

Importing TestLink test cases using Test Case Importer



What you'll learn

Features and Limitations

How it works

Prerequisites

Examples

- Different ways to export test cases from TestLink
- Use a script to convert the exported XML test cases to an Xray compatible CSV file
- Import different test cases (with different fields and options)
- Validate in Xray that the test cases are available
- Export one test suite (with 3 test cases) from TestLink and Import into Xray
- Export one test project from TestLink and import into Xray

Tips

Source-code for this tutorial

- All examples are available in [GitHub](#)

Overview

This tutorial explains how to export test cases from TestLink and import them into Xray.

TestLink is an open-source test management tool running in PHP. In this tutorial, you will learn how to export the test cases from TestLink in an XML format, how to use a script (available in [GitHub](#)) to convert the XML file into a compatible CSV so that it can be imported into Xray using the Test Case Importer.

In this tutorial, we will provide different examples of exporting and importing test cases from TestLink into Xray.

If you need help finding or using the tool, please check more information on the [Test Case Importer](#) page.

Features and Limitations

Below you can find a list of the support features and current limitations.

Migration from TestLink is limited by how much information we export from the tool and how many similarities we can find with Xray.

Most of the test case information will be migrated seamlessly, but please check the following table for more detail.

Supported Features	Unsupported
<ul style="list-style-type: none">• Test and Test steps• Preconditions• Test Suites	<ul style="list-style-type: none">• Test case attachments• Test Plans• Test Executions• Defects and Requirements

How it works

TestLink allows users to create different test projects (and connect the tool to bug-tracking or requirement management tools). The user can specify and join tests on those test projects into a test suite. Each test project can have multiple test suites.

TestLink also has a Test Plan entity available you can only execute tests if they are organized into test plans. One test plan must be associated with one build, so you must have created the builds before creating the test plans.

In TestLink, test execution is made up of both builds and test cases. If the build is not created for a project, you won't be able to perform test execution.

TestLink uses different entities to describe tests and the flow, so we have synthesized into the next table the existing mappings between TestLink and Xray.

TestLink	Xray	Comments
Test Case	Test	TestLink does not support data sets or Gherkin definition, only manual /automated test steps. All tests steps are imported unless they are empty.
Precondition (Field in Test Case entity)	Precondition (independent entity)	In TestLink the precondition is one field in the test case. In Xray a Precondition is a different entity and one Test Case can be linked to several Preconditions. The precondition field from TestLink is converted into a Precondition issue in Xray and linked back to the Test.
Test Suite	Test Set	The test suite of TestLink is mapped into a Test Set. The Test Set is added inline (before the Test) and will be created when uploading the CSV file. All Tests part of that test case

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 TestLink instance to export test cases in XML format (We have used version 1.9.20 of TestLink)
- Access and usage of scripts to convert TestLink XML test cases to CSV Xray import files
- 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 show how we obtained the XML file from TestLink, made the files available in GitHub, how to use the scripts to convert the XML files into compatible CSV files, and finally, how to import the CSV files into Xray.

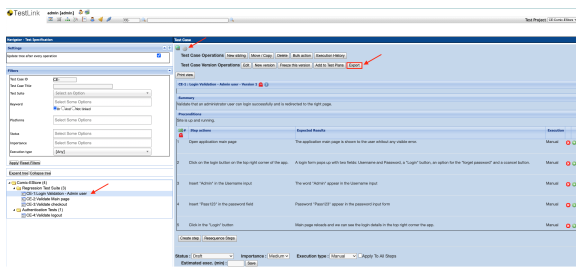
We will showcase different possibilities when exporting test cases from TestLink, using a combination of fields and possibilities that should cover most usages.

All of the examples in this tutorial have a XML file exported from TestLink, 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.

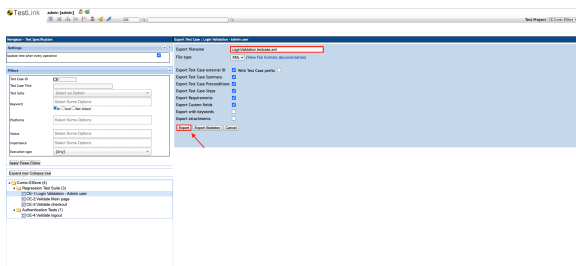
Export one test case from TestLink and Import into Xray

In TestLink you can export test cases only, in this example we are focusing on that option.

To extract one test case from TestLink we are using the *Export* option in the Test Specification left menu entry and selecting one test case.



