# **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	n feature	•							
🖋 E	dit Q Comment	Assign	More 🗸	To Do	In Progress	Done	Admin 🗸			
∽ De	tails									
Typ Prio Lab Reo	be: prity: pels: quirement Status:	Story O Trivial None	OVERED			Stai Res	tus: olution:	TO DO ( Unresolv	View Workflow) red	
✓ Des Clie	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 v B I U A v A v ⊘ v W v ∷≣ i≣ ⊙ v	· + ·
	Visual Text	#. م ع
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 🔻



Create another Create Cancel

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

Create Issue

Project\*

Xray Tutorials (XT)

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).

Edit O Commen	t Assign									
		More 🗸	To Do	In Progress	Done	Admin 🗸				
✓ Details										
Туре:	Story				Status	s:	TO DO (V	iew Work	flow)	
Priority:	O Trivial				Resolu	ution:	Unresolve	ed		
Labels:	None									
Requirement Status:	N	DTRUN		←						
<ul> <li>Description</li> <li>Click to add descriptio</li> <li>Test Coverage</li> </ul>	n					I	Create Test	Create S	ub-Test Execution	+ Link ~
TEST COVERAGE FOR TH	IE FOLLOWING AI	NALYSIS SCOPE								
Scope: Version; V	ersion: None -	latest execu	tion; Env	vironment: All I	Environmer	nts 👻				
<del>,</del> ∓ Filter(s) ा⊒्र ✓								Sh	ow 10 ✔ entries	Columns -
≜ P ≜ Sta	al and a sur	Resolution		Key	A Summa	rv.	Tes	t Runs	A Test Status	
	0 U	nresolved		XT-226	Test Logi	n feature	•		TODO	New Lot

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.

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

2 Luit Q Comme	nt Assign	More Y To Do In P	rogress Done	Admin 🗸					
✓ Details		Log work							
Type:	O Test	Agile Board	Sta	atus:	TO DO (View Workflow)				
Priority:	O Trivial	Rank to Top	Re	solution:	Unresolved				
Labels:	None	Rank to Bottom							
<ul> <li>Description</li> </ul>		Archive							
Click to add descriptin	on	Attach files	-						
		Attach files							
<ul> <li>Test Details</li> </ul>		Attach Screenshot	-						
Туре:	Cucumber	Voters							
Scenario Type:	Scenario	Stop watching							
Scenario:	Given I an	Watchers							
	When I Log Then I sho	Create sub-task	d <password> ge saying <me< td=""><td colspan="6">i <password> ze saving <message></message></password></td></me<></password>	i <password> ze saving <message></message></password>					
	Exampl	Convert to sub-task							
		Move	1	message	1				
		Link	tPassword!	You logged int Your username	o a secure area!   is invalid!				
		Clone							
		Labels			<u>Edit Steps</u>				
<ul> <li>Pre-Conditions</li> </ul>		Delete	_						
This test is not associ	iated with Pre-0	Reset TestRunStatus			Create Pre-Condition Associate Pre-Conditions				
		Export to Cucumber							
✓ Test Sets		Export Test to XML	]						
This test is not associ	iated with Test	Export Test Runs to CSV							
			,		Associate Test Sets				
FOT ADL (	nto here)								
EST API (more ir	/								
EST API (more ir example of a sl	hell script	to export/genera	te .feature	s from Xray					
EST API (more ir example of a sl	hell script	to export/genera	te .feature	s from Xray					
EST API (more ir example of a sl #!/bin/bash	hell script	to export/genera	ate .feature	s from Xray					
EST API (more in example of a sl #!/bin/bash	hell script	to export/genera	ate .feature	s from Xray					
EST API (more in example of a sl #!/bin/bash JIRA_BASEURL	hell script	to export/genera	ate .feature	es from Xray					
EST API (more in example of a sl #!/bin/bash JIRA_BASEURL JIRA_USERNAM	hell script	to export/genera	ate .feature	s from Xray					
REST API (more in example of a sl #!/bin/bash JIRA_BASEURL JIRA_USERNAM JIRA_PASSWOR	hell script ,=https: IE=admin 2D=admin	to export/genera	ate .feature	s from Xray					
REST API (more in example of a sl #!/bin/bash JIRA_BASEURL JIRA_USERNAM JIRA_PASSWOR KEYS="XT-142	hell script =https: IE=admin 2D=admin	to export/genera	ate .feature	⊧s from Xray					
EST API (more in example of a sl #!/bin/bash JIRA_BASEURL JIRA_USERNAM JIRA_PASSWOR KEYS="XT-142	hell script	to export/genera	ate .feature	s from Xray					
EST API (more in example of a sl #!/bin/bash JIRA_BASEURL JIRA_USERNAM JIRA_PASSWOR KEYS="XT-142 rm -f featur curl -u \$JIR	hell script =https: E=admin D=admin " es.zip RA USERN	to export/genera //192.168.2.16	ate .feature	s from Xray	URL/rest/raven/2.0/export/te				

