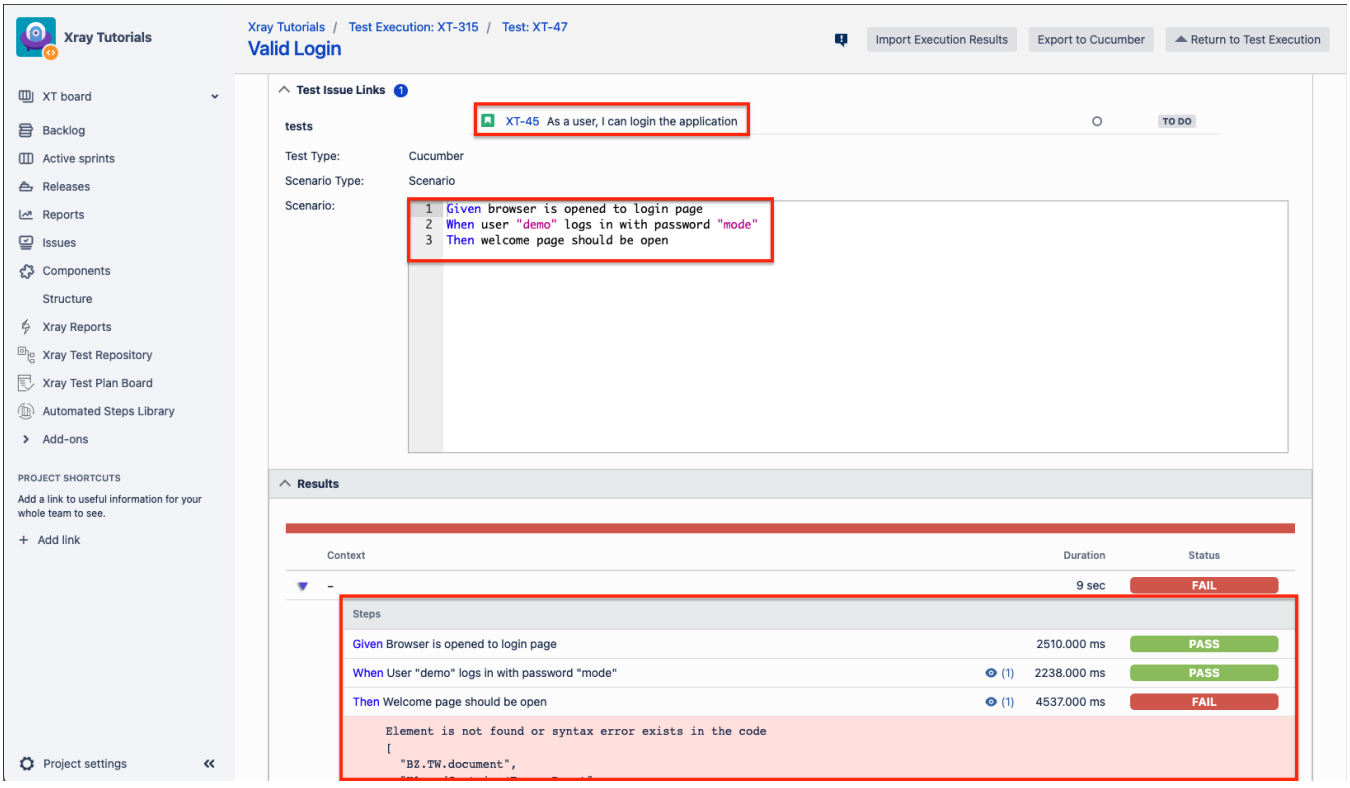


If we verify the Test Execution that was created in Xray, the uploaded results with the total number of Tests and the Overall Execution Status is available - in this case, it's not relevant since there is one Test but if they were several Tests it will be the overall result of all the Tests.



Click on the Details button (below the red arrow) will give users a better understanding of the characteristics of the execution since it navigates to the execution panel of the Test Execution with the following information:

- The requirement that is covered by the Test, in this case, XT-45
- The scenario definition
- The details of the execution of each step



Command line

If Jenkins (or another CI/CD tool) is not part of your workflow you can use the Xray API to import the result back into Xray through the command line. To do so, please follow the next steps:

- Authenticate to obtain the token
- Send the results to Xray using the token

Authenticate

The authentication int he Server/DC version is done using the username and password in every request, so no special step to perform before executing the import request.

