

# Integration with Microsoft Teams

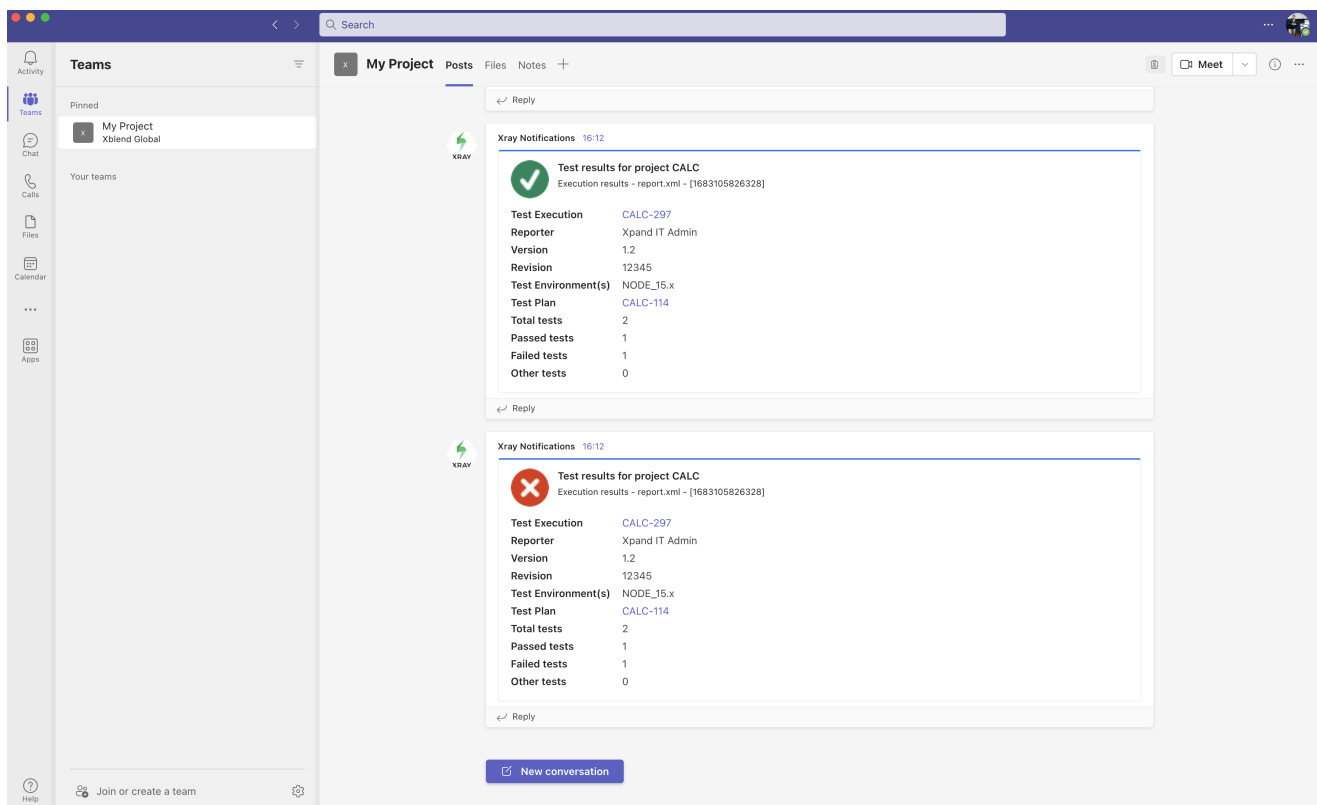
- [Overview](#)
- [Prerequisites](#)
- [Use cases](#)
  - [Share test results to a channel](#)
    - [Step-by-step implementation with Jira automation](#)
    - [Example of the output](#)
    - [Further ideas to try out](#)
- [Tips](#)
- [References](#)

## Overview

Microsoft Teams, or "Teams," is a popular communication platform many development and non-development teams adopt.