• 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.

atures/1_XT-225.feature
REQ_XT-225
@TEST_XT-226
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
essage
tomsmith   SuperSecretPassword!   You logged into a secure
foobar   bartoo   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.

The CucumberJson reporter does not produce reports containing the screenshots embedded.

However, we have added an After scenario to do so, but you can use the HTML or Cucumber reporter that allows you to add those as you can find more info here.

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\_PASSWORD --data @".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.

Execut	ion results	[162400	4905	נססכ									
Edit Q Cor	mment Assig	n More 🗸	To Do	In Progress	Done	Admin 🗸							
Details												✓ Xpor	ter
Type:	🕨 Test E	recution			Stat	us:	TO DO (Vi	iew Workflow)				Temp	late
Priority:	O Trivial				Res	olution:	Unresolve	d					
abels:	None											Outp	ut for
rest Flan. Fest Environment	ts: None												
Description												<ul> <li>Peop</li> </ul>	le
Click to add desc	cription											Assig	nee:
Tests												Repo	rter:
											1.0.1.1	Votes Wate	:: hers:
											+ Add V	mato	10101
Overall Execution S	Status											✓ Date:	5
1												Creat	ed:
FAIL												Upda	ted:
Total Tests: 1												✓ Agile	
												View	on E
÷ Filter(s)													
Apply Apply	y Rank							Show 100	✓ entries	0	olumno –		
											olullins 🕈		
– Rar	nk 🔺 Kev	Summary	≜ Tes	st Type #R	ea	#Def A	ssignee	≜ Status		Ĭ	olullins 🕈		
🔻 Rar	nk 🍦 Key	Summary	Tes	st Type #R	eq	#Def A	ssignee	Status					
<mark>▼</mark> Rar	nk ≜ Key XT-142	Summary	∳ Tes	st Type #R umber 1	eq	#Def A	ssignee Ipand IT Admin	Status			• ···		
▼ Rar □ 1 Showing 1 to 1 of 1	nk	Summary		st Type #R umber 1	eq	#Def A	ssignee pand IT Admin	Status FAIL	Previous	5 1	Execution	ion Details	
■ Rar □ 1 Showing 1 to 1 of 1	nk 🔶 Key XT-142 I entries	♣ Summary login feature	Cuci	st Type #R umber 1	eq	#Def A	ssignee Spand IT Admin	♣ Status FAIL	Previous	5 1	EXECUTE IN	ion Details	
Rar	nk ∳ Key XT-142 I entries	Summary login feature	∳ Te: • Cuci	st Type #R umber 1	eq	#Def A	ssignee	Status FAIL First	Previous	5 1	Execute IN PASS	ion Details	
Rar	nk ¢ Key XT-142 I entries	Summary	∳ Tes	st Type #R	eq	#Def A	ssignee	♦ Status FAIL First	Previous	5 1	Execute IN PASS	ion Details	
Rar Nowing 1 to 1 of 1 Attachments Tutorials / Test Exe n feature	nk ∳ Key XT-142 I entries	Summary login feature est: XT-142	∳ Tes	st Type #R umber 1	eq	#Def A	ssignee (pand IT Admin	Status  All  FAIL  First  Import Execut	Previous	s 1 Export	Execute IN EXECUTE IN PASS to Cucumber	ion Details	ēst E)
Rar Rar Showing 1 to 1 of 1 Attachments Tutorials / Test Exe in feature Test Details	nk \$Key XT-142 I entries	Summary login feature est: XT-142	Tes	st Type #R	eq	#Def A	ssignee (pand IT Admin	Status FAIL First Import Execut	Previous	s 1 Export	Execution Execut	ion Details NLINE Return to 1	est E
Rar Rar Rav	nk \$Key XT-142 I entries	Summary login feature est: XT-142	Tes	st Type #R	eq	#Def A	ssignee (pand IT Admin	Status     FAIL     First     Import Execut	Previous	s 1 Export	Execut     Execut     Execut     Execut     Cucumber	ion Details	ēst E
Rar Rav	nk ¢ Key XT-142 I entries secution: XT-153 / °	Summary login feature est: XT-142	Tes	st Type #R umber 1	eq	#Def A	ssignee (pand IT Admin	Status FAIL First Import Execut	Previous	s 1 Export	Execute IN PASS	ion Details	est Ex
▼ Ran       □     1       Showing 1 to 1 of 1       Attachments       Tutorials / Test Exection feature       Test Details       Test Type:       Scenario Type:       Scenario:	k ¢ Key XT-142 I entries secution: XT-153 / Cucumber Scenario Outline 1 Given 1 (2)	Summary login feature est: XT-142	tes	st Type #R	eq	#Def A	ssignee	Status FAIL First Import Execut	Previous	s 1 Export	Execute in PASS	ion Details	est Ex
