

Examples using Test Case Importer

 Overview **What you'll learn**

Prerequisites

- Examples

- ## Examples
- Import different test cases (with different fields and options)
 - Invalidate in Jira that the test cases are available
 - Import two manual test case with labels
 - Import two manual test cases to multiple projects
 - Import two manual test cases with Precondition and Test Set
 - Import two manual test cases with Precondition, Test Set and Test Repository
- ## Source-code for this tutorial
- Import two manual test cases with test step custom field
 - Import two manual test cases with custom field and labels
 - All examples are available in [Github](#)
 - With links to requirement
 - Update two manual test cases

Source-code for this tutorial

- Import two manual test cases with custom field and labels
- All examples are available in [GitHub](#)
 - Import generic and generic test cases
 - With links to requirement
 - Update two manual test cases

- Tips

Overview

Test Case Importer is used to import test cases in a CSV format into Xray, this functionality is used to migrate from other tools or to import manual test cases written in other formats into Xray.

In this tutorial we will provide different examples of how to import test case in the CSV format into Xray (Xray also have available the importation from Json but we are not focusing in Json in this tutorial).

If you have doubts on where to find the tool or use it please check more information in the [Test Case Importer](#) page.

Prerequisites

For these examples we will use the [Test Case Importer](#) tool, which is present when you install Xray in your Jira instance.

We will need:

- Access to a Jira instance with Xray installed
- Only users that have the **Make Bulk Changes** permission will have this option available. Additionally, the user will only be able to import data into projects where he has the **Create Issues** permission.

Examples

In each example we will have available the CSV file corresponding to the test cases definition and the configuration file used in the Test Case Importer to perform the operation.

We will showcase different possibilities when importing test cases into Xray, using combination of fields and possibilities that should cover most usages.

All of the examples in this tutorial have a CSV file with the definition of the test cases to import and a configuration file that will configure all associations and fields for the importation to be successful.

Import two manual test cases

The most usual usage of the Test Case Importer is to import manual test cases, so we are going to focus in this workflow and add some variations that should match the majority of importation cases we have seen.

The simplest case is to import two manual test cases, defining the test steps, summary, priority and component.

i The next examples are using Components existent in our project, to import successfully make sure that **visits** exists as a Component or replace it with Components that exists in your project.

Any Key/id used must exist in your project also.

Make sure your are using the Test Priority values that are defined in your project.

import_two_manual_tests.csv

```
Issue Id;Issue key;Test type;Test Summary;Test Priority;Component;Action;
Data;Result
1;;Manual;Test 1 for user story WEB-1;High;Visits;Go to login page;;
1;;Manual;;;Visits;Enter username;peter;
1;;Manual;;;Visits;Enter password;pwrld23;
1;;Manual;;;Visits;Click login button;;User successfully logged in
2;;Manual;Test 2 for user story WEB-1;High;Visits;Go to login page;;
2;;Manual;;;Visits;Enter username;peter;
2;;Manual;;;Visits;Enter password;abcl23;
2;;Manual;;;Visits;Click login button;;User logged in fails with message
```

You will find the configuration used in the file with the same name but with the extension Json (*two_manual_tests.json*).

Once imported we can see two new test cases with all the properties defined in the CSV.

The screenshot displays two test case entries in a software interface. Each entry has a 'Test details' panel on the left and a 'Details' panel on the right. The 'Test details' panel for 'Test 1' shows a 'Test Type' dropdown set to 'Manual' and a table with columns 'Action', 'Data', 'Expected Result', and 'Result'. The 'Details' panel for 'Test 1' shows fields for 'Assignee' (Unassigned), 'Reporter' (Chiraleen Cutha), 'Environment' (Create Branch), 'Labels' (None), 'Components' (None), 'Priority' (High), 'Automation' (No instructions), and 'Test Status' (Open Test Status). The 'Test details' panel for 'Test 2' shows a 'Test Type' dropdown set to 'Manual' and a table with columns 'Action', 'Data', 'Expected Result', and 'Result'. The 'Details' panel for 'Test 2' shows fields for 'Assignee' (Unassigned), 'Reporter' (Chiraleen Cutha), 'Environment' (Create Branch), 'Labels' (None), 'Components' (None), 'Priority' (High), 'Automation' (No instructions), and 'Test Status' (Open Test Status).