Teams use it daily to collaborate, make calls, share information, ask for help, and be notified of certain events to increase visibility of the status of the projects they're working on.

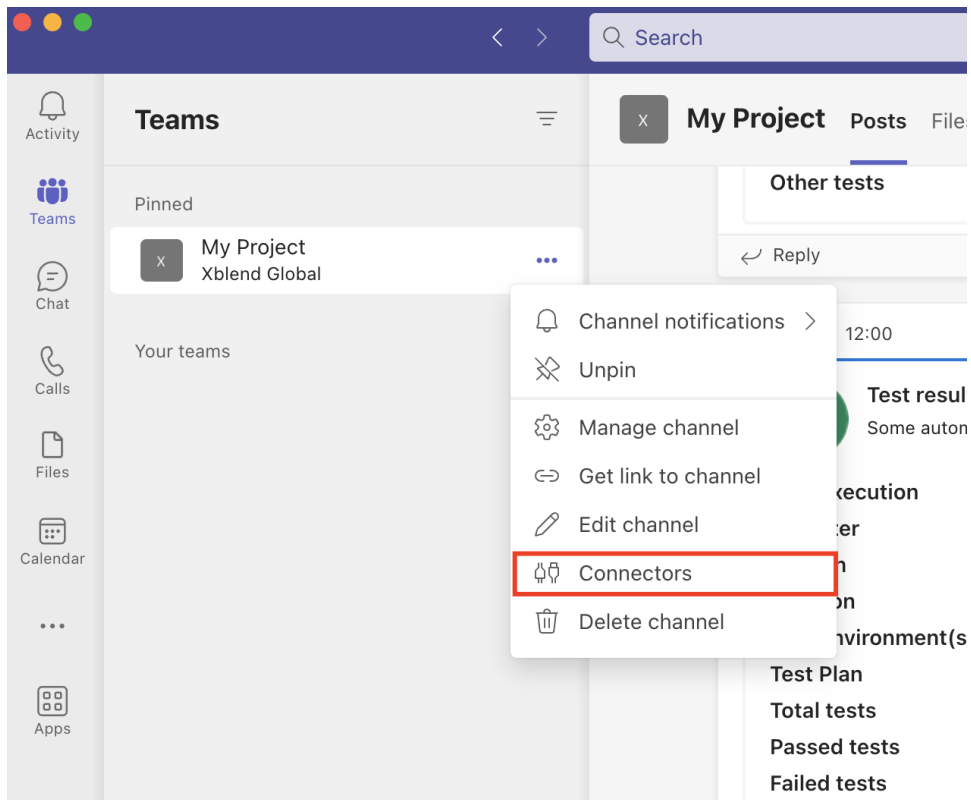
In this article, we'll see an everyday use case: sharing test results with the team in their collaboration tool (i.e., "MS Teams"). However, Xray and Jira's built-in automation capabilities can implement many more use cases.