We provide a name for the XML field that will be exported and click "Export".



The file generated by TestLink has the following content:

LoginValidation.testcase.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<testcases>
<testcase internalid="4" name="Login Validation - Admin user">
  <node_order><![CDATA[1000]]></node_order>
  <externalid><![CDATA[1]]></externalid>
  <version><![CDATA[1]]></version>
  <summary><![CDATA[<p>Validate that an administrator user can login successfully and is redirected to the right page.</p>]]></summary>
  <preconditions><![CDATA[<p>Site is up and running.</p>]]></preconditions>
  <execution_type><![CDATA[1]]></execution_type>
  <importance><![CDATA[2]]></importance>
  <estimated_exec_duration></estimated_exec_duration>
  <status>1</status>
  <is_open>1</is_open>
  <active>1</active>

  <steps>
<step>
    <step_number><![CDATA[1]]></step_number>
    <actions><![CDATA[<p>Open application main page</p>]]></actions>
    <expectedresults><![CDATA[<p>The application main page is shown to the user whitout any visible error.</p>]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
  </step>

<step>
    <step_number><![CDATA[2]]></step_number>
    <actions><![CDATA[<p>Click on the login button on the top right corner of the app.</p>]]></actions>
    <expectedresults><![CDATA[<p>A login form pops up with two fields: Username and Password, a "Login" button, an option for the "forget password" and a ccancel button.</p>]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
  </step>
</steps>
</testcase>
</testcases>
```

```

</step>

<step>
  <step_number><![CDATA[3]]></step_number>
  <actions><![CDATA[<p>Insert "Admin" in the Username input</p>
]]></actions>
  <expectedresults><![CDATA[<p>The word "Admin" appear in the
Username input</p>
]]></expectedresults>
  <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
  <step_number><![CDATA[4]]></step_number>
  <actions><![CDATA[<p>Insert "Pass123" in the password field</p>
]]></actions>
  <expectedresults><![CDATA[<p>Password "Pass123" appear in the
password input form</p>
]]></expectedresults>
  <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
  <step_number><![CDATA[5]]></step_number>
  <actions><![CDATA[<p>Click in the "Login" button</p>
]]></actions>
  <expectedresults><![CDATA[<p>Main page reloads and we can see the
login details in the top right corner the app.</p>
]]></expectedresults>
  <execution_type><![CDATA[1]]></execution_type>
</step>
</steps>
</testcase>
</testcases>

```

To import into Xray, we created one script to convert this XML file into a CSV-compatible file to be imported in Xray.

The script will parse the XML file and create a CSV file that can be imported into Xray. To execute the script, use the following command:

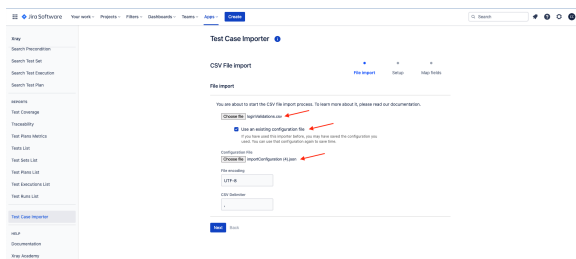
```
python3 testlink2Xray.py -i one_test_case/LoginValidation.testcase.xml -o
one_test_case/LoginValidation.testcase.csv
```

The output file has all the test steps and if you have used a precondition in TestLink, it will create a precondition in Xray and link it to the test.

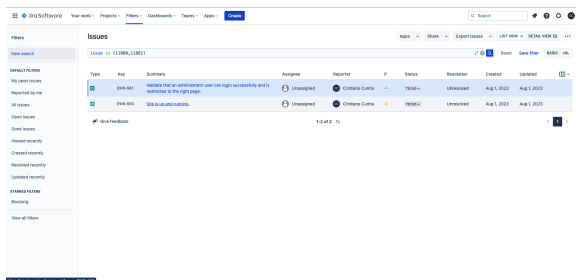
LoginValidation.testcase.csv

