Testing using WebDriverIO and Cucumber in JavaScript

- Overview
- Requirements
- Description
 - Using Jira and Xray as master
 - Using Git or other VCS as master
- FAQ and Recommendations
- References

Overview

In this tutorial, we will create UI tests as Cucumber Scenario(s)/Scenario Outline(s) and use WebDriverIO to implement the tests in JavaScript.

• code is available in GitHub

Requirements

- nodejs
- WebDriverIO

Description

For the purpose of this tutorial, we'll use a dummy website provided by Heroku. In our case, it contains just a few pages to support login features which we will be testing.

To start using WebDriverIO please follow the Get Started documentation.

WebDriverIO provides a client that after being installed will guide you through bootstrapping a *Hello World* test suite into your project, for this tutorial we will use the code generated by this tool for simplicity (with page objects).

The test consists in validating the login feature (with valid and invalid credentials) of the demo site, for that we have created a feature file that will have the description of the test supported by a base page that contains all methods and functionality that is shared across all page objects, a login page, that will extend the base page, that will have all the methods for interacting with the login page and a result page that will have the methods to interact in the page that is loaded after the login operation.

We have followed the documentation and executed the command to install the WebDriverIO test runner:

npm install @wdio/cli

Then we answered a series of questions that will define the code to be generated using:

npx wdio config

The output of the questionnaire will look like this:

```
WDIO Configuration Helper
```

? Where is your automation backend located? On my local machine

- Which framework do you want to use? cucumber
- Do you want to use a compiler? No!
- Where are your feature files located? ./features/**/*.feature
- ? Where are your step definitions located? ./features/step-definitions/steps.js
- P Do you want WebdriverIO to autogenerate some test files? Yes
- Do you want to use page objects (https://martinfowler.com/bliki/PageObject.html)? Yes
- Where are your page objects located? ./features/pageobjects/**/*.js
- Which reporter do you want to use? spec
- Do you want to add a service to your test setup? chromedriver
- What is the base url? http://localhost

This will automatically generate the following files:

./pageobjects/page.js

```
const Page = require('./page');
/**
* sub page containing specific selectors and methods for a specific page
*/
class LoginPage extends Page {
    /**
     * define selectors using getter methods
    */
   get inputUsername () { return $('#username') }
   get inputPassword () { return $('#password') }
   get btnSubmit () { return $('button[type="submit"]') }
    /**
    \ast a method to encapsule automation code to interact with the page
     * e.g. to login using username and password
    */
   async login (username, password) {
       await (await this.inputUsername).setValue(username);
       await (await this.inputPassword).setValue(password);
       await (await this.btnSubmit).click();
    }
    /**
    * overwrite specifc options to adapt it to page object
    */
    open () {
       return super.open('login');
    }
}
module.exports = new LoginPage();
```

./pageobjects/login.page.js

```
/**
* main page object containing all methods, selectors and functionality
* that is shared across all page objects
*/
module.exports = class Page {
    /**
    * Opens a sub page of the page
    * @param path path of the sub page (e.g. /path/to/page.html)
    */
    open (path) {
        return browser.url(`https://the-internet.herokuapp.com/${path}`)
    }
}
```

./pageobjects/secure.page.js

```
const Page = require('./page');
/**
 * sub page containing specific selectors and methods for a specific page
 */
class SecurePage extends Page {
    /**
    * define selectors using getter methods
    */
    get flashAlert () { return $('#flash') }
}
module.exports = new SecurePage();
```

And a feature file where we describe the tests:

```
      login.feature

      Feature: As a user, I can log into the secure area

      Scenario Outline: As a user, I can log into the secure area

      Given I am on the login page

      When I login with <username> and <password>

      Then I should see a flash message saying <message>

      Examples:

      username | password
      | message

      tomsmith | SuperSecretPassword!
      You logged into a secure area!

      foobar
      barfoo
```

With the respective code behind

./step-definitions/steps.js

```
const { Given, When, Then } = require('@cucumber/cucumber');
const LoginPage = require('../pageobjects/login.page');
const SecurePage = require('../pageobjects/secure.page');
const pages = {
   login: LoginPage
}
Given(/^I am on the (\w+) page$/, async (page) => {
   await pages[page].open()
});
When(/^I login with (\w+) and (.+)$/, async (username, password) => {
   await LoginPage.login(username, password)
});
Then(/^I should see a flash message saying (.*)$/, async (message) => {
   await expect(SecurePage.flashAlert).toBeExisting();
   await expect(SecurePage.flashAlert).toHaveTextContaining(message);
});
```

The last two steps to have everything configured is to define that we will use the CucumberJS framework, for that we execute the following command:

npm install wdio-cucumberjs-json-reporter --save-dev

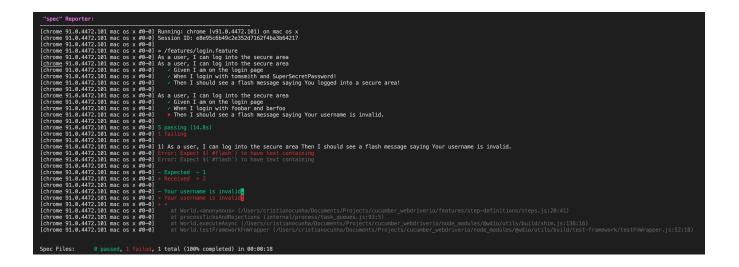
And in the wdio.conf.js we have added, in the reporters area, the following CucumberJS definition:

```
/wdio.conf.js
...
reporters: ['spec',
    ['cucumberjs-json', {
        jsonFolder: '.tmp/json/',
        language: 'en',
        },
        ...
    ],
...
```

Once the code is implemented (and we will make it fail on purpose on one test, to show the failure reports), it can be executed with the following command:

npx wdio run ./wdio.conf.js

The results are immediately available in the terminal



In case you need to interact with the Xray REST API at low-level using scripts (e.g. Bash/shell scripts), this tutorial uses auxiliary files that will handle those interactions.

Example of cloud_auth.json used in this tutorial

- export_features.sh
- import_features.sh
- import_results.sh
- run_all_git_workflow.sh
- run_all_standard_workflow.sh

Now we need to decide which workflow to use: do you want to use Xray/Jira as the master for writing the declarative specification (i.e. the Gherkin based Scenarios), or do you want to manage those outside using some editor and store them in Git, for example?

Learn more

Please see Testing in BDD with Gherkin based frameworks (e.g. Cucumber) for an overview of the possible workflows.

The place that you'll use to edit the Cucumber Scenarios will affect your workflow. There are teams that prefer to edit Cucumber Scenarios in Jira using Xray, while there are others that prefer to edit them by writing the .feature files by hand using some IDE.

Using Jira and Xray as master

This section assumes using Xray as master, i.e. the place that you'll be using to edit the specifications (e.g. the scenarios that are part of .feature files).