Rar Rattachments Attachments Tutorials / Test Exec in feature Test Details Test Type: Scenario Type: Scenario:	nk	Summary login feature est: XT-142 m on the login p gin with <userna uuld see a flash</userna 	Test Cuck Cuck age me> and message s	st Type #R umber 1 - - assword> aying <message< td=""><td>eq</td><td>#Def A</td><td>ssignee (pand IT Admin</td><td>Status     FAIL     First     Import Execut</td><td>Previous</td><td>Export</td><td>Execute in PASS</td><td>ion Details</td><td>est E</td></message<>	eq	#Def A	ssignee (pand IT Admin	Status     FAIL     First     Import Execut	Previous	Export	Execute in PASS	ion Details	est E
Rar Rat Showing 1 to 1 of 1 Attachments Tutorials / Test Exe n featUre Test Type: Scenario Type: Scenario:	k \$Key XT-142 L entries cucumber Scenario Outline 1 Given I of 3 Then I st 4 5 Examp 6 Examp	Summary login feature est: XT-142 n on the login p gin with <userna uld see a flash les: username   pass</userna 	Te: Cuct e Cuct me> and <p me&gt;sage me&gt;sage s word</p 	st Type #R umber 1 - 	eq 	#Def A	ssignee (pand IT Admin	Status FAIL First Import Execut	Previous	Export	Execute IN EXECUTE IN PASS to Cucumber	ion Details	est Ez
▼ Rar         □       1         Showing 1 to 1 of 1         Attachments         Tutorials / Test Exe         In feature         Test Details         Test Type:         Scenario Type:         Scenario:	k \$Key XT-142 L entries cucumber Scenario Outline Scenario Outline Scenario Outline Scenario Outline Scenario State Scenario State State Scenario State Scenario State S	Summary login feature login feature est: XT-142 est: XT-142 est: xt-142 est: xt-142 est: xt-142 est: username uld see a flash les: username   pass tomsmth   Super foobar   Super fooba	Test e Cucu me> and <p message s word rsecretPas oo</p 	st Type #R Jmber 1 assword> aying <message sword!   You T   You T</message 	eq ge ggged into username i	#Def A	ssignee	Status FAIL First Import Execut	Previous	Export	Execute IN PASS	ion Details	est E
▼ Rar       □     1       Showing 1 to 1 of 1       Attachments       Tutorials / Test Exection feature       Test Details       Test Type:       Scenario Type:       Scenario:	nk	Summary login feature ast: XT-142 st: XT-142 st: xt-142 username   pass foobar   barf	¢ Te: : Cuck : Cuck me> and <pre> cpret for the second second</pre>	st Type #R umber 1 assword> aying <message sword!   messa your Your</message 	eq > ge ogged into username i	#Def A 0 >	ssignee	Status     FAIL     First     Import Execut	in Results	Export	Execute in PASS	ion Details	est E
Rar Rat Showing 1 to 1 of 1 Attachments Tutorials / Test Exe in featUre Test Type: Scenario Type: Scenario:	nk	Summary login feature est: XT-142 mon the login p gin with <userna uld see a flash les: username   pass foobar   barf</userna 	¢ Te: Cucu age me> and <p message s word r5ecretPas oo</p 	st Type #R umber 1 assword> aying <message sword!   messa   Your</message 	eq > ge ogged into username i	#Def A 0 >	ssignee (pand IT Admin () () () () () () () () () () () () ()	Status     FAIL     First     Import Execut	Previous	Export	Execute in PASS	ion Details	ēst E
▼ Rar         □       1         Showing 1 to 1 of 1         Attachments         Tutorials / Test Exe         In feature         Test Details         Test Type:         Scenario Type:         Scenario:	nk	Summary login feature est: XT-142 n on the login p gin with <userna uld see a flash les: username   pass tomsmith   Supe foobar   barf</userna 	Test e Cuca age me> and <p message s word rSecretPas oo</p 	st Type #R umber 1 assword> aying <message sword!   You 1   Your</message 	eq > ge ogged into username i	#Def A 0 >	ssignee (pand IT Admin	Status FAIL First Import Execut	Previous	Export	Execute in PASS	ion Details	est E