```
Issue ID,Issue Key,Test Type,Test Summary,Test Priority,Action,Data,Result,
Test Set,Precondition,Issue Type,Precondition Type
1,,,Site is up and running. ,,,,,,precondition,Manual
2,,Manual,Validate that an administrator user can login successfully and
is redirected to the right page. ,2,Open application main page ,,The
application main page is shown to the user whitout any visible error. ,,1,,
2,,Manual,,,Click on the login button on the top right corner of the app.
,,,"A login form pops up with two fields: Username and Password, a
"Login" button, an option for the "forget password" and a ccancel
button. " ,,,,
2,,Manual,,,,"Insert "Admin" in the Username input " ,,"The word "Admin"
appear in the Username input " ,,,,
2,,Manual,,,,"Insert "Pass123" in the password field " ,,"Password
"Pass123" appear in the password input form " ,,,,
2,,Manual,,,,"Click in the "Login" button " ,,"Main page reloads and we can
see the login details in the top right corner the app. ,,,,
```

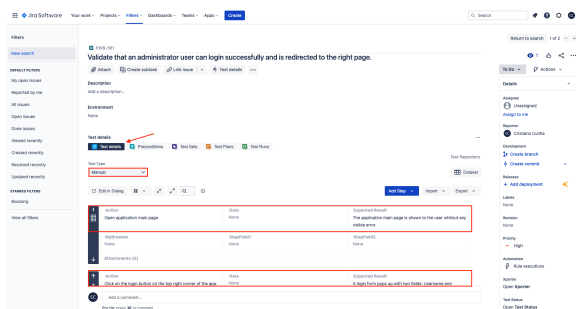
To import it into Xray we use Test Case Importer with the recently created CSV file and you can use the configuration file provided with the extension Json (*importConfiguration_onetestcase.json*).



Once imported we can see that it has created one Precondition and one Test with all the properties defined in the CSV.



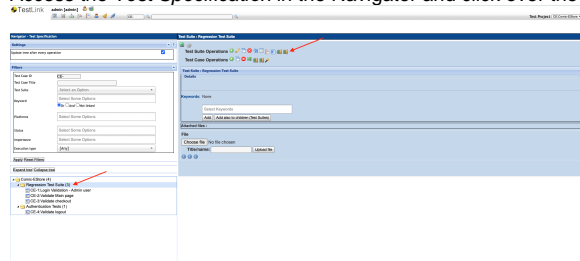
The Test created will have all the steps that were defined in TestLink and is linked to a Precondition (also created in the process).



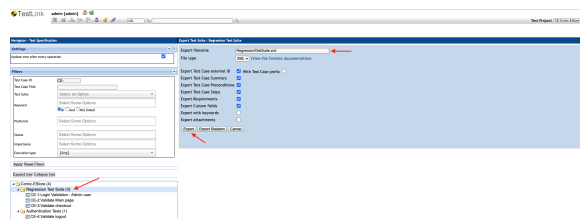
Export one test suite (with 3 test cases) from TestLink and Import into Xray

In TestLink we can export one test suite with all test cases. In this example, we export one test suite with 3 test cases.

Access the *Test Specification* in the Navigator and click over the test suite you want to export.



Once selected you can choose to export that test suite in an XML format by providing a name and pressing the *Export* button.



The file generated by TestLink has one test suite and 3 test cases with several steps as we can see.

RegressionTestSuite.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<testsuite id="3" name="Regression Test Suite" >
<node_order><![CDATA[1]]></node_order>
<details><![CDATA[]]></details>

<testcase internalid="4" name="Login Validation - Admin user">
  <node_order><![CDATA[1000]]></node_order>
  <externalid><![CDATA[1]]></externalid>
  <version><![CDATA[1]]></version>
  <summary><![CDATA[<p>Validate that an administrator user can login
successfully and is redirected to the right page.</p>
]]></summary>
  <preconditions><![CDATA[<p>Site is up and running.</p>
]]></preconditions>
  <execution_type><![CDATA[1]]></execution_type>
  <importance><![CDATA[2]]></importance>
  <estimated_exec_duration></estimated_exec_duration>
  <status>1</status>
  <is_open>1</is_open>
  <active>1</active>

  <steps>
  <step>
    <step_number><![CDATA[1]]></step_number>
    <actions><![CDATA[<p>Open application main page</p>
]]></actions>
    <expectedresults><![CDATA[<p>The application main page is shown to
the user whitout any visible error.</p>
]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
  </step>

  <step>
    <step_number><![CDATA[2]]></step_number>
```

```

        <actions><![CDATA[<p>Click on the login button on the top right
corner of the app.</p>
]]></actions>
        <expectedresults><![CDATA[<p>A login form pops up with two fields:
Username and Password, a "Login" button, an option for the "forget
password" and a ccancel button.</p>
]]></expectedresults>
        <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
    <step_number><![CDATA[3]]></step_number>
    <actions><![CDATA[<p>Insert "Admin" in the Username input</p>
]]></actions>
    <expectedresults><![CDATA[<p>The word "Admin" appear in the
Username input</p>
]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
    <step_number><![CDATA[4]]></step_number>
    <actions><![CDATA[<p>Insert "Pass123" in the password field</p>
]]></actions>
    <expectedresults><![CDATA[<p>Password "Pass123" appear in the
password input form</p>
]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
    <step_number><![CDATA[5]]></step_number>
    <actions><![CDATA[<p>Click in the "Login" button</p>
]]></actions>
    <expectedresults><![CDATA[<p>Main page reloads and we can see the
login details in the top right corner the app.</p>
]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
</step>
</steps>
</testcase>

<testcase internalid="11" name="Validate Main page">
    <node_order><![CDATA[1001]]></node_order>
    <externalid><![CDATA[2]]></externalid>
    <version><![CDATA[1]]></version>
    <summary><![CDATA[<p>Tests to validate the main page of the
application</p>
]]></summary>
    <preconditions><![CDATA[<p>The application must be executing and
available at the expected endpoint</p>
]]></preconditions>
    <execution_type><![CDATA[1]]></execution_type>
    <importance><![CDATA[2]]></importance>
    <estimated_exec_duration></estimated_exec_duration>
    <status>1</status>
    <is_open>1</is_open>
    <active>1</active>

    <steps>
    <step>
        <step_number><![CDATA[1]]></step_number>
        <actions><![CDATA[<p>Open main page of the application</p>
]]></actions>
        <expectedresults><![CDATA[<p>The main page of the applciation
appears</p>
]]></expectedresults>
        <execution_type><![CDATA[1]]></execution_type>
    </step>

    <step>
        <step_number><![CDATA[2]]></step_number>