The overall flow would be something like this:

- 1. create Scenario/Scenario Outline as a Test in Jira; usually, it would be linked to an existing "requirement"/Story (i.e. created from the respective issue screen)
- 2. implement the code related to Gherkin statements/steps and store it in Git, for example
- 3. generate .feature files based on the specification made in Jira
- 4. checkout the code from Git
- 5. run the tests in the CI
- 6. import the results back to Jira

Usually, you would start by having a Story, or similar (e.g. "requirement"), to describe the behavior of a certain feature and use that to drive your testing.

If you have it, then you can just use the "Create Test" on that issue to create the Scenario/Scenario Outline and have it automatically linked back to the Story/"requirement."

Otherwise, you can create the Test using the standard (issue) Create action from Jira's top menu.

0	Xray Tutorials / 3		•							
🖋 E	dit Q Comment	Assign	More 🗸	To Do	In Progress	Done	Admin 🗸			
∽ De	tails									
Lab	be: prity: pels: quirement Status:	Story Trivial None UNC	OVERED			Stai Res	tus: olution:	TO DO (Unresolv	View Workflow) red	
	scription ck to add descriptior	7								
✓ Tes	t Coverage							Create Test	Create Sub-Test Execution	+ Link ~
No	Tests were found te	sting the requ	irement.							
∽ Att	achments									
					C Drop files	to attach	, or browse.			
> Str	ucture									0

In this case, we'll create a Cucumber Scenario.

We need to create the Test issue first and fill out the Gherkin statements later on in the Test issue screen.

Create Issue		Configure Fields -
Project*	Xray Tutorials (XT)	
Issue Type•	O Test ✓ ⑦	
General Test Detai	ls Test Sets Pre-Conditions Test Plans Link Issues	
Summary*	Test login feature	<u>ا</u>
Description	Style × B I U A × A° × ⊗ × W × ∷≣ ∷≣ ⊙ ×	· + ·
	Visual Text	# ۲ ۲
Reporter*	Start typing to get a list of possible matches.	
Assignee	 Automatic Assign to me 	~
Fix Version/s	None	
Priority	O Trivial ✓ ⑦	
Component/s	None	
Attachment	Drop files to attach, or browse.	
	Create and	other Create Cancel

▼ ⑦

🗘 Configure Fields 👻

15	sue Type*	Test			• ?						
eneral	Test Details	Test Sets	Pre-Conc	litions	Test I	Plans	Link Is	sues			
	Test Type Steps	✓ Manual Cucumber Generic									
		👪 Grid 🗸	* *	¥.4					+	- Add	Step
		1 Action*	E								
		Style ∨	BI	U	<u>A</u> ~ .	<u>^</u> ~ (&∽ [) ~	© ~	+~	*
		Visua	Text								.ati
		Data									
		Enter v	alue								
		Expect	ed Result								

After the Test is created it will impact the coverage of related "requirement," if any.

The coverage and the test results can be tracked on the "requirements" side (e.g. user story). In this case, you may see that coverage changed from being UNCOVERED to NOTRUN (i.e. covered and with at least one test not run).

Create Issue

Project* 🧧 Xray Tutorials (XT)

	ay Tutorials / X emo Logir)										
🖋 Edit	Q Comment	Assign	More 🗸	To Do	In Progress	Done	Admin 🗸						
✓ Details													
Type:	1	Story				Statu	s:	TO DO	(View W	/orkflow)			
Priority:		O Trivial				Reso	ution:	Unres	olved				
Labels:	I	None											
Requiren	nent Status:	N	DTRUN			•							
 Descript Click to a Test Cov 	add description						I	Create Test	Creat	te Sub-Tes	st Execution	+ Li	nk v
TEST CO	OVERAGE FOR THE	FOLLOWING A	NALYSIS SCOPE	E									
Scop	e: Version; Ver	sion: None -	latest execu	tion; Env	ironment: All	Environme	nts +					OT RUN	
÷	Filter(s)												
										Show 10	✓ entries	Colum	ns 🕶
÷	P 🔶 Status	\$	Resolution	4	Кеу	Summa	ary		Test Runs		Test Status		
	О ТО ВО	U	nresolved		XT-226	Test Log	in feature			-	торо		
Showing	1 to 1 of 1 entries									First	Previous 1	Next	Last

Additional tests could be created and eventually linked to the same Story or linked to another one (e.g. logout).

The related statement's code is managed outside of Jira and stored in Git, for example.

In our source code, test code is stored under steps-definitions directory, which itself can contain several other directories or files. In this case, we've only one referring to the login feature:

./step-definitions/steps.js

```
const { Given, When, Then } = require('@cucumber/cucumber');
const LoginPage = require('../pageobjects/login.page');
const SecurePage = require('.../pageobjects/secure.page');
const pages = {
   login: LoginPage
}
Given(/^I am on the (\w+) page$/, async (page) => {
   await pages[page].open()
});
When(/^I login with (\w+) and (.+)$/, async (username, password) => {
    await LoginPage.login(username, password)
});
Then(/^I should see a flash message saying (.*)$/, async (message) => {
   await expect(SecurePage.flashAlert).toBeExisting();
    await expect(SecurePage.flashAlert).toHaveTextContaining(message);
});
After((scenario) => {
   const path = '.tmp/screenshots/Error.png';
    if(scenario.result.status == 6){
       browser.saveScreenshot(path);
       const cucumberJson = require('wdio-cucumberjs-json-reporter').default;
       const data = fs.readFileSync(path);
       if (data) {
            const base64Image = Buffer.from(data, 'binary').toString('base64')
            cucumberJson.attach(base64Image, 'image/png');
        }
    }
});
```

Notice that we have added an After scenario that will be executed after each scenario. After validating that an error occurred it will take a screenshot and attach it to the report using the wdio-cucumberjs-json-reporter library.

You can then export the specification of the test to a Cucumber .feature file via the REST API, or the **Export to Cucumber** UI action from within the Test /Test Execution issue or even based on an existing saved filter. A plugin for your CI tool of choice can be used to ease this task.

So, you can either:

use the UI

Sedit Q Com	nment Assign	More Y To Do In P	rogress Done Admi	n Y
✓ Details		Log work		
Type:	Test	Agile Board	Status:	TO DO (View Workflow)
Priority:	O Trivial	Rank to Top	Resolution:	Unresolved
Labels:	None	Rank to Bottom		
 Description 		Archive		
Click to add descr	ription	Attach files		
		Attach Screenshot		
 Test Details 				
Туре:	Cucumber	Voters		
Scenario Type:	Scenario	Stop watching		
Scenario:	Given I an			
	When I log Then I sho	oreate sub task	d <password> ge saying <message></message></password>	
	Exampl	Convert to sub-task		
		Move	l message	
		Link		ged into a secure area! ername is invalid!
		Clone		d = to as
		Labels		Z Edit Steps
 Pre-Conditions 		Delete		
This test is not as	sociated with Pre-0	Reset TestRunStatus		Create Pre-Condition Associate Pre-Conditions
		Export to Cucumber		
✓ Test Sets		Export Test to XML		
	sociated with Test	Export Test Runs to CSV		
			, ,	Associate Test Sets
	- !			
REST API (more	e into nere)			
example of a	shell script	to export/genera	ate .features fron	n Xray
#!/bin/bas	h			
	-	//192.168.2.10	8	
JIRA_USERN JIRA_PASSW				
KEYS="XT-1				
NDID- AI-I				
rm -f feat	ures.zip			
curl -u \$J	IRA_USERN	AME:\$JIRA_PASS	WORD "\$JIRA	_BASEURL/rest/raven/2.0/export/te
keys=\$KEYS	&fz=true"	-o features.	zip	
ungin o f	ooturoa a	ip -d feature	20	

• use one of the available CI/CD plugins (e.g. see an example of Integration with Jenkins)

We will export the features to a new directory named features/ on the root folder of your project.