## Prerequisites

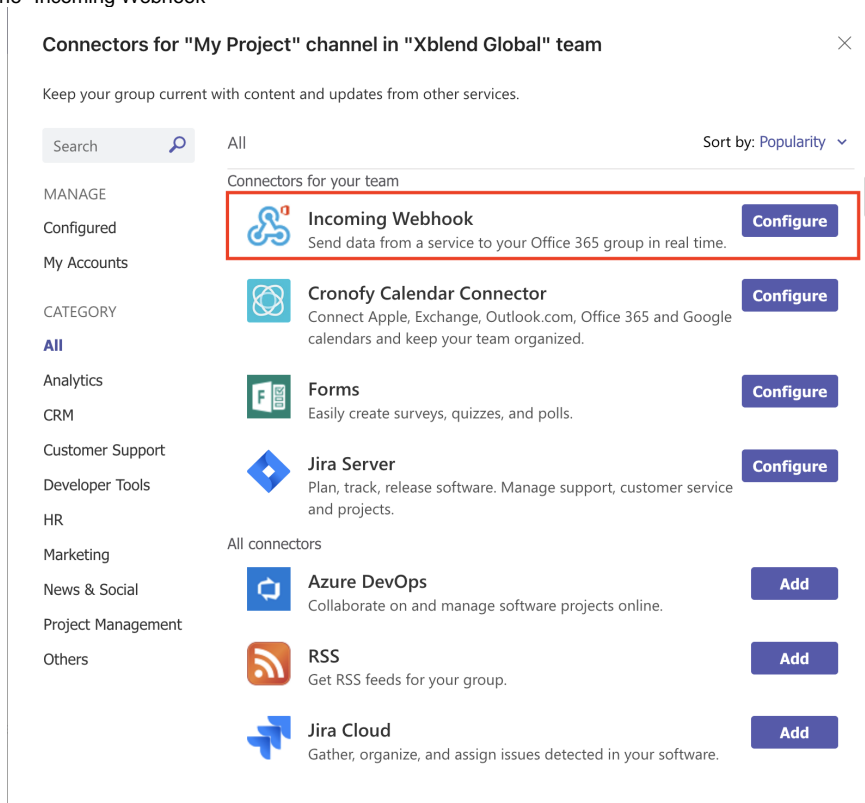
To integrate with Teams, we can set up an incoming webhook in a channel and use it to send notifications (i.e., "messages").

1. In Teams, select the channel and the "Connectors" option



a.


2. Select the "Incoming Webhook"



a.

3. Configure the "Incoming Webhook", giving it a name and uploading an image to be associated with it. Copy the unique URL and keep it safe.

Connectors for "My Project" channel in "Xblend Global" team

 Incoming Webhook
Send feedback


The Incoming Webhook connector enables external services to notify you about activities that you want to track. To use this connector, you'll need to create certain settings on the other service, which needs to support a webhook that's compatible with the [Office 365 connector format](#).

Fields marked with \* are mandatory


Enter a name for your IncomingWebhook connection. \*

Customize the image to associate with the data from this Incoming Webhook.

Upload Image



Copy the URL below to save it to the clipboard, then select Save. You'll need this URL when you go to the service that you want to send data to your group.



Url is up-to-date.

- a.
- b. Note: You can use the following image containing the Xray logo



i.

## Use cases

### Share test results to a channel

This common use-case is about sharing information about test results, usually from test automation, to a Teams channel so the team can be aware of testing progress and analyze the results if needed.



At high-level, we will:

1. obtain the Test Execution details, including recorded test results, using information stored as JSON on the "Test Execution Status" custom field of the corresponding Test Execution issue;
2. build an HTTP request to send a notification with the test results summary by invoking a webhook on MS Teams

## Step-by-step implementation with Jira automation

1. In Jira's project settings, under Automation, create the Jira automation rule
  - a. define the trigger; we can have a manual trigger that will provide an action from the issue screen or a trigger based on creating a specific issue type. In this case, it would probably make the most sense to have a trigger based on the transition to some status (e.g., "Done"). We can also restrict this to the issue types "Test Execution" and "Sub-Test Execution" using an IF condition.

**Project settings**

Summary  
Details  
Audit log  
Re-index project  
Delete project

Issue types  
- Bug  
- Epic  
- Improvement  
- New Feature

**Automation** Return to list

Automation rules allow you to automate repetitive tasks based on criteria that you set. Here you can manage existing rules and create new ones. [Learn more about automation](#)

**send test results to Teams / Slack** DRAFT

**Issue transitioned**

Rule is run when an issue is transitioned through its workflow.

From status  
Leave blank to match all statuses...