```

```

        <actions><![CDATA[<p>Scroll down on the main page</p>
]]></actions>
        <expectedresults><![CDATA[<p>Products are loaded when you reach
the end of page without errors</p>
]]></expectedresults>
        <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
    <step_number><![CDATA[3]]></step_number>
    <actions><![CDATA[<p>Check the left menu filters</p>
]]></actions>
    <expectedresults><![CDATA[<p>All filters appear</p>
]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
    <step_number><![CDATA[4]]></step_number>
    <actions><![CDATA[]]></actions>
    <expectedresults><![CDATA[]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
</step>
</steps>
</testcase>

<testcase internalid="17" name="Validate checkout">
    <node_order><![CDATA[1002]]></node_order>
    <externalid><![CDATA[3]]></externalid>
    <version><![CDATA[1]]></version>
    <summary><![CDATA[<p>Test to validate the ability to checkout of
the app</p>
]]></summary>
    <preconditions><![CDATA[<p>The application is running</p>
<p>Login operation is working</p>
]]></preconditions>
        <execution_type><![CDATA[1]]></execution_type>
        <importance><![CDATA[2]]></importance>
        <estimated_exec_duration></estimated_exec_duration>
        <status>1</status>
        <is_open>1</is_open>
        <active>1</active>
<steps>
<step>
    <step_number><![CDATA[1]]></step_number>
    <actions><![CDATA[<p>Access the application</p>
]]></actions>
    <expectedresults><![CDATA[<p>Tha application main page is shown</p>
]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
    <step_number><![CDATA[2]]></step_number>
    <actions><![CDATA[<p>Login with user "John"</p>
]]></actions>
    <expectedresults><![CDATA[<p>User is logged in in the application<
/p>
]]></expectedresults>
    <execution_type><![CDATA[2]]></execution_type>
</step>

<step>
    <step_number><![CDATA[3]]></step_number>
    <actions><![CDATA[<p>Click on a product from the main page</p>
]]></actions>
    <expectedresults><![CDATA[<p>The details page of the product is
loaded</p>
]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>

```



```

</step>

<step>
  <step_number><![CDATA[4]]></step_number>
  <actions><![CDATA[<p>Add that product to your shopping bag</p>
]]></actions>
  <expectedresults><![CDATA[<p>The product appear in your shopping
bag</p>
]]></expectedresults>
  <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
  <step_number><![CDATA[5]]></step_number>
  <actions><![CDATA[<p>Proceed to check out</p>
]]></actions>
  <expectedresults><![CDATA[<p>Validate that the product is there
with the right quantities and price </p>
]]></expectedresults>
  <execution_type><![CDATA[1]]></execution_type>
</step>
</steps>
</testcase>
</testsuite>

```

To import it into Xray we use Test Case Importer with the recently created CSV file, and you can use the configuration file provided with the extension Json (*importConfiguration_onetestcase.json*).