Import two manual test cases with descriptions

In this example we are importing two manual tests cases with a description.

A *Description* field was added to add the information, this field accept Wiki Markup.



The next examples are using Components existent in our project, to import successfully make sure that **Visits** exists as a Component or replace it with Components that exists in your project.

Any Key/id used must exist in your project also.

Make sure you are using the Test Priority values that are defined in your project.

import_two_manual_tests_description.csv

```
Issue Id;Issue key;Test type;Test Summary;Test Priority;Component;Action;
Data;Result;Description
1;;Manual;Test 1 for user story WEB-1;High;Visits;Go to login page;;; "My
description \\ multiline and *strong*"
1;;Manual;;;Visits;Enter username;peter;;
1;;Manual;;;Visits;Enter password;pwr123;;
1;;Manual;;;Visits;Click login button;;User successfully logged in;
2;;Manual;Test 2 for user story WEB-1;High;Visits;Go to login page;;; "
[Xray|http://www.getxray.app]"
2;;Manual;;;Visits;Enter username;peter;;
2;;Manual;;;Visits;Enter password;abc123;;
2;;Manual;;;Visits;Click login button;;User logged in fails with message;
```

As we can from the CSV above we have added two different descriptions that after the import will appear in Jira:

- The description added to the first test where we can see that is a multiline text with the word **strong** in bold

Projects / Essentials Workshop ... / EWB-92

Test 1 for user story WEB-1

Attach Create subtask Link issue Test details

Description
My description
multiline and **strong**

Environment
None

Test details

Test details Preconditions Test Sets Test Plans Test Runs

Test Type
Manual

Test Repository
Dataset

Edit in Dialog Add Step

1	Action	Data	Expected Result
	Go to login page	None	None
	MyBrowser	None	
	Attachments (0)		

- The description of the second test where. we are adding a link to an application.

Projects / Essentials Workshop ... / EWB-93

Test 2 for user story WEB-1

[Attach](#)
[Create subtask](#)
[Link issue](#)
[Test details](#)

Description

Xray

Environment

None

Test details

[Test details](#)
[Preconditions](#)
[Test Sets](#)
[Test Plans](#)
[Test Runs](#)

Test Type

Manual

Test Repository

Dataset

[Edit in Dialog](#)
[Add Step](#)

1	Action	Data	Expected Result
	Go to login page	None	None
	MyBrowser	None	
	Attachments (0)		

Import two manual test case with labels

One common case is also the usage of labels, in the following example we are inserting a new field called *Label*. When you need to import more then one *Label* you must use quotes and divide the multiple values with a separator, in our case we have used: "*Label1;Label2*".

We will see that after importing these test cases they will have associated the labels added in this CSV.



The next examples are using Components existent in our project, to import successfully make sure that **visits** exists as a Component or replace it with Components that exists in your project.

Any Key/id used must exist in your project also.

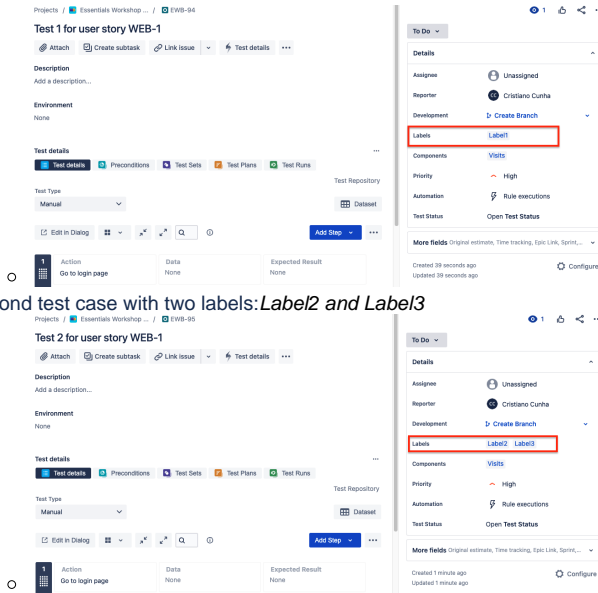
Make sure your are using the Test Priority values that are defined in your project.