After being exported, the created .feature(s) will contain references to the Test issue key, eventually prefixed (e.g. "TEST_") depending on an Xray global setting, and the covered "requirement" issue key, if that's the case. The naming of these files is detailed in Export Cucumber Features backup.

features/1_XT-225.feature	
@REQ_XT-225 Feature: Login feature	
@TEST_XT-226 Scenario: Test Login featur Scenario Outline: A Giv Whe	e s a user, I can log into the secure area en I am on the login page n I login with <username> and <password> n I should see a flash message saying <message></message></password></username>
message area! invalid.	Examples: username password tomsmith SuperSecretPassword! You logged into a secure foobar barfoo Your username is

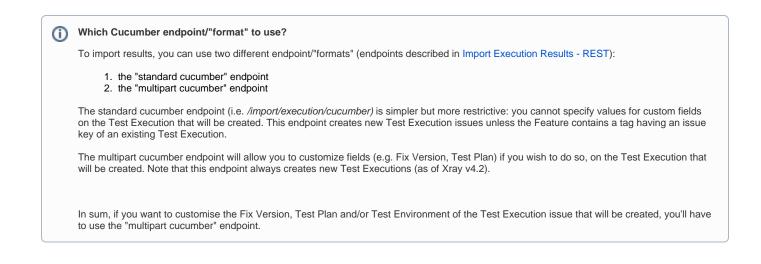
To run the tests and produce Cucumber JSON reports(s), we can either use the same command as before.

npx wdio run ./wdio.conf.js

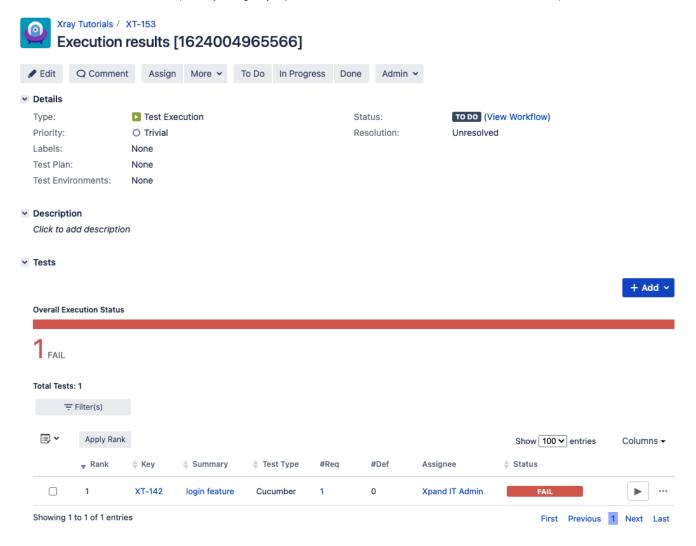
This will produce one results file that will hold the test results.

After running the tests, results can be imported to Xray via the REST API, or the **Import Execution Results** action within the Test Execution, or by using one of the available CI/CD plugins (e.g. see an example of Integration with Jenkins).

import_results.sh	
#!/bin/bash JIRA_BASEURL=https://192.168.0.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin	
curl -H "Content-Type: application/json" -X POST -u \$JIRA_USERNAME:\$JIRA_PASSWORDdata @".tmp/json/login- feature.json" \$JIRA_BASEURL/rest/raven/2.0/import/execution/cucumber	



A new Test Execution will be created (unless you originally exported the Scenarios/Scenario Outlines from a Test Execution).



The tests have failed (on purpose).

The execution screen details of the Test Run will provide overall status information and Gherkin statement-level results, therefore we can use it to analyze the failing test.

