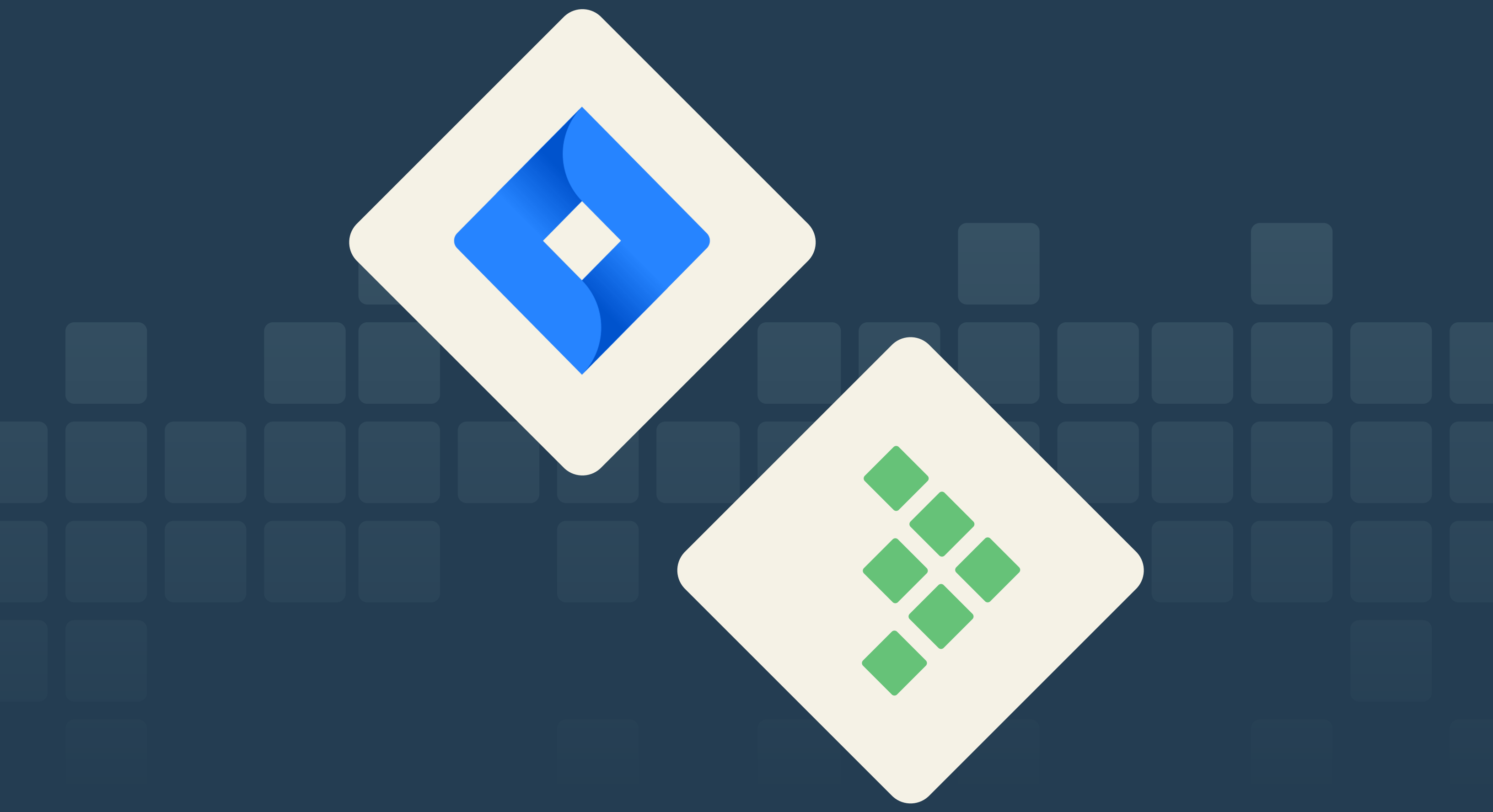


How to Align Development and QA with Jira and TestRail

Many software development teams use Jira to track requirements, user stories, and development status. QA has to figure out how to track their testing as well — and show test coverage and traceability against development work.