import_two_manual_tests_description_labels.csv

```
Issue Id;Issue key;Test type;Test Summary;Test Priority;Component;Action;
Data;Result;Labels
1;;Manual;Test 1 for user story WEB-1;High;Visits;Go to login page;;;
Label1"
1;;Manual;;;Visits;Enter username;peter;;
1;;Manual;;;Visits;Enter password;pwr123;;
1;;Manual;;;Visits;Click login button;;User successfully logged in;
2;;Manual;Test 2 for user story WEB-1;High;Visits;Go to login page;;;
Label2;Label3"
2;;Manual;;;Visits;Enter username;peter;;
2;;Manual;;;Visits;Enter password;abc123;;
2;;Manual;;;Visits;Click login button;;User logged in fails with message;
```

When importing these test cases we have:

- one test case with *Label1*



- a second test case with two labels: *Label2* and *Label3*

Import two manual test cases to multiple projects

When importing manual Tests you can define to which project you want to import each Test by defining it in the CSV instead of defining it globally for all tests in the configuration.

With this mechanism you can import Tests to different projects seamlessly.

The CSV file we are using to import these Tests is the same as the previous one plus a field for the *Project Key*.



The next examples are using Components existent in our project, to import successfully make sure that *visits* exists as a Component or replace it with Components that exists in your project.

Any Key/id used must exist in your project also.

Make sure your are using the Test Priority values that are defined in your project.

import_two_manual_tests_project.csv

```
Issue Id;Issue key;Test type;Test Summary;Test Priority;Component;Action;
Data;Result;Project key
1;;Manual;Test 1 for user story WEB-1;High;Visits;"Go to
login page";;EWB
1;;Manual;;;Visits;Enter username;peter;;
1;;Manual;;;Visits;Enter password;pwr123;;
1;;Manual;;;Visits;Click login button;;User successfully logged in;
2;;Manual;Test 2 for user story WEB-1;High;Visits;"Go to
login page";;XT
2;;Manual;;;Visits;Enter username;peter;;
2;;Manual;;;Visits;Enter password;abc123;;
2;;Manual;;;Visits;Click login button;;User logged in fails with message;
```

Notice that you must use valid Project Keys (that exists in your Jira installation) otherwise the import will fail.

In order to use the Project Keys defined in the CSV file we must check the option in the configuration (if you are using the configuration file provided it will take care of this for you).

Test Case Importer 1

CSV File import

File import

Setup

Map fields

Setup

Project

Select Project
☒ Defined in the CSV for every test

List value delimiter

,

Date format

DD/MM/YYYY HH:mm

☒ Create Test Repository folders if needed

Next

Back

Once the import operation is successful we will have two new manual test cases in two different projects.

In order to have this example working for you you must replace the Project Keys (*EWB*, *XT*) by Project Keys that exists in your Jira instance.

The top screenshot shows the 'Test 2 for user story WEB-1' in the 'EWB' project. The 'Test details' tab is active, showing a table with two test steps. The 'Details' sidebar on the right shows the test is 'Unassigned' and has a 'High' priority. The bottom screenshot shows the 'Test 1 for user story WEB-1' in the 'XT' project. The 'Test details' tab is active, showing a table with two test steps. The 'Details' sidebar on the right shows the test is 'Unassigned' and has a 'High' priority.

Import two manual test cases with Precondition and Test Set

In the next example we are importing two manual Tests and associate them to a Test Set and two PreConditions (that we are creating with this file also).

We have added three new fields:

- Test Set - That can have the issue key from Jira of the Test Set that you want to associate or the Issue Id of the Test Set if we are creating it in the same CSV file. In our case we will use the Issue Id of the test case we are creating with the CSV (before the manual test case).
- Precondition - The Precondition issue key that you want to associate that exists in your Jira project or the Issue Id of the Precondition that we are importing.
- Precondition Type - Must have the Precondition type that we want to import, in our case we are using *Manual*.



The next examples are using Components existent in our project, to import successfully make sure that **Visits** exists as a Component or replace it with Components that exists in your project.

Any Key/id used must exist in your project also.

Make sure you are using the Test Priority values that are defined in your project.

As we can see in the contents of the below file, we are using the three first lines to create two Preconditions and one Test Set.