▼ Rar         □       1         Showing 1 to 1 of 1         Attachments         Tutorials / Test Exe         In feature         Test Details         Test Type:         Scenario Type:         Scenario:	nk \$ Key XT-142 I entries ecution: XT-153 / - Cucumber Scenario Outline 1 Given I of 2 When I to 3 Then I st 4 5 Examp 6 7 8	Summary login feature est: XT-142 n on the login p gin with <userna ould see a flash les: username   pass tomsmith   Supe foobar   barf</userna 	¢ Tet c Cucu age me> and <p message s word rSecretPas oo</p 	st Type #R Jmber 1 assword> aying <message sword!   You 1   Your</message 	eq > ge ogged into username i	#Def A 0  X a secure area s invalid.	ssignee pand IT Admin	Status FAIL First Import Execut	Previous	Export	Execute in PASS	ion Details	est Ex
▼ Ran         □       1         Showing 1 to 1 of 1         Attachments         Totorials / Test Exection feature         Test Details         Test Type:         Scenario Type:         Scenario:	nk	Summary login feature est: XT-142 non the login p gin with <userna buld see a flash les: username   pass foobar   barf</userna 	e Test	st Type #R umber 1 assword> aying <message sword!   messa your</message 	eq > ge ogged into username i	#Def A 0  X a secure area s invalid.	ssignee (pand IT Admin  pand IT Admi	Status     FAIL     First     Import Execut	Previous     on Results	Export	Execute IN PASS	ion Details	rest E
Rar Rar Showing 1 to 1 of 1 Attachments Tutorials / Test Exe in feature Test Type: Scenario Type: Scenario: Scenario: Examples	nk	Summary login feature ast: XT-142 n on the login p gin with <userna buld see a flash les: username   pass foobar   barf</userna 	¢ Te: Cucu age me> and <p message s word rSecretPas oo</p 	st Type #R umber 1 assword> aying <message sword!   messa sword!   You 1   You r</message 	eq > ge ogged into username i	#Def A 0 >	ssignee (pand IT Admin (pand IT Admin )  1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	Status     FAIL     First     Import Execut	Previous     on Results	Export	Execute in PASS To Cucumber	ion Details	est Ex
Rar Rar Attachments Tutorials / Test Exe in feature Test Type: Scenario Type: Scenario: Examples	nk ¢ Key XT-142 I entries ecution: XT-153 / ^ Cucumber Scenario Outline 1 Given I of 2 When I to 3 Then I st 4 5 Example 6 7 8	Summary login feature est: XT-142 n on the login p gin with <userna uld see a flash les: username   pass tomsmith   Supe foobar   barf</userna 	Tet e Cuct age me> and me> and me>sage s word rSecretPasioo	st Type #R umber 1 assword> aying <message sword!   You 1   Your</message 	eq > ge ogged into username i	#Def A 0 >	ssignee (pand IT Admin	Status FAIL First Import Execut	Previous     on Results	Export	Execute in PASS To Cucumber	ion Details	est Ex
Rar          Rar         I         Showing 1 to 1 of 1         Attachments         Totorials / Test Exe         In feature         Test Details         Test Type:         Scenario Type:         Scenario:	Ak \$Key XT-142 I entries Execution: XT-153 / - Cucumber Scenario Outline 1 Given I of 2 When I to 3 Then I sh 4 5 Examp 6 7 8	Summary login feature ast: XT-142 m on the login p gin with <userna ould see a flash les: username   pass tomsmith   Supe foobar   barf foobar   barf superior</userna 	Test cucu age me> and cp message s word rSecretPass oo	st Type #R umber 1 assword> aying <message sword!   You 1   You 1</message 	eq > ge ogged into username i	#Def A 0  X a secure area s invalid.	ssignee	Status FAIL First Import Execut	Previous	Export	Execute in PASS     to Cucumber	ion Details	est Ex