This guide shows you how you can build an efficient testing process, track coverage, and build comprehensive traceability between development and QA using Jira and TestRail.

Contents

Can Jira be used for test management?

Can Jira be used for test management?

Some QA teams use Jira itself to track their testing activities, and it’s easy to see why. Their developers are already on Jira, and using a single tool to manage work across teams reduces confusion, contradicting information, and duplication of efforts.

However, Jira was not designed for test management! Even with a Jira testing add-on, many teams that try to use Jira for test management struggle to organize their testing activities efficiently, get real-time visibility, and ensure all user stories are tested adequately. Managing traceability between requirements and testing, integrating with other testing tools, and scaling testing processes quickly overextends Jira — and can cause your testers to waste valuable time just trying to make Jira work.

Even though Jira isn’t the ideal solution itself, most teams also don’t wish to completely silo their testing and development teams into two separate platforms. **So, how do you solve the Jira problem?**

Luckily, there are many dedicated testing platform solutions — like TestRail! — that integrate with Jira. The TestRail integration for Jira provides full two-way visibility between platforms so that testing information is automatically updated in real-time. Linking between Jira issues and TestRail testing artifacts make traceability and coverage reporting as easy as pushing a button.

Configure the TestRail Integration for Jira

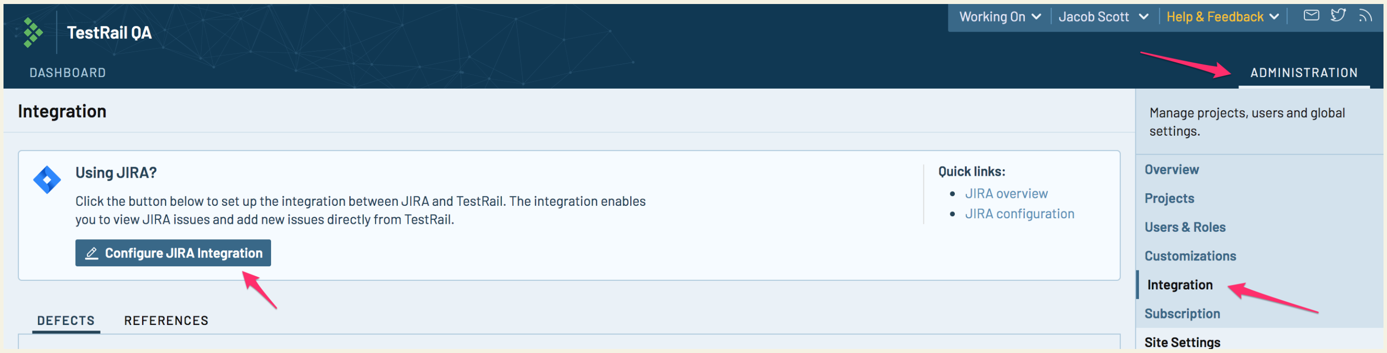
Configure TestRail for Jira Integration

TestRail supports integrating with Jira Cloud, Jira Server, and Jira Data Center. Simply follow these easy steps to configure your integration.

Before proceeding, ensure you are able to log in to a TestRail instance (start your [free trial here] if you don't currently have an active TestRail instance). You may need administrative access for some setup and configuration steps.