In the reminding lines we are creating two manual test cases that are part of the Test Set created above (with Issue Id 3) and that are linked to the two Preconditions also created here (with Issue Id 1 and 2).

import_two_manual_tests_with_project_precond_testset.csv

```
Issue Id;Issue key;Issue type;Test type;Test Summary;Test Priority;
Component;Action;Data;Result;Project key;Test Set;Precondition;
Precondition type
1;;precondition;;Must access application page in logout state;Highest;
Visits;;;EWB;;;Manual
2;;precondition;;Must clear cookies;Highest;Visits;;;EWB;;;Manual
3;;testset;;Test set for Login;High;Visits;;;EWB;;;
4;;test;Manual;Test 1 for user story WEB-1;High;Visits;"Go to
login page";;;EWB;3;"1;2";
4;;;Manual;;;Visits;Enter username;peter;;;;;
4;;;Manual;;;Visits;Enter password;pwrld123;;;;;
4;;;Manual;;;Visits;Click login button;;User successfully logged in;;;
5;;test;Manual;Test 2 for user story WEB-1;High;Visits;"Go to
login page";;;EWB;3;"1;2";
5;;;Manual;;;Visits;Enter username;peter;;;;;
5;;;Manual;;;Visits;Enter password;abc123;;;;;
5;;;Manual;;;Visits;Click login button;;User logged in fails with
message;;;
```

Note that if you want to insert a multiline value you should use quotes like we did in the Action of the first manual test case:

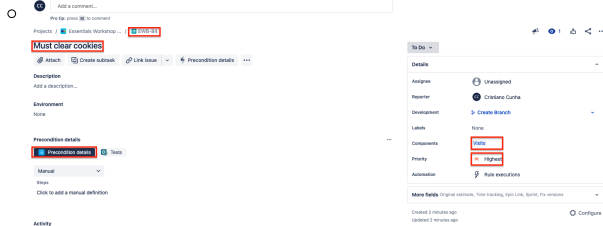
```
1;;Manual;Test 1 for user story WEB-1;High;Visits;"Go to
login page";;;EWB
```

If you want to refer multiple values in field, like we did when we are linking the test cases with the Preconditions created in this import, you should quote them and use a separator of your choice (usually ; or ,), as we have used in the above file: "1;2".

```
5;;test;Manual;Test 2 for user story WEB-1;High;Visits;"Go to
login page";;;EWB;3;"1;2";
```

After importing the CSV file we will have:

- Two new Preconditions



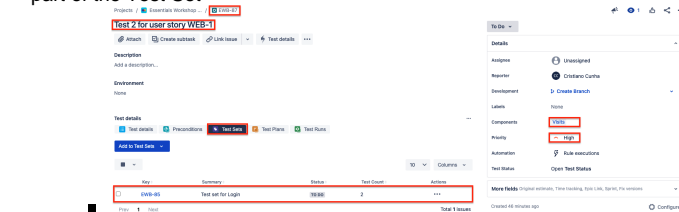
- One new Test Set



- Two new Tests linked to the Preconditions



- part of the Test Set



Import two manual test cases with Precondition, Test Set and Test Repository

In this example we will take the previous one and add the possibility to associate your tests to a specific folder in the Test Repository. In our case we are creating the folder/subfolder in the Test Repo and adding the tests to them.



The next examples are using Components existent in our project, to import successfully make sure that `visits` exists as a Component or replace it with Components that exists in your project.

Any Key/id used must exist in your project also.

Make sure your are using the Test Priority values that are defined in your project.

Looking at the CSV file to execute just that we can see a new field called *Test Repo* (bare in mind that you can use the description that you like here), this field will be used to create the Test Repo organization or, if it already exists, to associate the Tests to it. In this example we are associating these Tests to the folder *Main/login*.