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
	<pre>Error: Expect \$(`#flash`) to hav     Expected - 1     Received + 2     Your username is invalid.     Your username is invalid!     x     at World.<anonymous> (/Users     at processTickAAndRejections     at World.testFrameworkFnWrap     at World.testFrameworkFnWrap</anonymous></pre>	<pre>e text containing /cristianocunha/Documents/Proj (internal/process/task_queues /cristianocunha/Documents/Pro per (/Users/cristianocunha/Doc</pre>	ects/cucumber_webdriverio/features/step-definition .js:93:5) jects/cucumber_webdriverio/node_modules/@wdio/util uments/Projects/cucumber_webdriverio/node_modules/	s/steps.js:22:41) s/build/shim.js:136:16) @wdio/utils/build/test-framework	/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.

or Io	ay Tutorials / X Digin featur	T-143 <b>e</b>								
🖋 Edit	Q Comment	Assign	More 🖌 To Do	In Progress	Done	Admin	~			
✓ Details										
Type:		Story				5	Status:	TO DO (View Workflow)		
Priority:		O Trivial				F	Resolution:	Unresolved		
Labels:		None								
Requirer	nent Status:	( N	ок							
<ul> <li>Descript</li> </ul>	tion									
Click to	add description									
<ul> <li>Test Cov</li> </ul>	/erage									
								Create Test	Create Sub-Test Execution	+ Link ×
TEST C	OVERAGE FOR THE	FOLLOWING ANA	ALYSIS SCOPE							
Scor	be: Version: Ver	rsion: None - la	atest execution: Env	vironment: All	Environme	ents 👻			_	NOK
	,		,							
=	Filter(s)									
<b>·</b> ·									Show 10 🗸 entries	Columns -
\$	P ∳St	tatus	Resolution		🔺 Key		Summary	Test Runs	rightarrow Test Status	
	О то	DO	Unresolved		XT-142		login feature	IO <b>-</b>	FAIL	
Showing	1 to 1 of 1 entries	3							First Previous	1 Next Last