Integrate TestRail with your Jira project

- Log in to your TestRail instance and navigate to Administration > Integration to start configuring your settings. We recommend using the simple Jira configuration wizard at the top of the page. Simply click the Configure Jira Integration button to get started:



Integration

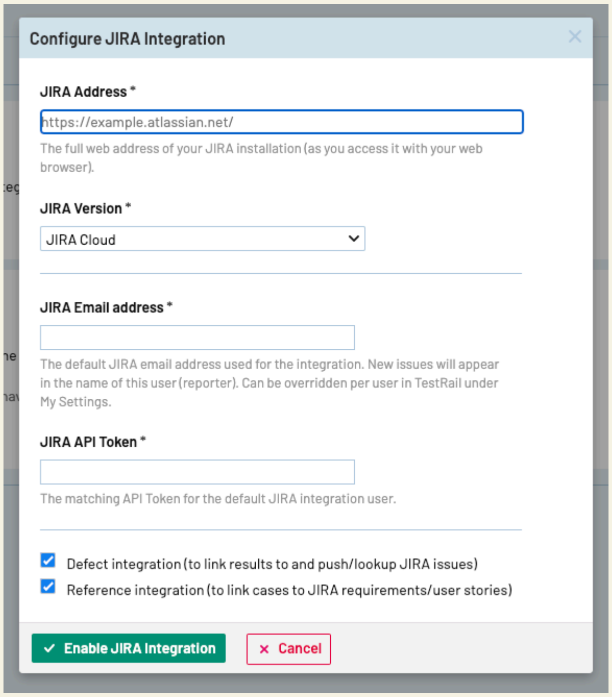
Using JIRA?

Click the button below to set up the integration between JIRA and TestRail. The integration enables you to view JIRA issues and add new issues directly from TestRail.

Manage projects, users and global settings.

- This will automatically help you set up the required Jira URLs, the defect integration to push/lookup Jira issues as well as the requirement integration.

- Scroll down and select **Save Settings** to finalize the changes.



Configure JIRA Integration

JIRA Address *

https://example.atlassian.net/

The full web address of your JIRA installation (as you access it with your web browser).

JIRA Version *

JIRA Cloud

JIRA Email address *

The default JIRA email address used for the integration. New issues will appear in the name of this user (reporter). Can be overridden per user in TestRail under My Settings.

JIRA API Token *

The matching API Token for the default JIRA integration user.

☑ Defect integration (to link results to and push/lookup JIRA issues)

☑ Reference integration (to link cases to JIRA requirements/user stories)

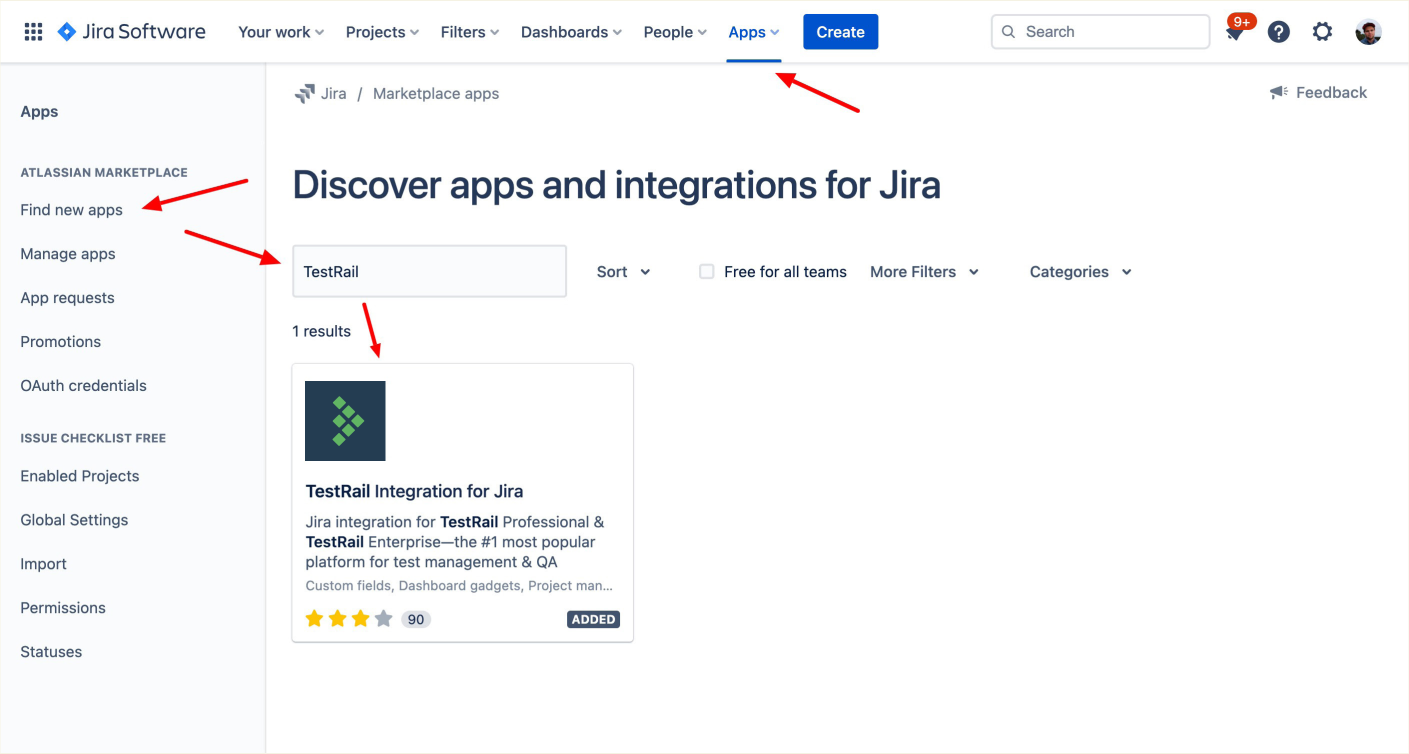
When integrating with Jira Cloud, you should use the **Jira Cloud plugin** in TestRail, and authenticate the integration using your **Jira email address** and **Jira API token**. For instructions on generating a Jira API token, [please refer to this documentation.]

