Working with Test Environments

Overview

The status of a Test (i.e., TestRunStatus) is generally calculated by looking at the Test’s last execution (more detail [here](#)). However, this does not work well if you execute the same Test in two different test environments (e.g., devices) and you want the two results to be consolidated.

Within a Test Execution, you may specify the Test Environment(s) where the tests will be executed in the respective attribute. A Test Environment is similar to a label, but Xray has special logic to deal with it.

If you use Test Environments, you may reuse the same Test for multiple test environments and create Test Executions for each one. If you don't use Test Environments and you want to track tests for multiple environments (e.g., devices), then the only way to do it is to create multiple tests, one per each test environment.

Let's say that you have executions for two Test Environments: “Android” and “iOS”. The test is considered as PASS only if the latest executions for Android and iOS are both PASS; otherwise, it will be FAIL.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Test Environment(s) of TE 1</th>
<th>Test Environment(s) of TE 2</th>
<th>Test run status in TE1</th>
<th>Test run status in TE2</th>
<th>Calculated value of “TestRunStatus” custom field</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Android</td>
<td>iOS</td>
<td>PASS</td>
<td>PASS</td>
<td>PASS</td>
</tr>
<tr>
<td>B</td>
<td>iOS</td>
<td>iOS</td>
<td>PASS</td>
<td>FAIL</td>
<td>FAIL</td>
</tr>
<tr>
<td>C</td>
<td>iOS</td>
<td>iOS</td>
<td>FAIL</td>
<td>PASS</td>
<td>PASS</td>
</tr>
</tbody>
</table>

Note: If you don’t use Test Environments, then the TestRunStatus is calculated as usual.

If you start using Test Environments in your Test Executions, then it’s not only your test status calculation that will change. All custom fields that depend on it (e.g., Requirement status, Test Sets status) will change. Consequently, the requirement coverage calculation and all associated charts/gadgets are also affected.

How it works

Let's say we have a Test being executed two times. We will start by creating Test Executions TE1 and TE2.

TE2 will be executed after TE1, so TE1 is considered older.

Below are example scenarios and shows how the TestRunStatus is calculated in each.

If you have a mix of Test Executions (i.e., with and without Test Environment as in scenario D), it will be treated the same way as scenarios A, B and C.
When creating a Test Execution, you must set the Test Environment in which the execution will be executed. You can use this field as a simple label: just add the environment or reuse a previously created one.

The Test Environments column is shown in your Test Runs table so you can distinguish each execution of the Test between the different environments.
The same Test has been executed in both Test Environments (a Test Execution per Test Environment).

The Requirement Status will show **NOK** because the Test has failed for one of the environments.
Tips

- Use Test Environments only if you want to run the same Test on different environments.
- If you do use it, then assign just one Test Environment per Test Execution.
- You may filter using Test Environments in your Test Executions panel to see how the executions are doing, per environment.

Learn more

For advanced Test Environment management capabilities, please check our Integration with Apwide Golive.