To status  
Done

**When: Issue transitioned**  
TO  
Done

**Issue fields condition**

Check whether an issue's field meets a certain criteria

Field\*  
Issue Type

Condition\*  
is one of

Value Field  
Test Execution Sub Test Execution

**Automation** Return to list

Automation rules allow you to automate repetitive tasks based on criteria that you set. Here you can manage existing rules and create new ones. [Learn more about automation](#)

**send test results to Teams / Slack** DRAFT

**When: Issue transitioned**  
TO  
Done

**If: Issue Type is one of**  
Test Execution, Sub Test Execution

**Please note**

Xray provides an automatic mechanism to transition Test Execution issues if all of its tests have a final status (i.e., if all tests have reported results). This is useful whenever uploading test automation results from a CI/CD pipeline. Please read the available configuration options to enable this under the [Miscellaneous](#) tab in Xray global settings.

2. make a IF THEN ELSE block to adjust the text message and image of the notification, to tailor our notification for successful and unsuccessful testing case
  - a. We can compare the number of tests that passed vs the total number of tests on the Test Execution, to identify whether all test results were successful or not. We'll use information from the "Test Execution Status" custom field that is associated with the Test Execution issue; if you want to see what's within this field, you can use the "Add value to the audit log"

**Automation** Return to list

Automation rules allow you to automate repetitive tasks based on criteria that you set. Here you can manage existing rules and create new ones. [Learn more about automation](#)

**send test results to Teams / Slack** DRAFT

**Log action**

Add value to the audit Log

Log message\*  
Test Execution Status: {{Test Execution Status}}

**When: Issue transitioned**  
TO  
Done

**If: Issue Type is one of**  
Test Execution, Sub Test Execution

**Then: Add value to the audit Log**  
Test Execution Status: {{Test Execution Status}}

**Automation** Return to list

Automation rules allow you to automate repetitive tasks based on criteria that you set. Here you can manage existing rules and create new ones. [Learn more about automation](#)

**send test results to Teams / Slack** DRAFT

**When: Issue transitioned**  
TO  
Done

**If: Issue Type is one of**  
Test Execution, Sub Test Execution

**Then: Add value to the audit Log**  
Test Execution Status: {{Test Execution Status}}

Automation rules allow you to automate repetitive tasks based on criteria that you set. Here you can manage existing rules and create new ones. [Learn more about automation](#)

send test results to Teams / Slack DRAFT

- Rule details
- Audit log

When: Issue transitioned  
to  
Done

If: Issue Type is one of  
Test Execution, Sub Test Execution

Then: Add value to the audit Log  
Test Execution Status: {{Test Execution Status}}

And: Add value to the audit Log  
Success: {{Test Execution Status.count.eq(Test Execution Status.statuses.get(0).statusCount)}}

If: matches  
{{Test Execution Status.statuses.get(1).name}} {{Test Execution Status.statuses.get(1).statusCount}}  
does not equal {{Test Execution Status.count}}

If block

The If block executes the actions within that block when the all specified conditions matches. Otherwise, the following else blocks will be evaluated.

Advanced compare condition

Compares two values using smart values and regular expressions.

First value\*

\$.statuses.get(1).name}} {{Test Execution Status.statuses.get(1).statusCount}}

Condition

does not equal

Second value

{{Test Execution Status.count}}

+ Add conditions...

Cancel Save

ii.

iii. {{Test Execution Status.statuses.get(0).statusCount}}

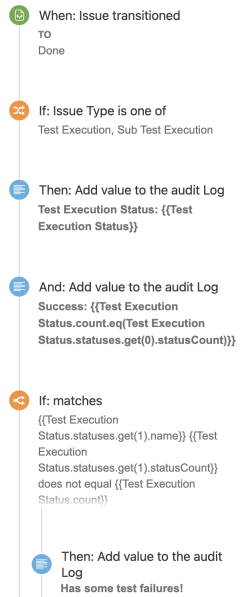
...

{{Test Execution Status.count}}

- Note: we could also do it by looking at the number of failed tests using the expression `{{Test Execution Status.statuses.get(1).statusCount}}`
- send the notification by using a "Send web request" to invoke the webhook on Teams (two requests, one for success and another if tests failed)
    - If not all tests passed, we can send the following content.

send test results to Teams / Slack DRAFT

- Rule details
- Audit log



i. **And: Send web request**

### Send web request

This action will send a HTTP request to the uri specified below:

Webhook URL\*  
`xbef7638141229ba94a92b594756c/ce34283c-537b-4732-bad3-9703a6ae99d2`

Request parameters must be url encoded, smart values should use: {{value.uriEncode}}.