Install the TestRail Jira App

The TestRail Jira app allows you to sync data between TestRail and Jira so that you and your team can check test coverage, view test plans and results, and even create new test cases or test runs from within Jira.

To get started, simply activate the TestRail app from the Atlassian Marketplace: Activate TestRail app from [Atlassian Marketplace.]()

Alternatively, you or your Jira administrator can enable the app within your Jira instance by navigating to Settings > Apps > Find New Apps , then search for TestRail. From there, click the TestRail app > Get App > Get It Now .



Jira / Marketplace apps

Discover apps and integrations for Jira

Search: TestRail

1 results

TestRail Integration for Jira

Jira integration for **TestRail** Professional & **TestRail** Enterprise—the #1 most popular platform for test management & QA

Custom fields, Dashboard gadgets, Project man...

Next, configure the app with your TestRail address and key. Click the **Apps** dropdown in the top menu, select **Manage apps**, expand the **TestRail for Jira Test Management** app, and **click Configure.** Then simply configure your TestRail address and integration key and any additional optional settings.

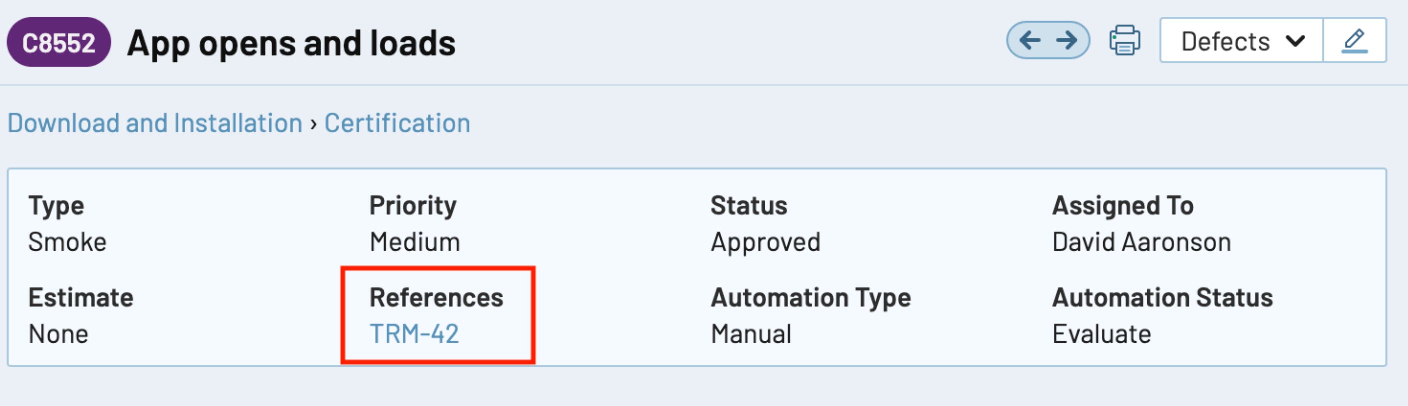
You can also limit the app to select Jira user groups or projects to hide the integration from users who don't currently use TestRail.