import_two_manual_tests_with_project_precond_testset_testrepo.csv

```
Issue Id;Issue key;Issue type;Test type;Test Summary;Test Priority;
Component;Action;Data;Result;Project key;Test Set;Precondition;
Precondition type;Test Repo
1;;precondition;;Must access application page in logout state;Highest;
Visits;;;EWB;;;Manual;
2;;precondition;;Must clear cookies;Highest;Visits;;;EWB;;;Manual;
3;;testset;;Test set for Login;High;Visits;;;EWB;;;
4;;test;Manual;Test 1 for user story WEB-1;High;Visits;"Go to
login page";;EWB;3;"1;2";;Main/login
4;;;Manual;;;Visits;Enter username;peter;;;;;
4;;;Manual;;;Visits;Enter password;pwr123;;;;;
4;;;Manual;;;Visits;Click login button;;User successfully logged in;;;;;
5;;test;Manual;Test 2 for user story WEB-1;High;Visits;"Go to
login page";;EWB;3;"1;2";;Main/login
5;;;Manual;;;Visits;Enter username;peter;;;;;
5;;;Manual;;;Visits;Enter password;abc123;;;;;
5;;;Manual;;;Visits;Click login button;;User logged in fails with
message;;;;;
```

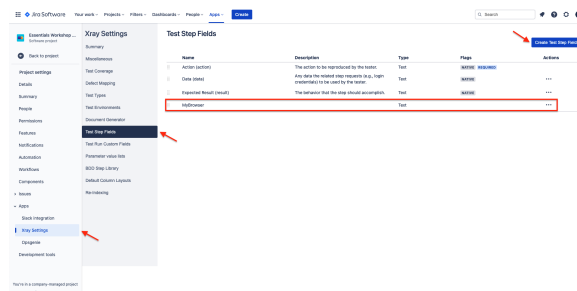
This will create two Preconditions, one Test Set and two manual Tests that will be present in the Test Repository under *Main/login* as we can see:



Import two manual test cases with test step custom field

In some cases we have the need to create custom field to support a specific behaviour in the test case steps and once we have it we need to be able to import test cases with information added to that field.

To exemplify this I have created a new custom test case Step called "MyBrowser" as we can see.



The next examples are using Components existent in our project, to import successfully make sure that **visits** exists as a Component or replace it with Components that exists in your project.

Any Key/id used must exist in your project also.

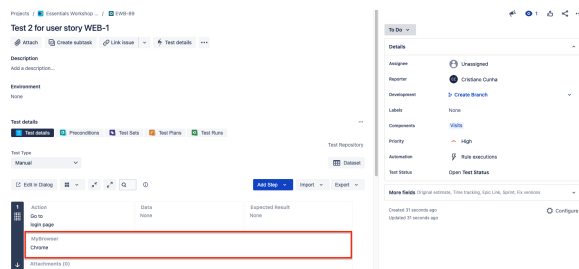
Make sure your are using the Test Priority values that are defined in your project.

I will be using the following CSV file to create two manual test cases and add relevant information (in my case *Firefox* or *Chrome*) to the custom filed in the test step.

import_two_manual_tests_project_customfield.csv

```
Issue Id;Issue key;Test type;Test Summary;Test Priority;Component;Action;Data;Result;Project key;My Browser
1;;Manual;Test 1 for user story WEB-1;High;Visits;"Go to login page";;;EWB;Firefox
1;;Manual;;;Visits;Enter username;peter;;;
1;;Manual;;;Visits;Enter password;pwrld23;;;
1;;Manual;;;Visits;Click login button;;User successfully logged in;;
2;;Manual;Test 2 for user story WEB-1;High;Visits;"Go to login page";;;EWB;Chrome
2;;Manual;;;Visits;Enter username;peter;;;
2;;Manual;;;Visits;Enter password;abc123;;;
2;;Manual;;;Visits;Click login button;;User logged in fails with message;;
```

Once imported we can see the new information appear in the Test Steps (on both Tests):



Import two manual test cases with custom field and labels

Building on top of the previous case, we are going to import two manual test cases that have a custom field and labels, to do that we are adding a field named *Labels* where we will define the labels we want to associate to the test cases.



The next examples are using Components existent in our project, to import successfully make sure that **visits** exists as a Component or replace it with Components that exists in your project.

Any Key/id used must exist in your project also.

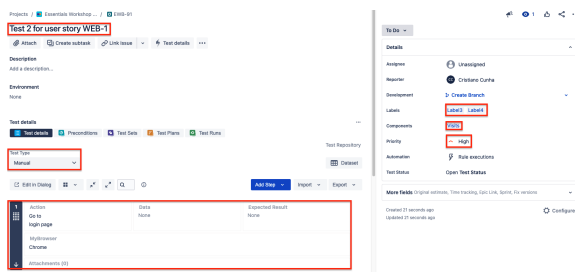
Make sure you are using the Test Priority values that are defined in your project.