Headers (optional)

Content-Type

Add

HTTP method  
POST

Webhook body  
Custom data

Wait for response  
☐ Delay execution of subsequent rule actions until we've received a response for this webhook

Custom data\*

```
{
  "@type": "MessageCard",
  "@context": "http://schema.org/extensions",
  "themeColor": "0076D7",
  "summary": "{{issue.summary}}",
  "sections": [
    {
      "activitySubtitle": "{{issue.summary}}",
      "activityTitle": "Test results for project {{project.name}}",
      "activityImage": "https://docs.getxray.app/s/e1tqew/8402/f0863dd17de361916f7914adff17e"
    }
  ]
}
```

> Validate your webhook configuration

Cancel Save

```

{
  "@type": "MessageCard",
  "@context": "http://schema.org/extensions",
  "themeColor": "0076D7",
  "summary": "{{issue.summary}}",
  "sections": [{
    "activitySubtitle": "{{issue.summary}}",
    "activityTitle": "Test results for project {{project.name}}",
    "activityImage": "https://docs.getxray.app/s/eltgew/8402/f0863dd17de361916f7914addff17e0432a0be98/_/images/icons/emoticons/error.png",
    "facts": [
      {
        "name": "Test Execution",
        "value": "[{{issue.key}}]({{issue.url}})"
      }, {
        "name": "Reporter",
        "value": "{{issue.reporter.displayName}}"
      }, {
        "name": "Version",
        "value": "{{issue.fixVersions.name}}"
      }, {
        "name": "Revision",
        "value": "{{issue.Revision}}"
      }, {
        "name": "Test Environment(s)",
        "value": "{{issue.Test Environments}}"
      }, {
        "name": "Test Plan",
        "value": "[{{issue.Test Plan}}]({{baseUrl}}/browse/{{issue.Test Plan}})"
      }, {
        "name": "Total tests",
        "value": "{{Test Execution Status.count}}"
      }, {
        "name": "Passed tests",
        "value": "{{Test Execution Status.statuses.get(0).statusCount}}"
      }, {
        "name": "Failed tests",
        "value": "{{Test Execution Status.statuses.get(1).statusCount}}"
      }, {
        "name": "Other tests",
        "value": "{{#=}}{{Test Execution Status.count}} - {{Test Execution Status.statuses.get(0).statusCount}} - {{Test Execution Status.statuses.get(1).statusCount}}{/{/}}"
      }
    ],
    "markdown": true
  }]
}

```

b. If all tests passed, within the "ELSE" section of the IF block, we can send the following content.

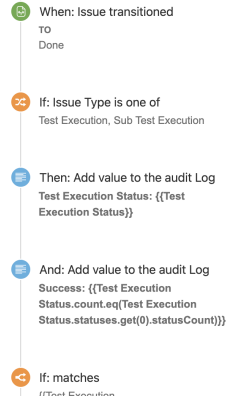
i.

## Automation

[Publish rule](#)[Return to list](#)

Automation rules allow you to automate repetitive tasks based on criteria that you set. Here you can manage existing rules and create new ones. [Learn more about automation](#)

### send test results to Teams / Slack

**DRAFT**[Rule details](#)[Audit log](#)

### Send web request

This action will send a HTTP request to the url specified below:

Webhook URL\*

<https://benjollydara.webhook.office.com/webhookb2/22b81e89-77bd-41c8-816>

Request parameters must be url encoded, smart values should use: {{value.uri[Encode]}}

Headers (optional)

Content-Type

[Add](#)

HTTP method

POST

Webhook body

Custom data

Wait for response

☐ Delay execution of subsequent rule actions until we've received a response for this webhook