Edit Q Cor	nment Assign	More 🛩 To Do Ir		ie Admin 🗸				
Details								 Xporter
Туре:	Test Execut	ion		Status:	TO DO (Vie	ew Workflow)		Template
Priority:	O Trivial		I	Resolution:	Unresolved			
Labels:	None							Output fo
Test Plan: Test Environment	None ts: None							
Description								 People
Click to add desc	cription							Assignee:
Tests								Reporter:
								Votes: Watchers:
							+ Add ~	Hutchers.
Overall Execution S	Status							✓ Dates
1								Created:
FAIL								Updated:
Total Tests: 1								✓ Agile
∓ Filter(s)								View on B
÷ Filter(S)								
Apply Apply	y Rank					Show 100 ✔ entries	Columns -	
							\	
- Ran	nk 🔺 Kev 🔺	Summary 💧 Test 1	Type #Reg	#Def A	ssianee	Status		
_▼ Ran		Summary 📥 Test T			ssignee			
<mark>↓</mark> Ran		Summary 👙 Test 1 ogin feature Cucum			ssignee pand IT Admin	Status FAIL	• ···	
	XT-142 ld						- Evolutiv	on Details
1 Showing 1 to 1 of 1	XT-142 ld					FAIL	- Evolutiv	
1 Showing 1 to 1 of 1 Attachments	XT-142 ld					FAIL	s 1 Execution EXECUTE INIT	
1 Showing 1 to 1 of 1 Attachments	XT-142 ld	ogin feature Cucum				FAIL First Previou	s 1 Executio	
1 Showing 1 to 1 of 1 Attachments Tutorials / Test Exe	XT-142 lc	ogin feature Cucum			pand IT Admin	FAIL First Previou	s 1 Execution	LINE
1 Showing 1 to 1 of 1 Attachments (Tutorials / Test Exe gin feature)	XT-142 lc	ogin feature Cucum			pand IT Admin	FAIL First Previou	s 1 Execution	LINE
1 Showing 1 to 1 of 1 Attachments (Tutorials / Test Exe gin feature Test Details Test Type: Scenario Type:	XT-142 lo I entries	ogin feature Cucum			pand IT Admin	FAIL First Previou	s 1 Execution	LINE
1 Showing 1 to 1 of 1 Attachments Tutorials / Test Exe gin feature Test Details Test Type:	XT-142 ld I entries ecution: XT-153 / Test: XT Cucumber Scenario Outline 1 Given I am on t 2 When I login with	the login page the vusernames and <pass< td=""><td>ber 1</td><td></td><td>pand IT Admin</td><td>FAIL First Previou</td><td>s 1 Execution</td><td>LINE</td></pass<>	ber 1		pand IT Admin	FAIL First Previou	s 1 Execution	LINE
1 Showing 1 to 1 of 1 Attachments (Tutorials / Test Exe gin feature Test Details Test Type: Scenario Type:	XT-142 ld l entries secution: XT-153 / Test: XT Cucumber Scenario Outline 1 Given I am on f 2 When I login wi 3 Then I should s 4	bgin feature Cucum	ber 1		pand IT Admin	FAIL First Previou	s 1 Execution	LINE
1 Showing 1 to 1 of 1 Attachments (Tutorials / Test Exe gin feature Test Details Test Type: Scenario Type:	XT-142 ld l entries secution: XT-153 / Test: XT Cucumber Scenario Outline 1 Given I am on f 2 When I login wi 3 Then I should s 4 5 Examples: 6 Luser	the login page the login page the susername- and spass see a flash message sayi	ber 1	0 X	pand IT Admin	FAIL First Previou	s 1 Execution	LINE
1 Showing 1 to 1 of 1 Attachments (Tutorials / Test Exe gin feature Test Details Test Type: Scenario Type:	XT-142 Id I entries becution: XT-153 / Test: XT Cucumber Scenario Outline 1 Given I agn on f 2 When I login with 3 Then I should s 4 5 Examples: 6 i userr 7 i tomsr	bgin feature Cucum	ber 1	0 X	pand IT Admin	FAIL First Previou	s 1 Execution	LINE
1 Showing 1 to 1 of 1 Attachments (Tutorials / Test Exe gin feature Test Details Test Type: Scenario Type:	XT-142 ld I entries becution: XT-153 / Test: XT Cucumber Scenario Outline 1 Given I agn on f 2 When I login wi 3 Then I should s 4 5 Examples: 6 user 7 toms	bgin feature Cucum 	sword> ing <message> rd! You logged i</message>	0 X	pand IT Admin	FAIL First Previou	s 1 Execution	LINE
1 Showing 1 to 1 of 1 Attachments (Tutorials / Test Exe gin feature Test Details Test Type: Scenario Type:	XT-142 ld I entries becution: XT-153 / Test: XT Cucumber Scenario Outline 1 Given I agn on f 2 When I login wi 3 Then I should s 4 5 Examples: 6 user 7 toms	bgin feature Cucum 	sword> ing <message> rd! You logged i</message>	0 X	pand IT Admin	FAIL First Previou	s 1 Execution	LINE
1 Showing 1 to 1 of 1 Attachments (Tutorials / Test Exe gin feature Test Details Test Type: Scenario Type:	XT-142 ld I entries becution: XT-153 / Test: XT Cucumber Scenario Outline 1 Given I agn on f 2 When I login wi 3 Then I should s 4 5 Examples: 6 user 7 toms	bgin feature Cucum 	sword> ing <message> rd! You logged i</message>	0 X	pand IT Admin	FAIL First Previou	s 1 Execution	LINE
1 Showing 1 to 1 of 1 Attachments (Tutorials / Test Exe gin feature Test Details Test Type: Scenario Type:	XT-142 ld I entries becution: XT-153 / Test: XT Cucumber Scenario Outline 1 Given I agn on f 2 When I login wi 3 Then I should s 4 5 Examples: 6 user 7 toms	bgin feature Cucum 	sword> ing <message> rd! You logged i</message>	0 X	pand IT Admin	FAIL First Previou	s 1 Execution	LINE
1 Showing 1 to 1 of 1 Attachments (Tutorials / Test Exe gin feature Test Details Test Type: Scenario Type:	XT-142 ld I entries becution: XT-153 / Test: XT Cucumber Scenario Outline 1 Given I agn on f 2 When I login wi 3 Then I should s 4 5 Examples: 6 user 7 toms	bgin feature Cucum 	sword> ing <message> rd! You logged i</message>	0 X	pand IT Admin	FAIL First Previou	s 1 Execution	LINE
1 Showing 1 to 1 of 1 Attachments 'Tutorials / Test Exergin feature Test Details Test Type: Scenario Type: Scenario:	XT-142 ld I entries becution: XT-153 / Test: XT Cucumber Scenario Outline 1 Given I agn on f 2 When I login wi 3 Then I should s 4 5 Examples: 6 user 7 toms	bgin feature Cucum 	sword> ing <message> rd! You logged i</message>	0 X	pand IT Admin	FAIL First Previou	s 1 Execution	LINE
1 Showing 1 to 1 of 1 Attachments 'Tutorials / Test Exergin feature Test Details Test Type: Scenario Type: Scenario:	XT-142 ld I entries becution: XT-153 / Test: XT Cucumber Scenario Outline 1 Given I agn on f 2 When I login wi 3 Then I should s 4 5 Examples: 6 user 7 toms	bgin feature Cucum 	sword> ing <message> rd! You logged i</message>	0 X	pand IT Admin	FAIL First Previou	s 1 Execution	LINE

A given example can be expanded to see all Gherkin statements and, if available, it is possible to see also the attached stack trace.

Exam	ples				^
	<username></username>	<password></password>	<message></message>	Duration	Status
	tomsmith	SuperSecretPassword!	You logged into a secure area!	3861.000 ms	PASS
	foobar	barfoo	Your username is invalid.	10952.000 ms	FAIL
	Steps				
	Before			2.000 ms	PASS
	Given I am on the login page			166.000 ms	PASS
	When I login with foobar and barfoo			484.000 ms	PASS
	Then I should see a flash message saying Your u	username is invalid.		10036.000 ms	FAIL
	at processTicksAndRejections at World.executeAsync (/User	/cristianocunha/Documents/Proj (internal/process/task_queues s/cristianocunha/Documents/Pro	ects/cucumber_webdriverio/features/step-definition .js:93:5) jects/cucumber_webdriverio/node_modules/@wdio/util uments/Projects/cucumber_webdriverio/node_modules/	s/build/shim.js:136:16)	/testFnWrapper.js:52:18)
	After				PASS
	After			1.000 ms	PASS

Note: in this case, the bug was on the Scenario Outline example which was expecting an invalid message.

Results are reflected on the covered item (e.g. Story). On its issue screen, coverage now shows that the item is NOK based on the latest testing results, this can also be tracked within the Test Coverage panel bellow.