The CSV file that we are using to demonstrate this is the next one.

import_two_manual_tests_project_customfield_Labels.csv

```
Issue Id;Issue key;Test type;Test Summary;Test Priority;Component;Action;Data;Result;Project key;My Browser;Labels
1;;Manual;Test 1 for user story WEB-1;High;Visits;"Go to login page";;;EWB;Firefox;"Label1;Label2"
1;;Manual;;;Visits;Enter username;peter;;;
1;;Manual;;;Visits;Enter password;pwrld23;;;
1;;Manual;;;Visits;Click login button;;User successfully logged in;;
2;;Manual;Test 2 for user story WEB-1;High;Visits;"Go to login page";;;EWB;Chrome;"Label3;Label4"
2;;Manual;;;Visits;Enter username;peter;;;
2;;Manual;;;Visits;Enter password;abc123;;;
2;;Manual;;;Visits;Click login button;;User logged in fails with message;;
```

Once we import these test cases we will see in Jira that they have the labels we have inserted in the CSV.



- If you want to use more than one label you must quote them like we can see in our example: "Label1;Label2"

Import gherkin and generic test cases

In this first example we exemplify how to import three test cases:

- One manual test case with a proper summary, priority, component association and where we define the manual steps (action, data, result).
- One generic test case with a proper summary, priority, component association and an unstructured definition.
- One cucumber test case with a proper summary, priority, component association and the gherkin definition



The next examples are using Components existent in our project, to import successfully make sure that **visits** exists as a Component or replace it with Components that exists in your project.

Any Key/id used must exist in your project also.

Make sure you are using the Test Priority that is defined in your project.

The CSV file used is the one below:

import_manual_gherkin_automated_tests.csv

```
Issue Id;Issue key;Test type;Test Summary;Test Priority;Component;Action;
Data;Result;Unstructured definition;Gherkin definition
1;;Manual;Test 1 for user story WEB-1;High;Visits;Go to login page;;;
1;;Manual;;;Visits;Enter username;peter;;;
1;;Manual;;;Visits;Enter password;pwrld23;;;
1;;Manual;;;Visits;Click login button;;User successfully logged in;;
2;;Generic;Test 2 for user story WEB-1;Low;Visits;;;FailedLogin;
3;;Cucumber;Addition validations;Low;Visits;;;;"Given I have entered 1
into the calculator
And I have entered 2 into the calculator
When I press Add
Then the result should be 3 on the screen"
```

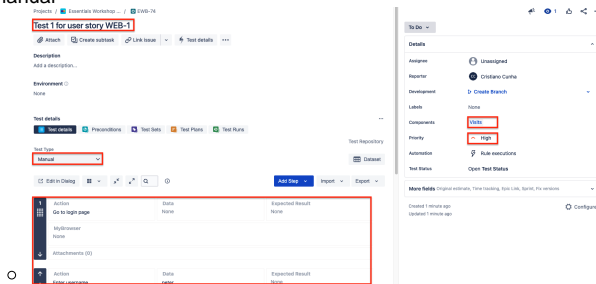
Notice that *Component* and *Priority* fields must match what you have defined in your project otherwise the import will fail.

The test type field have a limited possibility of values, they are: Manual, Generic and Cucumber. These will define the test type that will be created, if manual it expected to have test steps defined, in the case of Generic we must insert an unstructured definition and if it a Cucumber test type we need to insert the Gherkin configuration.

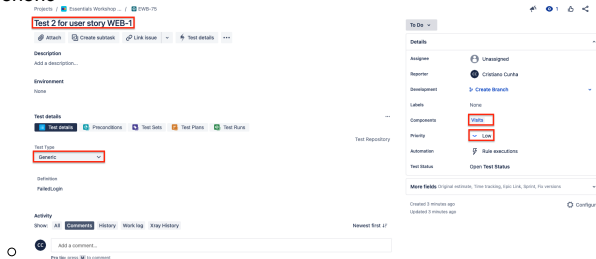
In the Cucumber test we have quoted the gherkin definition so that we can have multi-line support. This way the Gherkin definition will be more readable in Xray.