Custom data\*

```
{
  "@type": "MessageCard",
  "@context": "http://schema.org/extensions",
  "themeColor": "0076D7",
  "summary": "{{issue.summary}}",
  "sections": [
    {
      "activitySubtitle": "{{issue.summary}}",
      "activityTitle": "Test results for project {{project.name}}",
      "activityImage":
        "https://docs.getxray.app/s/e1tqew/8402/0863dd17de361916f7914addff17e"
    }
  ]
}
```

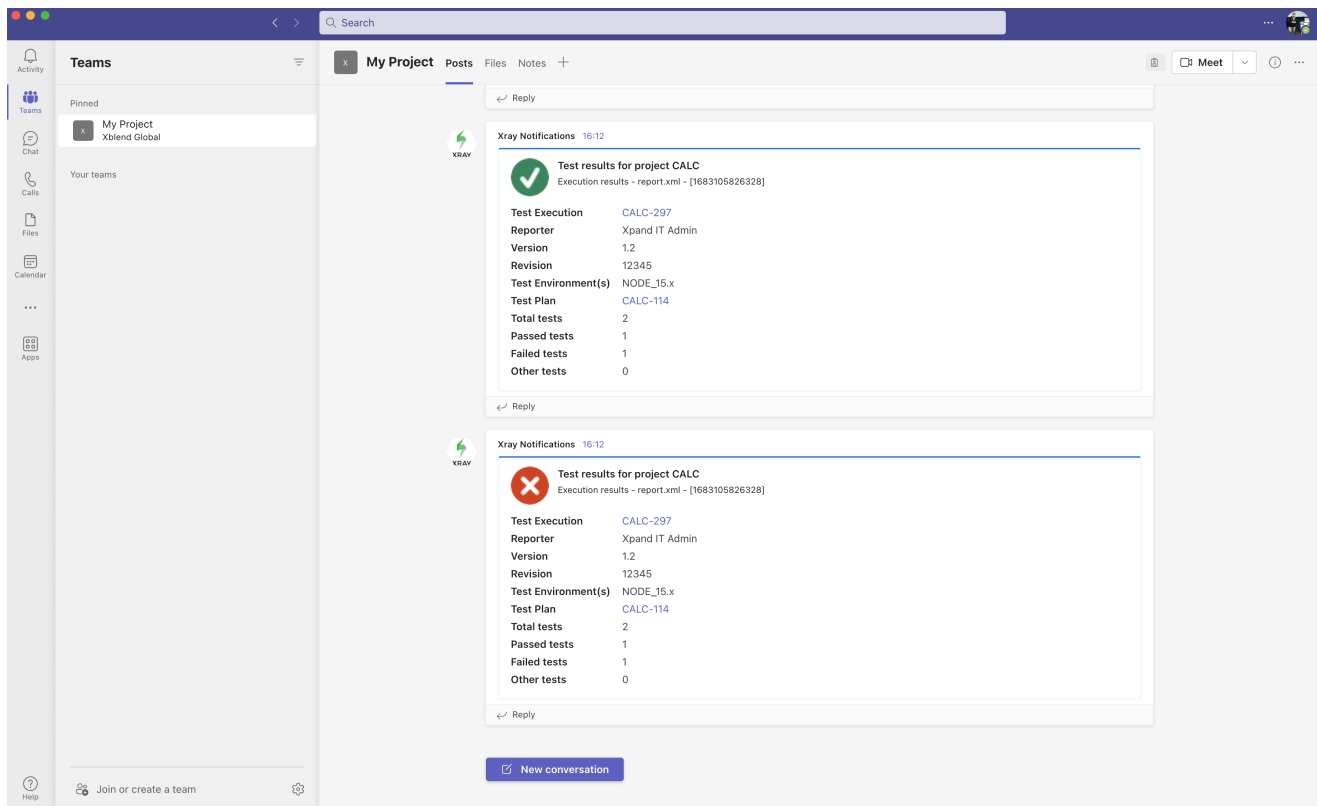


```

{
  "@type": "MessageCard",
  "@context": "http://schema.org/extensions",
  "themeColor": "0076D7",
  "summary": "{{issue.summary}}",
  "sections": [{
    "activitySubtitle": "{{issue.summary}}",
    "activityTitle": "Test results for project {{project.name}}",
    "activityImage": "https://docs.getxray.app/s/eltgew/8402/f0863dd17de361916f7914adddf17e0432a0be98/_/images/icons/emoticons/check.png",
    "facts": [
      {
        "name": "Test Execution",
        "value": "[{{issue.key}}]({{issue.url}})"
      }, {
        "name": "Reporter",
        "value": "{{issue.reporter.displayName}}"
      }, {
        "name": "Version",
        "value": "{{issue.fixVersions.name}}"
      }, {
        "name": "Revision",
        "value": "{{issue.Revision}}"
      }, {
        "name": "Test Environment(s)",
        "value": "{{issue.Test Environments}}"
      }, {
        "name": "Test Plan",
        "value": "[{{issue.Test Plan}}]({{baseUrl}}/browse/{{issue.Test Plan}})"
      }, {
        "name": "Total tests",
        "value": "{{Test Execution Status.count}}"
      }, {
        "name": "Passed tests",
        "value": "{{Test Execution Status.statuses.get(1).statusCount}}"
      }, {
        "name": "Failed tests",
        "value": "{{Test Execution Status.statuses.get(0).statusCount}}"
      }, {
        "name": "Other tests",
        "value": "{{#=}}{{Test Execution Status.count}} - {{Test Execution Status.statuses.get(0).statusCount}} - {{Test Execution Status.statuses.get(1).statusCount}}{}}"
      }, {
        "name": "Other tests",
        "value": "{{#=}}{{Test Execution Status.count}} - {{Test Execution Status.statuses.get(0).statusCount}} - {{Test Execution Status.statuses.get(1).statusCount}}{}}"
      }
    ],
    "markdown": true
  }]
}