Edit Q C	Comment Assign	More Y To Do In P	rogress Done Adm	in 🗸			
etails							
ype:	Story			Status:	TO DO (View Workflow)		
riority:	O Trivial			Resolution:	Unresolved		
abels:	None						
equirement S	status:						
escription							
lick to add de	escription						
lick to add de	escription						
					Create Test Cr	reste Sub-Tast Evecution	+ Lin
					Create Test Cr	reate Sub-Test Execution	+ Lini
est Coverage		ANALYSIS SCOPE			Create Test Cr	reate Sub-Test Execution	+ Lin
est Coverage	GE FOR THE FOLLOWING		nent: All Environments +		Create Test Cr		+ Lini
est Coverage	GE FOR THE FOLLOWING	ANALYSIS SCOPE e - latest execution; Environn	nent: All Environments +		Create Test Cr		
	BE FOR THE FOLLOWING		nent: All Environments +		Create Test Cr		
est Coverage TEST COVERAG Scope: Ver ≂ Filter(BE FOR THE FOLLOWING		nent: All Environments 👻		Create Test Cr		
est Coverage TEST COVERAG Scope: Ver ≂ Filter(BE FOR THE FOLLOWING		nent: All Environments 👻		Create Test Cr		NOK
est Coverage TEST COVERAG Scope: Ver	BE FOR THE FOLLOWING		nent: All Environments 👻	Summary	Create Test Cr	-	+ Link NOK Columns

Using Git or other VCS as master

You can edit your .feature files using your IDE outside of Jira (eventually storing them in your VCS using Git, for example) alongside the remaining test code.

In any case, you'll need to synchronize your .feature files to Jira so that you can have visibility of them and report results against them.

The overall flow would be something like this:

- 1. look at the existing "requirement"/Story issue keys to guide your testing; keep their issue keys
- 2. specify Cucumber/Gherkin .feature files in your IDE and store it in Git, for example
- 3. implement the code related to Gherkin statements/steps and store it in Git, for example
- 4. import/synchronise the .feature files to Xray to provision or update corresponding Test entities
- 5. export/generate .feature files from Jira, so that they contain references to Tests and requirements in Jira
- 6. checkout the WebDriverIO related code from Git
- 7. run the tests in the CI
- 8. import the results back to Jira

Usually, you would start by having a Story, or similar (e.g. "requirement"), to describe the behaviour of a certain feature and use that to drive your testing.

	emo Login)							
🖋 Edit	Q Comment	Assign	More 🗸	To Do	In Progress	Done	Admin 🗸			
✓ Details										
Type: Priority: Labels: Requirem	(Story Trivial None UNC	OVERED			Stat Res	us: olution:	TO DO (Unresolv	View Workflow) red	
 Descripting Click to a 	ion add description									
 Test Cov 	erage									
No Tests	were found test	ing the requ	irement.					Create Test	Create Sub-Test Execution	+ Link ~
✓ Attachm	ents									
					Orop files	to attach	, or browse.			
> Structure	e									0

Having those to guide testing, we could then move to our code to describe and implement the Cucumber test scenarios.

Test code is stored inside the step-definitions directory. We also have other directories present, to hold for instance the page object definitions in the page objects directory.

In this case, we've organized them as follows:

• step-definitions/steps.js: step implementation files, in JavaScript.

```
    step-definitions/steps.js
```

```
const { Given, When, Then } = require('@cucumber/cucumber');
const LoginPage = require('../pageobjects/login.page');
const SecurePage = require('../pageobjects/secure.page');
const pages = {
   login: LoginPage
}
Given(/^I am on the (\w+) page$/, async (page) => {
  await pages[page].open()
});
When(/^I login with (\w+) and (.+)$/, async (username, password) => {
  await LoginPage.login(username, password)
});
Then(/^I should see a flash message saying (.*)$/, async (message) => {
    await expect(SecurePage.flashAlert).toBeExisting();
    await expect(SecurePage.flashAlert).toHaveTextContaining(message);
});
```

• pageobjects: abstraction of different pages, somehow based on the page-objects model

o pageobjects/page.js

```
/**
* main page object containing all methods, selectors and functionality
* that is shared across all page objects
*/
module.exports = class Page {
    /**
    * Opens a sub page of the page
    * @param path path of the sub page (e.g. /path/to/page.html)
    */
    open (path) {
        return browser.url(`https://the-internet.herokuapp.com/${path}`)
    }
}
```

```
    pageobjects/login-page.js
```

```
const Page = require('./page');
/**
 * sub page containing specific selectors and methods for a specific page
*/
class LoginPage extends Page {
   /**
    * define selectors using getter methods
    */
    get inputUsername () { return $('#username') }
    get inputPassword () { return $('#password') }
    get btnSubmit () { return $('button[type="submit"]') }
    /**
    * a method to encapsule automation code to interact with the page
     \ast e.g. to login using username and password
     */
    async login (username, password) {
       await (await this.inputUsername).setValue(username);
       await (await this.inputPassword).setValue(password);
       await (await this.btnSubmit).click();
    }
    /**
    * overwrite specifc options to adapt it to page object
    */
    open () {
       return super.open('login');
    }
}
module.exports = new LoginPage();
```

o pageobjects/secure-page.js

```
const Page = require('./page');
/**
 * sub page containing specific selectors and methods for a specific page
 */
class SecurePage extends Page {
    /**
    * define selectors using getter methods
    */
    get flashAlert () { return $('#flash') }
}
module.exports = new SecurePage();
```

• features/login.feature: Cucumber .feature files, containing the tests as Gherkin Scenario(s)/Scenario Outline(s). Please note that each "Feature: <...>" section should be tagged with the issue key of the corresponding "requirement"/story in Jira. You may need to add a prefix (e.g. "REQ_") before the issue key, depending on an Xray global setting.

0	○ features/login.feature										
	@REQ_XT-225										
	Feature: Login feature										
	Scenario: Test Login feature										
	Scenario Outline: As a user, I can log into the secure area										
	Given I am on the login page										
	When I login with <username> and <password></password></username>										
	Then I should see a flash message saying <message></message>										
	Examples:										
	username password										
	message										
	tomsmith SuperSecretPassword! You loo	ged									
	into a secure area!										
	foobar barfoo Your us	sername									
	is invalid.										

Before running the tests in the CI environment, you need to import your .feature files to Xray/Jira; you can invoke the REST API directly or use one of the available plugins/tutorials for CI tools.

example of a shell script to import/	ynchronize Scenario(s)/Background(s) from .features to Jira and Xray	

#!/bin/bash

```
JIRA_BASEURL=https://192.168.0.168
JIRA_USERNAME=admin
JIRA_PASSWORD=admin
PROJECT=XT
FILE=features.zip
```

```
zip -r $FILE features/ -i \*.feature
curl -H "Content-Type: multipart/form-data" -u $JIRA_USERNAME:$JIRA_PASSWORD -F "file=@$FILE" "$JIRA_BASEURL
/rest/raven/1.0/import/feature?projectKey=$PROJECT"
```

Please note

Each Scenario of each .feature will be created as a Test issue that contains unique identifiers, so that if you import once again then Xray can update the existent Test and don't create any duplicated tests.

Afterward, you can export those features out of Jira based on some criteria, so they are properly tagged with corresponding issue keys; this is important because results need to contain these references.

You can then export the specification of the test to a Cucumber .feature file via the REST API, or the **Export to Cucumber** UI action from within the Test /Test Execution issue or even based on an existing saved filter. A plugin for your CI tool of choice can be used to ease this task.