Once imported we will have three new test cases available:

- one manual



- one generic



- one cucumber test



With links to requirement

In this example we are using the same base as the previous importation but adding a link to the requirement during the importation, usually when we are creating tests we want to associate them to the requirements they are covering, this example is covering exactly that situation.



The next examples are using Components existent in our project, to import successfully make sure that **visits** exists as a Component or replace it with Components that exists in your project.

Any Key/id used must exist in your project also.

Make sure your are using the Test Priority that is defined in your project.

What differs from the previous is that we have added a **Links** field where we are going to insert the issue keys of the requirements that we want to be associated with the test we are importing.

import_manual_gherkin_automated_tests_links.csv

```
Issue Id;Issue key;Test type;Test Summary;Test Priority;Component;Action;Data;Result;Unstructured definition;Gherkin definition;Links
1;;Manual;Test 1 for user story WEB-1;High;Visits;Go to login page;;;EWB-39
1;;Manual;;;Visits;Enter username;peter;;;
1;;Manual;;;Visits;Enter password;pwrld23;;;
1;;Manual;;;Visits;Click login button;;User successfully logged in;;
2;;Generic;Test 2 for user story WEB-1;Low;Visits;;;FailedLogin;;EWB-39
3;;Cucumber;Addition validations;Low;Visits;;;;"Given I have entered 1
into the calculator
And I have entered 2 into the calculator
When I press Add
Then the result should be 3 on the screen";EWB-39
```

Once imported three tests will be created, all linked to the User Story EWB-39, in this case:

- one manual

The screenshot shows the 'Test 1 for user story WEB-1' manual test case. The 'Test details' pane on the left shows the test type as 'Manual'. The 'Test steps' pane shows a single step: 'Go to login page'. The 'Details' pane on the right shows the test is assigned to 'Unassigned', has a priority of 'High', and is linked to the user story 'EWB-39'.

- one generic

The screenshot shows the 'Test 2 for user story WEB-1' generic test case. The 'Test details' pane on the left shows the test type as 'Generic'. The 'Test steps' pane shows a single step: 'FailedLogin'. The 'Details' pane on the right shows the test is assigned to 'Unassigned', has a priority of 'Low', and is linked to the user story 'EWB-39'.

- one Cucumber

The screenshot shows the 'Addition validations' Cucumber test case. The 'Test details' pane on the left shows the test type as 'Cucumber'. The 'Test steps' pane shows three steps: 'Given I have entered 1 into the calculator', 'And I have entered 2 into the calculator', and 'When I press Add'. The 'Details' pane on the right shows the test is assigned to 'Unassigned', has a priority of 'Low', and is linked to the user story 'EWB-39'.

Update two manual test cases

If your intent is not to create new test cases but to update a test case you must refer to the test case Issue Key and when importing Test Case Importer will update the test case.



The next examples are using Components existent in our project, to import successfully make sure that **Visits** exists as a Component or replace it with Components that exists in your project.

Any Key/id used must exist in your project also.

Make sure you are using the Test Priority that is defined in your project.

To exemplify I have created the following CSV file

update_two_manual_tests.csv

```
Issue Id;Issue key;Test type;Test Summary;Test Priority;Component;Action;Data;Result
1;EWB-59;Manual;Test 1 for user story WEB-1 Changed;High;"Visits;Pets";Go to login page;;
1;;Manual;;;Visits;Enter username;peter;
1;;Manual;;;Visits;Enter password;pwrcl23;
1;;Manual;;;Visits;Click login button;;User successfully logged in with new message
2;EWB-60;Manual;Test 2 for user story WEB-1 Changed;High;Visits;Go to login page;;
2;;Manual;;;Visits;Enter username;peter;
2;;Manual;;;Visits;Enter password;abcl23;
2;;Manual;;;Visits;Click login button;;User logged in fails with another message
```

With this import we will update those two Tests (*EWB-59 and EWB-60*).

Tips

- Use the CSV file provided to import the examples and the Json configuration associated to it in order to have the setup and mapping correct.
- Remind to replace the Project Keys, test case Keys, Custom Fields and Requirements for some existing in your Jira instance.