Once imported, we can see that it has created one Precondition and one Test with all the properties defined in the CSV.

```

python3 testlink2Xray.py -i one_test_suite/RegressionTestSuite.xml -o
one_test_suite/RegressionTestSuite.csv

```

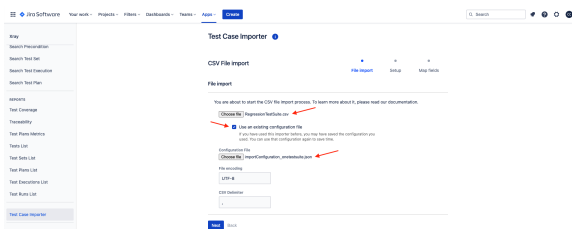
The output file has one Test Set (that corresponds to the test suite), all the test steps of the three tests and if you have used preconditions in TestLink, it will create a precondition in Xray and link it to the test.

LoginValidation.testcase.csv

```
Issue ID,Issue Key,Test Type,Test Summary,Test Priority,Action,Data,Result,
Test Set,Precondition,Issue Type,Precondition Type
1,,,Regression Test Suite,,,,,,,,testset,
2,,,Site is up and running. ,,,,,,precondition,Manual
3,,Manual,Validate that an administrator user can login successfully and
is redirected to the right page. ,2,Open application main page ,,The
application main page is shown to the user whitout any visible error. ,
1,2,,
3,,Manual,,,Click on the login button on the top right corner of the app.
,,,"A login form pops up with two fields: Username and Password, a
"Login" button, an option for the "forget password" and a ccancel
button. ",,,,
3,,Manual,,,,"Insert "Admin" in the Username input ",,"The word "Admin"
appear in the Username input ",,,,
3,,Manual,,,,"Insert "Pass123" in the password field ",,"Password
"Pass123" appear in the password input form ",,,,
3,,Manual,,,,"Click in the "Login" button ",,Main page reloads and we can
see the login details in the top right corner the app. ,,,,
4,,,The application must be executing and available at the expected
endpoint ,,,,,,precondition,Manual
5,,Manual,Tests to validate the main page of the application ,2,Open main
page of the application ,,The main page of the applciation appears ,1,4,,
5,,Manual,,,Scroll down on the main page ,,Products are loaded when you
reach the end of page without errors ,,,,
5,,Manual,,,Check the left menu filters ,,All filters appear ,,,,
5,,Manual,,,,,,,,
6,,,The application is running Login operation is working ,,,,,,
precondition,Manual
7,,Manual,Test to validate the ability to checkout of the app ,2,Access
the application ,,Tha application main page is shown ,1,6,,
7,,Manual,,,,"Login with user "John" " ",,User is logged in in the
application ,,,,
7,,Manual,,,Click on a product from the main page ,,The details page of
the product is loaded ,,,,
7,,Manual,,,Add that product to your shopping bag ,,The product appear in
your shopping bag ,,,,
7,,Manual,,,Proceed to check out ,,Validate that the product is there with
the right quantities and price ,,,,

```

To import it into Xray we use Test Case Importer with the recently created CSV file and you can use the configuration file provided with the extension Json (*importConfiguration_onetestsuite.json*).



The result of the importation shows three new Tests, one Test Set and three Preconditions.

Each Test has the summary defined and the steps that were defined in TestLink.

Notice that a Precondition was added also and linked to the Test.

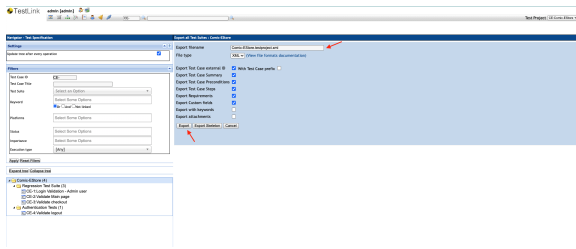
As we have exported from a test suite from TestLink we can see that in Xray the process has created a Test Set and linked it to the Tests.

Export one test project from TestLink and import into Xray

On TestLink, we can export one entire test project with all the test suites and test cases associated with it.

In this last example, we export one test project with two test suites and several test cases. On the same panel in TestLink we click the test project and export it.