So, you can either:

use the UI

v Details V Details Log work Agile Board Agile Board Agile Board Agile Board Priority: O Thirlal Rank to Bottom Archive Click to add description Attach Screenshot Type: Cucumber Scenario Type: Scenario: Given I at When I tog Create sub-task Pre-Conditions This test is not associated with Pre- Labels: Pre-Conditions This test is not associated with Pre- Excorp V Test Sets Export Test Runs to CSV Resolution: V Test Sets Export Test Runs to CSV Resolution: V Test Sets This test is not associated with Pre- Export Test Runs to CSV Resolution: Agle Set Test TAPI (more info here) • export Test Runs to CSV Status:: I Trans.passEvr.l=https://192.168.2.168 JIRA_DASEVRL=https://192.168.2.168 JIRA_DASEVRD=admin JIRA_PASSWORD=admin KEYS = "XT-142" rm -f features.zip	Sedit Q Commer	nt Assign		Progress Done Admin -
Priority: O Trivial Rank to Top Rank to Top Rank to Top Rank to Top Rank to Top Rank to Bottom Vers Scenario	 Details 		Log work	
Labels: None Rank to Bop Rank to Bop Rank to Bottom Click to add description Click to add description Click to add description Attach files Attach Screenshot Type: Cucumber Scenario Type: Scenario Scenario: Cliven I or Scenario: Cliven I or Scenario: Cliven I or Convert to sub-task Excorr Unix Convert to sub-task Convert to			Agile Board	
Rank to Bottom Archive Archive Archive Attach files Attach files Attach files Attach files Attach Screenshot Type: Scenario Type: Scenario Scenario: Given I at When I to Create sub-task Convert to sub-task convert to sub-task Example Move Link Clock to add description Move Imassage Example Move Labels Password-1 Pre-Conditions Delete This test is not associated with Pre- Reset TestRunStatus Export To Cucumber Export Test to XML Pre-Conditions The test is not associated with Pre- Rest Test Sets Export Test Runs to CSV This test is not associated with Pre- Export Test Runs to CSV Attach Scenario Labels Export Test Runs to CSV Associate Test Sets This test is not associated with Test Export Test Runs to CSV Attach Scenario Labels Link Labels Export Test Runs to CSV As			Rank to Top	Resolution: Unresolved
Click to add description Attach files Atta	Labels:	None	Rank to Bottom	
Y Test Details Attach files Type: Cucumber Scenario Sign Vatching Scenario: Given I or Then I sk Convert to sub-task I cpassword> e saying <message> Convert to sub-task Excorp Move I message Link Clone Password! Link Clone Password! Delete Delete Password! This test is not associated with Pre- Export Test to XML Reset TestRunStatus Export Test to XML Export Test to XML Export Test to XML Export Test Runs to CSV Associate Test Sets Export Test Runs to CSV EST API (more info here) example of a shell script to export/generate .features from Xray #1/bin/bash JIRA_BASEURL=https://192.168.2.168 JIRA_PASSWORD=admin JIRA_PASSWORD=admin JIRA_PASSWORD=admin KEYS = "XT - 142"</message>	 Description 		Archive	-
<pre> test Details</pre>	Click to add description	on	Attach files	
Type: Cucumber Scenario Type: Scenario Scenario: Given I a Watchers Create sub-task Convert to sub-task i -possword> Example Convert to sub-task Convert to sub-task i -possword> Example Move Link Clone Labels Delete Delete Export to Cucumber Export to Cucumber Export to Cucumber Export to Cucumber Export Test to XML Export Test to XML Export Test to XML Export Test to XML Export Test Runs to CSV Associate Test Sets Export Test Runs to CSV This test is not associated with Test Export Test to XML Export Test to XML Export Test Runs to CSV Associate Test Sets Export Test Runs to CSV Example of a shell script to export/generate .features from Xray # ! /bin/bash JIRA_BASEURL=https: //192.168.2.168 JIRA_DSERNAME=admin JIRA_PASSWORD=admin KEYS="XT-142" Scenario				
Scenario Type: Scenario Scenario: Given I or When I leg Then I sh Example of a shell script to export/generate .features from Xray # 1/bin/bash JIRA_BASEURL=https://192.168.2.168 JIRA_DSSWORD=admin JIRA_PASSWORD=admin KEYS="XT-142"				-
Scenario: Given I or Scenario: Given I or Then I st Example I watchers Create sub-task Example I watchers Create sub-task Example I watchers Create sub-task Example I watchers Create sub-task Example I watchers Password- Pas	Туре:	Cucumber		
Scenario: Convert to sub-task Example Move Link Clone Labels Pessword! message Password! message Pass	Scenario Type:	Scenario		
Then I sho Convert to sub-task pe saying <message> Convert to sub-task Convert to sub-task pe saying <message> Ink Unk Your Usgged into a secure areal Unk Convert to sub-task Your Usgged into a secure areal Vere-Conditions Delete Pre-Condition This test is not associated with Pre- Reset TestRunStatus Create Pre-Condition Export Test to XML Export Test to XML Export Test Runs to CSV This test is not associated with Test Export Test Runs to CSV Associate Test Sets This test is not associated with Test Export Test Runs to CSV Associate Test Sets REST API (more info here) Example of a shell script to export/generate .features from Xray #1/bin/bash JIRA_BASEURL=https://192.168.2.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin JIRA_PASSWORD=admin KEYS="XT-142"</message></message>	Scenario:		1	
Convert to sub-task Exampl Move Link Clone Labels Delete This test is not associated with Pre- Reset TestRunStatus Export to Cucumber Export Test to XML Export Test Runs to CSV Associate Test Sets RESET API (more info here) example of a shell script to export/generate .features from Xray # 1/bin/bash JIRA_BASEURL=https://192.168.2.168 JIRA_DASSWORD=admin KEYS="XT-142"				
Move Imessage Imessage Imessage Imessage Link Clone Labels Imessage You logged into a secure area! Imessage Yere-Conditions Delete Reset TestRunStatus Create Pre-Condition Associate Pre-Conditions This test is not associated with Pre- Export to Cucumber Export Test to XML Export Test to XML Export Test to XML Export Test to XML Export Test to XML Export Test to XML Export Test to XML Export Test Runs to CSV Associate Test Sets RESET API (more info here) example of a shell script to export/generate .features from Xray # ! /bin/bash JIRA_BASEURL=https://192.168.2.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin JIRA_PASSWORD=admin KEYS="XT-142"			Convert to sub-task	
I Link Cone Labels Cone Labels Delete Pre-Conditions This test is not associated with Pre-Cest Sets Export Test to XML Export Test to XML Export Test to XML Export Test Runs to CSV Associate Test Sets REST API (more info here) example of a shell script to export/generate .features from Xray #!/bin/bash JIRA_BASEURL=https://192.168.2.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin KEYS="XT-142"		E xamp i		l message l
Cione Labels Delete This test is not associated with Pre- This test is not associated with Pre- Export to Cucumber Export Test to XML Export Test to XML Export Test Runs to CSV Test Sets This test is not associated with Test Export Test Runs to CSV Test Sets EXECT API (more info here) EXECT API (more info here) EXECT API (more info here) TASSOCIATE TEST Sets TASSOCIATE TEST SETS SETS SETS TASSOCIATE TEST SETS SETS SETS SETS SETS SETS S			Link	
 Pre-Conditions This test is not associated with Pre- Test Sets This test is not associated with Test Export To Cucumber Export Test to XML Export Test to XML Export Test Runs to CSV Associate Test Sets This test is not associated with Test Export Test Runs to CSV Associate Test Sets This test is not associated with Test Export Test Runs to CSV Associate Test Sets This test is not associated with Test Export Test Runs to CSV Associate Test Sets This test is not associated with Test Export Test Runs to CSV Associate Test Sets Test API (more info here) Use API (more info here) JIRA_BASEURL=https://192.168.2.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin KEYS="XT-142"			Clone	
<pre>> Pre-Conditions This test is not associated with Pre- Paper to Cucumber Export to Cucumber Export Test to XML Export Test to XML Export Test Runs to CSV Associate Test Sets REST API (more info here) example of a shell script to export/generate .features from Xray #!/bin/bash JIRA_BASEURL=https://192.168.2.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin KEYS="XT-142"</pre>			Labels	Z Edit Steps
★ Test Sets This test is not associated with Test. Export Test to XML Export Test XML Export Test Runs to CSV Associate Pre-Condition Associate Pre-Conditions Export Test Runs to CSV Associate Test Sets EEST API (more info here) Example of a shell script to export/generate .features from Xray #!/bin/bash JIRA_BASEURL=https://192.168.2.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin KEYS="XT-142"	 Pre-Conditions 		Delete	
► Test Sets This test is not associated with Test Export Test to XML Export Test Runs to CSV EEST API (more info here) example of a shell script to export/generate .features from Xray #!/bin/bash JIRA_BASEURL=https://192.168.2.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin KEYS="XT-142"	This test is not associa	ated with Pre-0	Reset TestRunStatus	Create Pre-Condition Associate Pre-Conditions
This test is not associated with Test Export Test Runs to CSV Associate Test Sets REST API (more info here) example of a shell script to export/generate .features from Xray #!/bin/bash JIRA_BASEURL=https://192.168.2.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin KEYS="XT-142"			Export to Cucumber	
This test is not associated with Test Export Test Runs to CSV Associate Test Sets REST API (more info here) example of a shell script to export/generate .features from Xray #!/bin/bash JIRA_BASEURL=https://192.168.2.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin KEYS="XT-142"	 Test Sets 		Export Test to XML	
EEST API (more info here) example of a shell script to export/generate .features from Xray #!/bin/bash JIRA_BASEURL=https://192.168.2.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin KEYS="XT-142"		ated with Test	Export Test Runs to CSV	
example of a shell script to export/generate .features from Xray #!/bin/bash JIRA_BASEURL=https://192.168.2.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin KEYS="XT-142"				Associate Test Sets
example of a shell script to export/generate .features from Xray #!/bin/bash JIRA_BASEURL=https://192.168.2.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin KEYS="XT-142"		ofo here)		
<pre>#!/bin/bash JIRA_BASEURL=https://192.168.2.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin KEYS="XT-142"</pre>	'EST API (more in			
JIRA_BASEURL=https://192.168.2.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin KEYS="XT-142"	,			
JIRA_USERNAME=admin JIRA_PASSWORD=admin KEYS="XT-142"	, ,	hell script	t to export/genera	ate .features from Xray
JIRA_USERNAME=admin JIRA_PASSWORD=admin KEYS="XT-142"	example of a sh	hell script	t to export/genera	ate .features from Xray
JIRA_PASSWORD=admin KEYS="XT-142"	example of a sh #!/bin/bash			
KEYS="XT-142"	example of a sh #!/bin/bash JIRA_BASEURL	-=https:	//192.168.2.16	
	example of a sh #!/bin/bash JIRA_BASEURL JIRA_USERNAM	-=https: IE=admin	//192.168.2.16	
rm -f features zip	example of a sh #!/bin/bash JIRA_BASEURL JIRA_USERNAM JIRA_PASSWOR	=https: E=admin 2D=admin	//192.168.2.16	
1	example of a sh #!/bin/bash JIRA_BASEURL JIRA_USERNAM JIRA_PASSWOR	=https: E=admin 2D=admin	//192.168.2.16	
	example of a sh #!/bin/bash JIRA_BASEURL JIRA_USERNAM JIRA_PASSWOR KEYS="XT-142 rm -f featur	=https: IE=admin 2=admin 2= res.zip	//192.168.2.16	68