## 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	feature	)							
🖋 Edit	Q Comment	Assign	More 🗸	To Do	In Progress	Done	Admin 🗸			
✓ Details										
Type: Priority: Labels: Requirem	nent Status:	Story Trivial None UNC	OVERED			Stat Res	us: olution:	TO DO ( Unresolv	View Workflow) /ed	
<ul> <li>Descripting</li> <li>Click to a</li> </ul>	ion add description									
✓ Test Cov	erage									
No Tests	were found test	ing the requ	irement.					Create Test	Create Sub-Test Execution	+ Link v
<ul> <li>Attachm</li> </ul>	ents									•••
					C Drop 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();
```

## 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/s	vnchronize Sce	nario(s)/Backg	round(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

Sedit Q Comm	nent Assign	More Y To Do In P	rogress Done Admin 🗸	
✓ 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		
<ul> <li>Description</li> </ul>		Archive		
Click to add descrip	tion	Attach files		
		Attach Screenshot		
<ul> <li>Test Details</li> </ul>	Guanda	Votoro		
Type:	Cucumper	Stop watching		
Scenario Type:	Scenario	Watchers		
Scenario:	Given I ar		d <password></password>	
	Then I sho	Create SUD-Task	ge saying <message></message>	
	Examp			
		Move	message Password!   You logaed in	nto a secure area!
	i	Link	Your username	e is invalid!
		Cione		🗾 Edit Steps
		Labels		
<ul> <li>Pre-Conditions</li> </ul>		Delete		
This test is not asso	ciated with Pre-	Reset TestRunStatus		Create Pre-Condition Associate Pre-Conditions
		Export to Cucumber		
<ul> <li>Test Sets</li> </ul>		Export Test to XML		
This test is not asso	ciated with Test	Export Test Runs to CSV		
				Associate lest Sets
	info horo)			
CEST AFT (IIIOIE	into here)			
example of a s	shell scrip	t to export/genera	te .features from Xra	y .
#!/bin/bash				
	T 1	//100 100 0 1/	- 0	
	MF-admin	//192.100.2.10	00	
JIRA_BASEUR	ann – aum Lli			
JIRA_BASEUR JIRA_USERNA JIRA_PASSWO	RD=admin			
JIRA_BASEUR JIRA_USERNA JIRA_PASSWO KEYS="XT-14	RD=admin 2"			
JIRA_BASEUR JIRA_USERNA JIRA_PASSWO KEYS="XT-14	RD=admin 2"			
JIRA_BASEUR JIRA_USERNA JIRA_PASSWO KEYS="XT-14 rm -f featu	RD=admin 2" res.zip			
JIRA_BASEUR JIRA_USERNA JIRA_PASSWO KEYS="XT-14 rm -f featu curl -u \$JI	RD=admin 2" res.zip RA_USERN	AME:\$JIRA_PASS	SWORD "\$JIRA_BAS	EURL/rest/raven/2.0/export/test

• 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.

```
      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.

P E	ray Tutorials / >	(T-153 <b>'esults [</b> '	1624004	4965	566]						
🖋 Edit	Q Comment	Assign	More 🗸	To Do	In Progress	Done	Admin 🗸				
✓ Details											
Type:		Test Exec	cution			Sta	itus:	TO DO (Vie	w Workflow)		
Priority	:	O Trivial				Res	solution:	Unresolved			
Labels:		None									
Test Pla	an:	None									
lest En	vironments:	None									
✓ Descrip	otion										
Click to	add description	,									
<ul> <li>Tests</li> </ul>											
										+ Add	
Overall	Execution Status										
Overall	Execution Status										
1											
FAIL	-										
T-4-1 T-											
lotal le	STS: 1										
	⇒ Filter(s)										
	Apply Rank								Show 100 x ont	rios Columns -	
<b>_</b> <i>y</i>										ies columns •	
	🔻 Rank	Кеу	Summary		st Type #	Req	#Def	Assignee	Status		
	1	XT-142	login feature	Cuc	umber 1		0	Xpand IT Admin	FAIL	• …	
Showing	g 1 to 1 of 1 entrie	s							First Prev	vious 1 Next Las	t