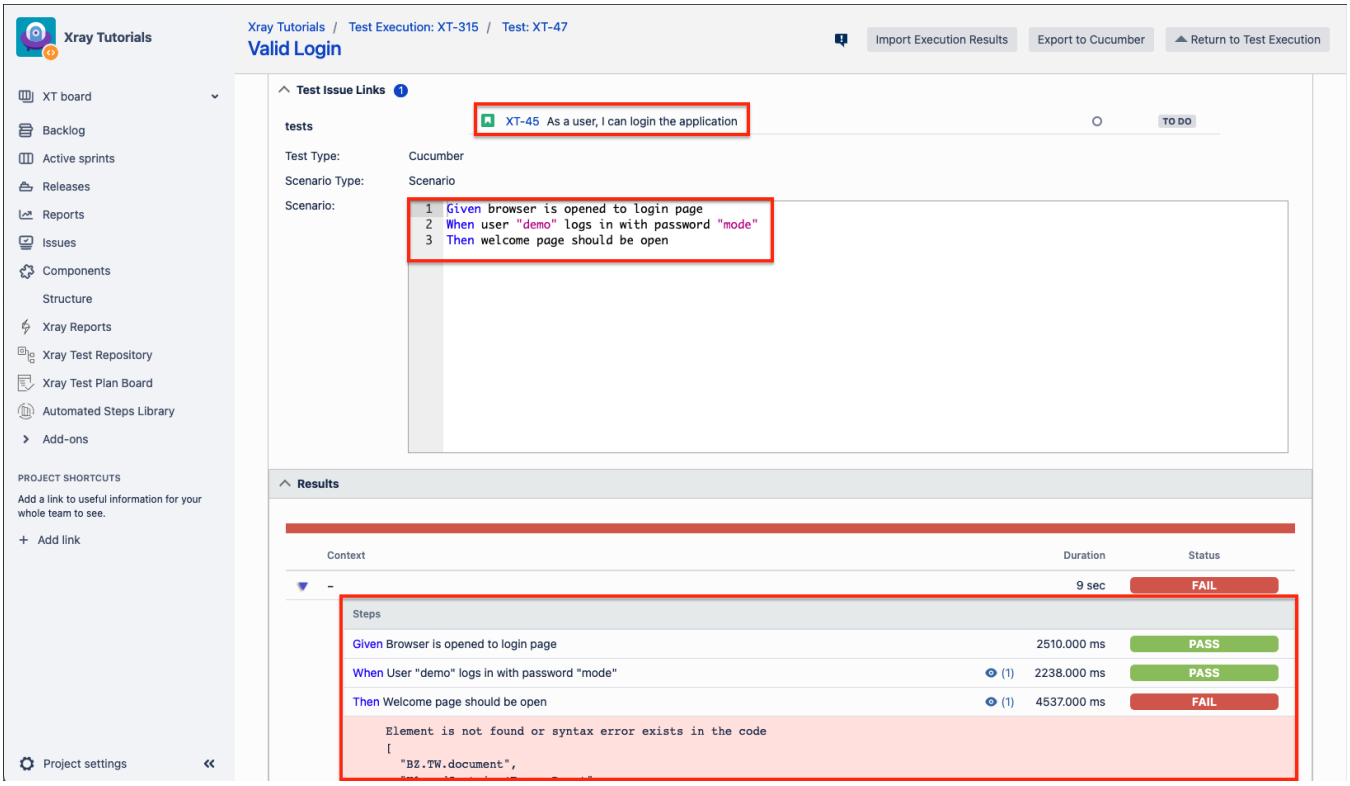
Send results to Xray

Finally to upload the results back into Xray you must use the following request:

```
curl -H "Content-Type: application/json" -X POST -u admin:admin --data @‘xrayResults.json’ ‘https://xray-demo3.xpand-it.com/api/v2/import/execution'
```

When successful, the answer will have information of the Test Execution created.

We can see the detailed results in Xray:



Learn more

- Boozang home page
- Boozang features overview
- Boozang videos