• use one of the available CI/CD plugins (e.g. see an example of Integration with Jenkins)

For CI only purposes, we will export the features to a new temporary directory named features/ on the root folder of your project. Please note that while implementing the tests, feature files should be edited inside their respective folder.

After being exported, the created .feature(s) will contain references to the Test issue keys, eventually prefixed (e.g. "TEST_") depending on an Xray global setting, and the covered "requirement" issue key, if that's the case. The naming of these files is detailed in Export Cucumber Features backup.

```
      features/1_COM_19.feature

      @REQ_XT-225

      Feature: Login feature

      @TEST_XT-226

      Scenario Outline: As a user, I can log into the secure area

      Given I am on the login page

      When I login with <username> and <password>

      Then I should see a flash message saying <message>

      Examples:

      username | password
      | message

      tomsmith | SuperSecretPassword!
      You logged into a secure area!

      foobar
      barfoo
```

To run the tests and produce Cucumber JSON reports(s), we will use the following command:

npx wdio run ./wdio.conf.js

This will produce one Cucumber JSON report in .tmp/json directory per each .feature file.

After running the tests, results can be imported to Xray via the REST API, or the **Import Execution Results** action within the Test Execution, or by using one of the available CI/CD plugins (e.g. see an example of Integration with Jenkins).

Example of shell script to import results (e.g. import_results_cloud.sh)

#!/bin/bash

JIRA_BASEURL=https://192.168.0.168 JIRA_USERNAME=admin JIRA_PASSWORD=admin

curl -H "Content-Type: application/json" -X POST -u \$JIRA_USERNAME:\$JIRA_PASSWORD --data @".tmp/json/login-feature.json" \$JIRA_BASEURL/rest/raven/2.0/import/execution/cucumber

Which Cucumber endpoint/"format" to use?

To import results, you can use two different endpoints/"formats" (endpoints described in Import Execution Results - REST):

1. the "standard cucumber" endpoint

2. the "multipart cucumber" endpoint

The standard cucumber endpoint (i.e. */import/execution/cucumber*) is simpler but more restrictive: you cannot specify values for custom fields on the Test Execution that will be created. This endpoint creates new Test Execution issues unless the Feature contains a tag having an issue key of an existing Test Execution.