```

## Example of the output



## Further ideas to try out

- send the notification only if there are test failures
  - send the notification only if there is a specific label or custom field on the Test Execution
  - add comments and/or logs of the failed tests to the notification

## Tips

To get information about test results using the "Test Execution Status" custom field, the scripts above use the "Test Execution Status" custom field that has an internal representation about the overall testing progress. Note that this internal representation may change in the future.

The table below presents some information you may obtain and how to perform it using Jira automation smart values/expressions. Please see the code snippets above to see how additional information.

Information to obtain	Jira Automation smart values	Notes
<b>Total Tests</b>	<code>{{Test Execution Status.count}}</code>	
<b>Number of tests passing (i.e., reported as "Pass")</b>	<code>{{Test Execution Status.statuses.get(0).statusCount}}</code>	<p><u>A more accurate expression would be:</u></p> <pre>{{#issue.Test Execution Status.statuses}}{{#if(equals(name,"PASS"))}}{{statusCount}}{{/}}{{/}}</pre>
<b>Number of failed tests (i.e., reported as "FAIL")</b>	<code>{{Test Execution Status.statuses.get(1).statusCount}}</code>	<p><u>A more accurate expression would be:</u></p> <pre>{{#issue.Test Execution Status.statuses}}{{#if(equals(name,"FAIL"))}}{{statusCount}}{{/}}{{/}}</pre>
<b>Test Environments</b>	<code>{{issue.Test Environments}}</code>	
<b>Linked Test Plan(s)</b>	<code>{{issue.Test Plan}}</code>	

# References

- Teams
  - [Teams: Types of cards](#)
    - [Create and send messages](#)
    - [Adaptive Cards: Samples and Templates](#)
  - [Jira Automation in Jira DC](#)
    - [Jira smart values - issues](#)
    - [Jira smart values - lists](#)
    - [Jira smart values - text fields](#)
    - [Jira smart values - users](#)
    - [Jira smart values - conditional logic](#)
    - [Jira smart values - JSON functions](#)