[illegible]



The exported file holds the entire test project from TestLink in an XML format.

import_two_manual_tests_description_labels.csv

```
<?xml version="1.0" encoding="UTF-8"?>
<testsuite id="" name="" >
<node_order><![CDATA[]]></node_order>
<details><![CDATA[]]></details>
<testsuite id="3" name="Regression Test Suite" >
<node_order><![CDATA[1]]></node_order>
<details><![CDATA[]]></details>

<testcase internalid="4" name="Login Validation - Admin user">
  <node_order><![CDATA[1000]]></node_order>
  <externalid><![CDATA[1]]></externalid>
  <version><![CDATA[1]]></version>
  <summary><![CDATA[<p>Validate that an administrator user can login
successfully and is redirected to the right page.</p>
]]></summary>
  <preconditions><![CDATA[<p>Site is up and running.</p>
]]></preconditions>
  <execution_type><![CDATA[1]]></execution_type>
  <importance><![CDATA[2]]></importance>
  <estimated_exec_duration></estimated_exec_duration>
  <status>1</status>
  <is_open>1</is_open>
  <active>1</active>

  <steps>
  <step>
    <step_number><![CDATA[1]]></step_number>
    <actions><![CDATA[<p>Open application main page</p>
]]></actions>
    <expectedresults><![CDATA[<p>The application main page is shown to
the user whitout any visible error.</p>
]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
  </step>

  <step>
    <step_number><![CDATA[2]]></step_number>
    <actions><![CDATA[<p>Click on the login button on the top right
corner of the app.</p>
]]></actions>
    <expectedresults><![CDATA[<p>A login form pops up with two fields:
Username and Password, a "Login" button, an option for the "forget
password" and a ccancel button.</p>
]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
  </step>

  <step>
    <step_number><![CDATA[3]]></step_number>
    <actions><![CDATA[<p>Insert "Admin" in the Username input</p>
]]></actions>
    <expectedresults><![CDATA[<p>The word "Admin" appear in the
Username input</p>
]]></expectedresults>
```

```

        <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
    <step_number><![CDATA[4]]></step_number>
    <actions><![CDATA[<p>Insert "Pass123" in the password field</p>
]]></actions>
    <expectedresults><![CDATA[<p>Password "Pass123" appear in the
password input form</p>
]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
    <step_number><![CDATA[5]]></step_number>
    <actions><![CDATA[<p>Click in the "Login" button</p>
]]></actions>
    <expectedresults><![CDATA[<p>Main page reloads and we can see the
login details in the top right corner the app.</p>
]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
</step>
</steps>
</testcase>

<testcase internalid="11" name="Validate Main page">
    <node_order><![CDATA[1001]]></node_order>
    <externalid><![CDATA[2]]></externalid>
    <version><![CDATA[1]]></version>
    <summary><![CDATA[<p>Tests to validate the main page of the
application</p>
]]></summary>
    <preconditions><![CDATA[<p>The application must be executing and
available at the expected endpoint</p>
]]></preconditions>
    <execution_type><![CDATA[1]]></execution_type>
    <importance><![CDATA[2]]></importance>
    <estimated_exec_duration></estimated_exec_duration>
    <status>1</status>
    <is_open>1</is_open>
    <active>1</active>
</steps>
<step>
    <step_number><![CDATA[1]]></step_number>
    <actions><![CDATA[<p>Open main page of the application</p>
]]></actions>
    <expectedresults><![CDATA[<p>The main page of the applciation
appears</p>
]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
    <step_number><![CDATA[2]]></step_number>
    <actions><![CDATA[<p>Scroll down on the main page</p>
]]></actions>
    <expectedresults><![CDATA[<p>Products are loaded when you reach
the end of page without errors</p>
]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
    <step_number><![CDATA[3]]></step_number>
    <actions><![CDATA[<p>Check the left menu filters</p>
]]></actions>
    <expectedresults><![CDATA[<p>All filters appear</p>
]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
</step>