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 Type::       Cucumber         Security Type::       Security Type::         Security Type::       Securitype::         Secu	est Details				
Security open sources of a generation of generatio	Test Type: Cuc	umber			
cuername>       queename>	Scenario: 1 Scenario: 1 3 4 5 6 7 7 8	ario Outine Given I am on the login page When I login with <username> and <pass Then I should see a flash message sayi Examples: I username   password I tonsmith   SuperSecretPasswo I foobar   barfoo</pass </username>	word> ng ⊲message>   message   rd!   You logged into a secure area!     Your username is invalid.		
cutername>cpassword>cmessage>DurationStattomsmithSuperSecretPassword!You logged into a secure area!3861.000 msPASfoobarbarfooYour username is invaild.10952.000PAesStatusPASStomsmithSuperSecretPassword!You logged into a secure area!3861.000 msPASStomsmithSuperSecretPassword!You logged into a secure area!3861.000 msPASStomsmithSuperSecretPassword!You logged into a secure area!3861.000 msPASStomsmithSuperSecretPassword!You logged into a secure area!3861.000 msPASStoobarbarfooYour username is invaild.10952.000FAILstore10952.000msPASSGiven I an on the login page484.000 msPASSWhen I login with foobar and barfoo10952.000FAILThen I should see a flash message saying Your username is invaild.10952.000msstroer:Stepeet 4 - 110952.000FAIL* Expect 5(*#flash') to have text containing10952.000ms* Stepeet 4 - 110952.000ms* Kocial:10952.000ms* Stepeet 5(*#flash') to have text containing </td <td>xamples</td> <td></td> <td></td> <td></td> <td></td>	xamples				
cusernamescusernamescusernamesDurationStatetomsmithSuperSecretiPassword!You logged into a secure area!3861.000 msPASfoobarbarlooYour username is invalid.10952.000FAIes					
tonsmith       SuperSecretPassword!       You logged into a secure area!       3981.000 ms       PAS         foobar       barfoo       Your username is invalid.       109502.000 ms       PA         es	<username></username>	<password></password>	<message></message>	Duration	Status
ses	foobar	SuperSecretPassword	You logged into a secure area! Your username is invalid.	10952.000 ms	FAIL
tomsmithSuperSecretPassword!You logged into a secure area!3861.000 msPASSfoobarbarfooYour username is invalid.10952.000 msFAILSteps2.000 msPASSGiven I am on the login page166.000 msPASSWhen I login with foobar and barfoo484.000 msPASSThen I should see a flash message saying Your username is invalid.10036.000 msFAILError: Expect \$ (*#flash') to have text containingFAIL- Stoperde - 1 + Received + 2 - Your username is invalid Stoperde - 1 + Received + 2 - Your username is invalid Your username is invalid You esername is invalid You rusername is invalid.+ at World. <anonymous> (/Users/cristianocunha/Documents/Projects/cucumber_vebdriverio/features/step-definitions/steps.js:22:41)</anonymous>	s				
Inode and barloo     Tour username is invalid.     10952,000 ms     IN952,000 ms       Steps     Steps     2.000 ms     PASS       Given I am on the login page     166.000 ms     PASS       When I login with foobar and barloo     484.000 ms     PASS       Then I should see a flash message saying Your username is invalid.     10036.000 ms     FAIL       Error: Expect \$(*#flash`) to have text containing     -     Farce is invalid.       - Your username is invalid.     - Your username is invalid.     -       + Your username is invalid.     + Your username is invalid.     + Your username is invalid.       + Your username is invalid.     + Your username is invalid.     + Your username is invalid.	S :username>	<pre>cpassword&gt;</pre>	<message></message>	Duration	Status
Steps       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 username is invalid.       10036.000 ms       FAIL         Error: Expect \$(*#flash') to have text containing       Error: Expect 4 - 1       +         - Your username is invalid.       + Your username is invalid.       + Your username is invalid.         + Your username is invalid.       + Your username is invalid.       + Your username is invalid.	susername>	<pre><password> SuperSecretPassword! </password></pre>	<message> You logged into a secure area!</message>	Duration 3861.000 ms	Status PASS