The multipart cucumber endpoint will allow you to customize fields (e.g. Fix Version, Test Plan) if you wish to do so, on the Test Execution that will be created. Note that this endpoint always creates new Test Executions (as of Xray v4.2).

In sum, if you want to customize the Fix Version, Test Plan and/or Test Environment of the Test Execution issue that will be created, you'll have to use the "multipart cucumber" endpoint.

	ay Tutorials / xecution		1624004	49655	66]							
🖋 Edit	Q Commen	t Assign	More 🗸	To Do	In Progress	Done	Admin 🗸					
Details												
Type:		🔁 Test Exe	cution			Stat	us:	TO DO (V	'iew Workflow)			
Priority:							olution:	Unresolve	,			
Labels:		None										
Test Plan	1:	None										
Test Envi	ironments:	None										
Descript Click to a	ion add descriptio	n										
Tests Overall Ex	ecution Status									+ Add ~		
1 FAIL Total Test	:s: 1	_										
-	Filter(s)											
·	Apply Rank								Show 100 🗸 entries	Columns 🗸		
	🗕 Rank	Key	Summary	🔶 Test	Type #R	eq	#Def	Assignee	Status			
	1	XT-142	login feature	Cucur	mber 1		0	Xpand IT Admin	FAIL	• …		
Showing '	1 to 1 of 1 entrie	es							First Previous	1 Next Last		

One of the tests fails (on purpose).

The execution screen details of the Test Run will provide overall status information and Gherkin statement-level results, therefore we can use it to analyse the failing test.

Test Details						
Test Type:	Cucumber					
Scenario Type Scenario:	e: Scenario Outline					
	3 Then I should 4 5 Examples: 6 user	ith <username> and <password> see a flash message saying <mess name password me mith SuperSecretPassword! Yo</mess </password></username>	issage			
Examples						
<userr< td=""><td>name></td><td><pre><password></password></pre></td><td><message></message></td><td></td><td>Duration</td><td>Status</td></userr<>	name>	<pre><password></password></pre>	<message></message>		Duration	Status
▶ tomsn	nith	SuperSecretPassword!	You logged into a secure area!		3861.000 ms	PASS
fooba	r	barfoo	Your username is invalid.		10952.000 ms	FAIL
bles					1115	
oles <username></username>		<pre>cpassword></pre>	<message></message>		Duration	Status
		<pre>cpassword> SuperSecretPassword!</pre>	<message> You logged into a secure area!</message>	38		Status PASS
<username></username>					Duration	
<username> tomsmith</username>		SuperSecretPassword!	You logged into a secure area!		Duration 361.000 ms	PASS
<username> tomsmith foobar</username>		SuperSecretPassword!	You logged into a secure area!		Duration 361.000 ms	PASS
<username> tomsmith foobar Steps</username>	he login page	SuperSecretPassword!	You logged into a secure area!		Duration 361.000 ms 10952.000 ms	PASS FAIL
<username> tomsmith foobar Steps Before Given I am on t</username>	he login page ith foobar and barfoo	SuperSecretPassword!	You logged into a secure area!	1	Duration 361.000 ms 10952.000 ms 2.000 ms	PASS FAIL
<username> tomsmith foobar Steps Before Given I am on t When I login w</username>		SuperSecretPassword! barfoo	You logged into a secure area!	1	Duration 361.000 ms 10952.000 ms 2.000 ms 166.000 ms	PASS FAIL
<username> tomsmith foobar Steps Before Given I am on to When I login w Then I should s Error: - Expect</username>	ith foobar and barfoo	SuperSecretPassword! barfoo	You logged into a secure area!	1	Duration 361.000 ms 10952.000 ms 2.000 ms 10052.000 10052.000 10052.000 10036.000 10036.000	PASS FAIL PASS PASS PASS
<username> tomsmith foobar Steps Before Given I am on t When I login w Then I should s Error: - Expec + Reces - Your + Your + X at</username>	th foobar and barfoo the a flash message saying Your us Expect \$(`#flash`) to have sted - 1 ved + 2 username is invalid. username is invalid. World. <anonymous> (/Users/</anonymous>	SuperSecretPassword! barfoo sername is invalid. e text containing (cristianocunha/Documents/Projec (internal/process/task_queues.;	You logged into a secure area! Your username is invalid.	1 4	Duration 361.000 ms 10952.000 ms 2.000 ms 166.000 ms 10036.000 ms	PASS FAIL PASS PASS PASS
tomsmith foobar Steps Before Given I am on t When I login w Then I should s Error: - Expec + Recei - Your + Your + Your t at at	th foobar and barfoo the a flash message saying Your us Expect \$(`#flash`) to have ted - 1 ived + 2 username is invalid. username is invalid! World. <anonymous> (/Users/ processTickSAndRejections World.executeAsync (/Users/</anonymous>	SuperSecretPassword! barfoo sername is invalid. • text containing foristianocunha/Documents/Projec (internal/process/task_gueues.j //cristianocunha/Documents/Projec	You logged into a secure area! Your username is invalid.	ns/steps.js:22:41) ils/build/shim.js:136:10 /@wdio/utils/build/test	Duration 361.000 ms 10952.000 ms 2.000 ms 1666.000 ms 184.000 ms 10036.000 ms	PASS FAIL PASS PASS PASS FAIL

Results are reflected on the covered item (e.g. Story). On its issue screen, coverage now shows that the item is OK based on the latest testing results, that can also be tracked within the Test Coverage panel bellow.

Xray Tutorials / XT-143 login feature														
🖋 Ed	it Q	Comment	Assign	More 🗸	To Do	In Progress	Done	Admin	*					
✓ Deta	ils													
Туре	2		Story						Status:	TO DO (Vie	w Workflow	v)		
Prior) Trivial						Resolution:	Unresolved				
Labe			lone											
Requ	uirement	Status:		NOK										
✓ Deset	cription													
Click	to add	description												
	-													
 Test 	Coverag	je												
										Crea	ate Test	Create	Sub-Test Execution	+ Link ~
TE	ST COVER	AGE FOR THE I	OLLOWING AN	NALYSIS SCOP	E									
\$	cope: V	ersion; Vers	ion: None -	latest execu	ition; Envi	ronment: All	Environm	ients 👻						NOK
	∓ Filte	er(s)												
P	~											5	Show 10 🗸 entries	Columns -
	$\Rightarrow \mathbf{P}$		itus	♦ Res	olution		🔺 Key		Summary		Test Runs	1	Test Status	
Ο	0	то	00	Unres	solved		XT-142	2	login feature		≣0		FAIL	
Show	ving 1 to	l of 1 entries											First Previous 1	Next Last

If we change the specification (i.e. the Gherkin scenarios), we need to import the .feature(s) once again.

Therefore, in the CI we always need to start by importing the .feature file(s) to keep Jira/Xray on synch.

FAQ and Recommendations

Please see this page.

References

- WebDriverIO
- WebDriverIO documentation