```

```

<step>
  <step_number><![CDATA[4]]></step_number>
  <actions><![CDATA[]]></actions>
  <expectedresults><![CDATA[]]></expectedresults>
  <execution_type><![CDATA[1]]></execution_type>
</step>
</steps>
</testcase>

<testcase internalid="17" name="Validate checkout">
  <node_order><![CDATA[1002]]></node_order>
  <externalid><![CDATA[3]]></externalid>
  <version><![CDATA[1]]></version>
  <summary><![CDATA[<p>Test to validate the ability to checkout of
the app</p>
]]></summary>
  <preconditions><![CDATA[<p>The application is running</p>
<p>Login operation is working</p>
]]></preconditions>
  <execution_type><![CDATA[1]]></execution_type>
  <importance><![CDATA[2]]></importance>
  <estimated_exec_duration></estimated_exec_duration>
  <status>1</status>
  <is_open>1</is_open>
  <active>1</active>
<steps>
<step>
  <step_number><![CDATA[1]]></step_number>
  <actions><![CDATA[<p>Access the application</p>
]]></actions>
  <expectedresults><![CDATA[<p>Tha application main page is shown</p>
]]></expectedresults>
  <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
  <step_number><![CDATA[2]]></step_number>
  <actions><![CDATA[<p>Login with user "John"</p>
]]></actions>
  <expectedresults><![CDATA[<p>User is logged in in the application<
/p>
]]></expectedresults>
  <execution_type><![CDATA[2]]></execution_type>
</step>

<step>
  <step_number><![CDATA[3]]></step_number>
  <actions><![CDATA[<p>Click on a product from the main page</p>
]]></actions>
  <expectedresults><![CDATA[<p>The details page of the product is
loaded</p>
]]></expectedresults>
  <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
  <step_number><![CDATA[4]]></step_number>
  <actions><![CDATA[<p>Add that product to your shopping bag</p>
]]></actions>
  <expectedresults><![CDATA[<p>The product appear in your shopping
bag</p>
]]></expectedresults>
  <execution_type><![CDATA[1]]></execution_type>
</step>

<step>
  <step_number><![CDATA[5]]></step_number>
  <actions><![CDATA[<p>Proceed to check out</p>
]]></actions>
  <expectedresults><![CDATA[<p>Validate that the product is there

```

```

with the right quantities and price </p>
]]></expectedresults>
    <execution_type><![CDATA[1]]></execution_type>
</step>
</steps>
</testcase>
</testsuite><testsuite id="24" name="Authentication Tests" >
<node_order><![CDATA[2]]></node_order>
<details><![CDATA[<p>Tests to validate authentication in the application<
/p>
]]></details>

<testcase internalid="25" name="Validate logout">
    <node_order><![CDATA[1000]]></node_order>
    <externalid><![CDATA[4]]></externalid>
    <version><![CDATA[1]]></version>
    <summary><![CDATA[<p>Validate that the logout is working</p>
]]></summary>
    <preconditions><![CDATA[<p>Application is running</p>
<p>User is in the logged in state</p>
]]></preconditions>
    <execution_type><![CDATA[1]]></execution_type>
    <importance><![CDATA[2]]></importance>
    <estimated_exec_duration></estimated_exec_duration>
    <status>1</status>
    <is_open>1</is_open>
    <active>1</active>

</testcase>
</testsuite></testsuite>

```

We can see that the test project has two test suites, one with three test cases and another with one test case.

Next we use the scripts provided to convert this XML file into a CSV compatible file. The command used is below.

```

python3 testlink2Xray.py -i one_test_project/Comic-EStore.testproject.xml -
o one_test_project/Comic-EStore.testproject.csv

```

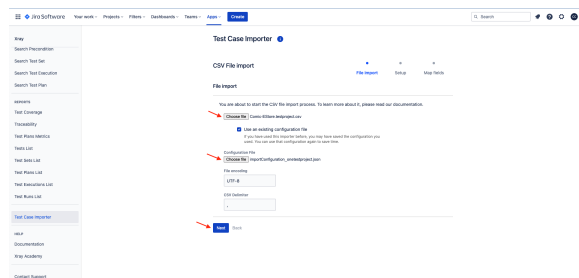
The output file has two Test Sets (corresponding to the test suites) that list the associated tests. All the test steps and if you have used preconditions in TestLink, it will create a precondition in Xray and link it to the test.