getore       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 username is invalid.       10036.000 ms       FAIL         Error: Expect \$ (`#flash`) to have text containing       -       -         - Expected - 1       +       Received + 2       -         - Your username is invalid.       + Your username is invalid!       + ×         * at World. <anonymous> (/Users/cristianocunha/Documents/Projects/cucumber_webdriverio/features/step-definitions/steps.js:22:41)       -</anonymous>	username> omsmith oobar	<password> SuperSecretPassword! barfoo</password>	<message> You logged into a secure area! Your username is invalid.</message>	Duration 3861.000 ms 10952.000 ms	Status PASS FAIL
Entromander lagen pege       1000.000 mis       1783 s         When I login with foobar and barfoo       484.000 ms       PASS         Then I should see a flash message saying Your username is invalid.       10036.000 ms       FAIL         Error: Expect \$(`#flash`) to have text containing       -       Expected - 1         + Received + 2       -       Your username is invalid.       + Your username is invalid!         + x       at World.       vesdriverio/features/step-definitions/steps.js:22:41)	susername> omsmith oobar teps	<password> SuperSecretPassword! barfoo</password>	<message> You logged into a secure area! Your username is invalid.</message>	Duration 3861.000 ms 10952.000 ms	Status PASS FAIL
Then I should see a flash message saying Your username is invalid. Error: Expect \$(`#flash`) to have text containing - Expected - 1 + Received + 2 - Your username is invalid. + Your username is invalid! + x at World. <anonymous> (/Users/cristianocunha/Documents/Projects/cucumber_webdriverio/features/step-definitions/steps.js:22:41)</anonymous>	username> omsmith oobar iteps lefore iteps	<password> SuperSecretPassword! barfoo</password>	<message> You logged into a secure area! Your username is invalid.</message>	Duration 3861.000 ms 10952.000 ms 2.000 ms 166.000 ms	Status PASS FAIL PASS PASS
<pre>Fror: Expect \$(`#flash`) to have text containing     Expected = 1     Received + 2     Your username is invalid.     Your username is invalid!     **     at World.<anonymous> (/Users/cristianocunha/Documents/Projects/cucumber_webdriverio/features/step-definitions/steps.js:22:41)</anonymous></pre>	username> omsmith oobar iteps Jefore Jiven I am on the login page	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	<message> You logged into a secure area! Your username is invalid.</message>	Duration 3861.000 ms 10952.000 ms 2.000 ms 166.000 ms 484.000 ms	Status PASS FAIL PASS PASS PASS
at processTickSAndRejections (internal/process/task_queues.js:93:5) at World.executeAsync (/Users/cristianocunha/Documents/Projects/cucumber_webdriverio/node_modules/@wdio/utils/build/shim.js:136:16) at World.testFrameworkFMWrapper (/Users/cristianocunha/Documents/Projects/cucumber_webdriverio/node_modules/@wdio/utils/build/test_framework/testFTWrapper	username> omsmith oobar iteps Jefore Siven I am on the login page Vhen I login with foobar and hen I should see a flash me	ssage saying Your username is invalid.	<message> You logged into a secure area! Your username is invalid.</message>	Duration 3861.000 ms 10952.000 ms 2.000 ms 166.000 ms 484.000 ms 10036.000 ms	Status PASS FAIL PASS PASS PASS FAIL
at noise trameworkrentapper (/users/cristianocunna/bocuments/rojects/cucumber_webariverio/node_modules/ewalo/dtls/bulla/test-rramework/testrnwrapper.	s cusername> consmith coobar coobar iteps lefore le	<pre><pre><pre><pre><pre><pre><pre><pre></pre></pre></pre></pre></pre></pre></pre></pre>	<message> You logged into a secure area! Your username is invalid. s/Projects/cucumber_webdriverio/features/step-definiti gueues.js:93:5) ts/Projects/cucumber_webdriverio/node_modules/#winiviti</message>	Duration 3861.000 ms 10952.000 ms 2.000 ms 2.000 ms 166.000 ms 484.000 ms 10036.000 ms 10036.000 ms	Status PASS FAIL 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	<b>v</b> )		
Prior	ity:	(	) Trivial						Resolution:	Unresolved				
Labe	els:		lone											
Requ	uirement	Status:		NOK										
✓ Deset	cription													
Click	to add	description												
<ul> <li>Test</li> </ul>	Covera	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
	<del>∓</del> Filte	er(s)												
P	~											5	Show 10 🗸 entries	Columns -
	$\Rightarrow \mathbf{P}$		itus	♦ Res	olution		🔺 Key		Summary		Test Runs	1	Test Status	
	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