LoginValidation.testcase.csv

```
Issue ID,Issue Key,Test Type,Test Summary,Test Priority,Action,Data,Result,
Test Set,Precondition,Issue Type,Precondition Type
1,,,Regression Test Suite,,,,,,testset,
2,,,Site is up and running. ,,,,,,precondition,Manual
3,,Manual,Validate that an administrator user can login successfully and
is redirected to the right page. ,2,Open application main page ,,The
application main page is shown to the user whitout any visible error. ,
1,2,,
3,,Manual,,,Click on the login button on the top right corner of the app.
,,,"A login form pops up with two fields: Username and Password, a
"Login" button, an option for the "forget password" and a ccancel
button. ",,,,
3,,Manual,,,,"Insert "Admin" in the Username input ",,"The word "Admin"
appear in the Username input ",,,,
3,,Manual,,,,"Insert "Pass123" in the password field ",,"Password
"Pass123" appear in the password input form ",,,,
3,,Manual,,,,"Click in the "Login" button ",,Main page reloads and we can
see the login details in the top right corner the app. ,,,,
4,,,The application must be executing and available at the expected
endpoint ,,,,,,precondition,Manual
5,,Manual,Tests to validate the main page of the application ,2,Open main
page of the application ,,The main page of the appciation appears ,1,4,,
5,,Manual,,,Scroll down on the main page ,,Products are loaded when you
reach the end of page without errors ,,,,
5,,Manual,,,Check the left menu filters ,,All filters appear ,,,,
5,,Manual,,,,,,,
6,,,The application is running Login operation is working ,,,,,,
precondition,Manual
7,,Manual,Test to validate the ability to checkout of the app ,2,Access
the application ,,Tha application main page is shown ,1,6,,
7,,Manual,,,,"Login with user "John" " ",,User is logged in in the
application ,,,,
7,,Manual,,,Click on a product from the main page ,,The details page of
the product is loaded ,,,,
7,,Manual,,,Add that product to your shopping bag ,,The product appear in
your shopping bag ,,,,
7,,Manual,,,Proceed to check out ,,Validate that the product is there with
the right quantities and price ,,,,
8,,,Authentication Tests,,,,,,testset,
9,,,Application is running User is in the logged in state ,,,,,,
precondition,Manual
10,,Manual,Validate that the logout is working ,2,,,8,9,,
```

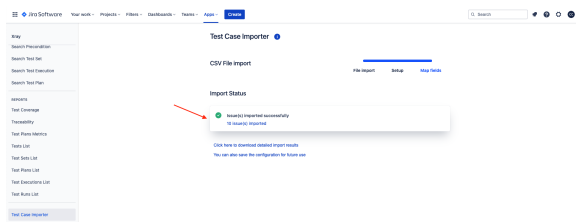
Once you have the CSV file you can proceed and import it into Xray using Test Case importer.

Import the CSV file using a configuration file that is also available in the code repository and check if all Tests were imported.



The output will have a link for you to see all issues that were created.

If some problem occurs, you will find more detail in this window also.



Clicking on that link we can see that 10 issues were created: four Tests, two Test Sets and four Preconditions.

Type	Key	Summary	Assignee	Reporter	Priority	Status	Resolution	Created	Updated
Test	TEST-001	Verify that the login is working	Unassigned	Unassigned	High	Unresolved	None	Aug 1, 2023	Aug 1, 2023
Test Set	TEST-002	Verify that the login is working	Unassigned	Unassigned	High	Unresolved	None	Aug 1, 2023	Aug 1, 2023
Test	TEST-003	Verify that the login is working	Unassigned	Unassigned	High	Unresolved	None	Aug 1, 2023	Aug 1, 2023
Test Set	TEST-004	Verify that the login is working	Unassigned	Unassigned	High	Unresolved	None	Aug 1, 2023	Aug 1, 2023
Test	TEST-005	Verify that the login is working	Unassigned	Unassigned	High	Unresolved	None	Aug 1, 2023	Aug 1, 2023
Test Set	TEST-006	Verify that the login is working	Unassigned	Unassigned	High	Unresolved	None	Aug 1, 2023	Aug 1, 2023
Test	TEST-007	Verify that the login is working	Unassigned	Unassigned	High	Unresolved	None	Aug 1, 2023	Aug 1, 2023
Test Set	TEST-008	Verify that the login is working	Unassigned	Unassigned	High	Unresolved	None	Aug 1, 2023	Aug 1, 2023
Test	TEST-009	Verify that the login is working	Unassigned	Unassigned	High	Unresolved	None	Aug 1, 2023	Aug 1, 2023
Test Set	TEST-010	Verify that the login is working	Unassigned	Unassigned	High	Unresolved	None	Aug 1, 2023	Aug 1, 2023

Tips

- Use the CSV file provided to import the examples and the Json configuration associated with it in order to have the setup and mapping correct.
- Make sure you are using the scripts available for the right version of Jira (Cloud or Server).