Start Using the TestRail Jira Integration

Link Issues to Test Cases to Create Two-Way Visibility Between TestRail and Jira

The greatest benefit of TestRail's integration with Jira is its two-way sync between platforms. After you have configured the integration, you can start the data sync by linking your requirements or issues, known as references within TestRail, to TestRail test artifacts like test cases, runs, plans, or milestones.

The References field is a vital part of the TestRail integration for Jira — **this field links your test cases to their associated Jira issues and epics.** Once your integration is set up, you simply have to **enter the ID number of the Jira issue you want to link to your TestRail artifact in the References field and you're all set.**



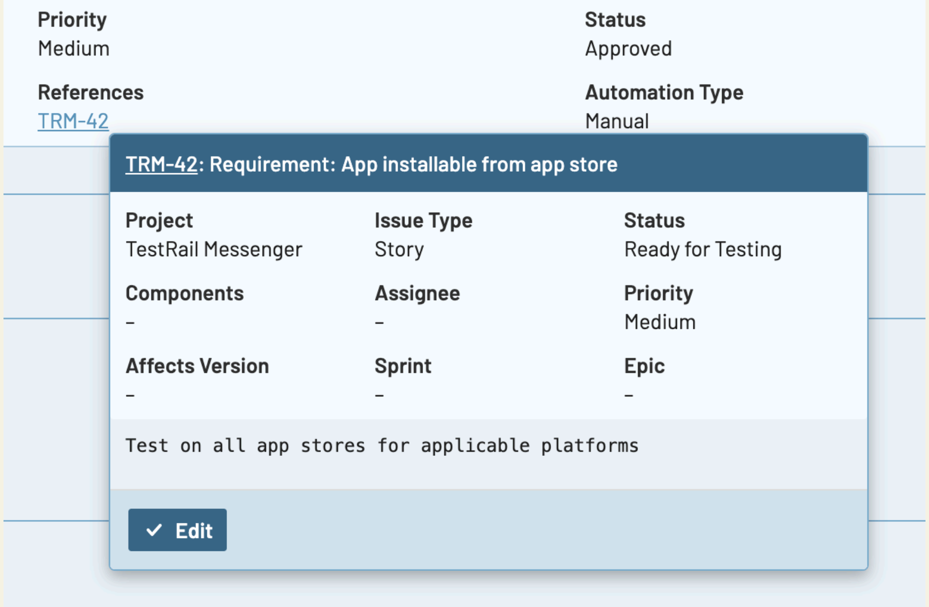
App opens and loads

Download and Installation › Certification

Type Smoke	**Priority** Medium	**Status** Approved	**Assigned To** David Aaronson
Estimate None	**References** TRM-42	**Automation Type** Manual	**Automation Status** Evaluate

You can also add references to test cases in bulk using the "bulk edit" feature in the test case repository, or [create test cases directly from user stories or issues in Jira.](#)

Once a reference is successfully linked, the Jira issue ID number will appear as a hyperlink. If you hover your cursor over the reference, it will display a window containing information about the linked Jira issue.



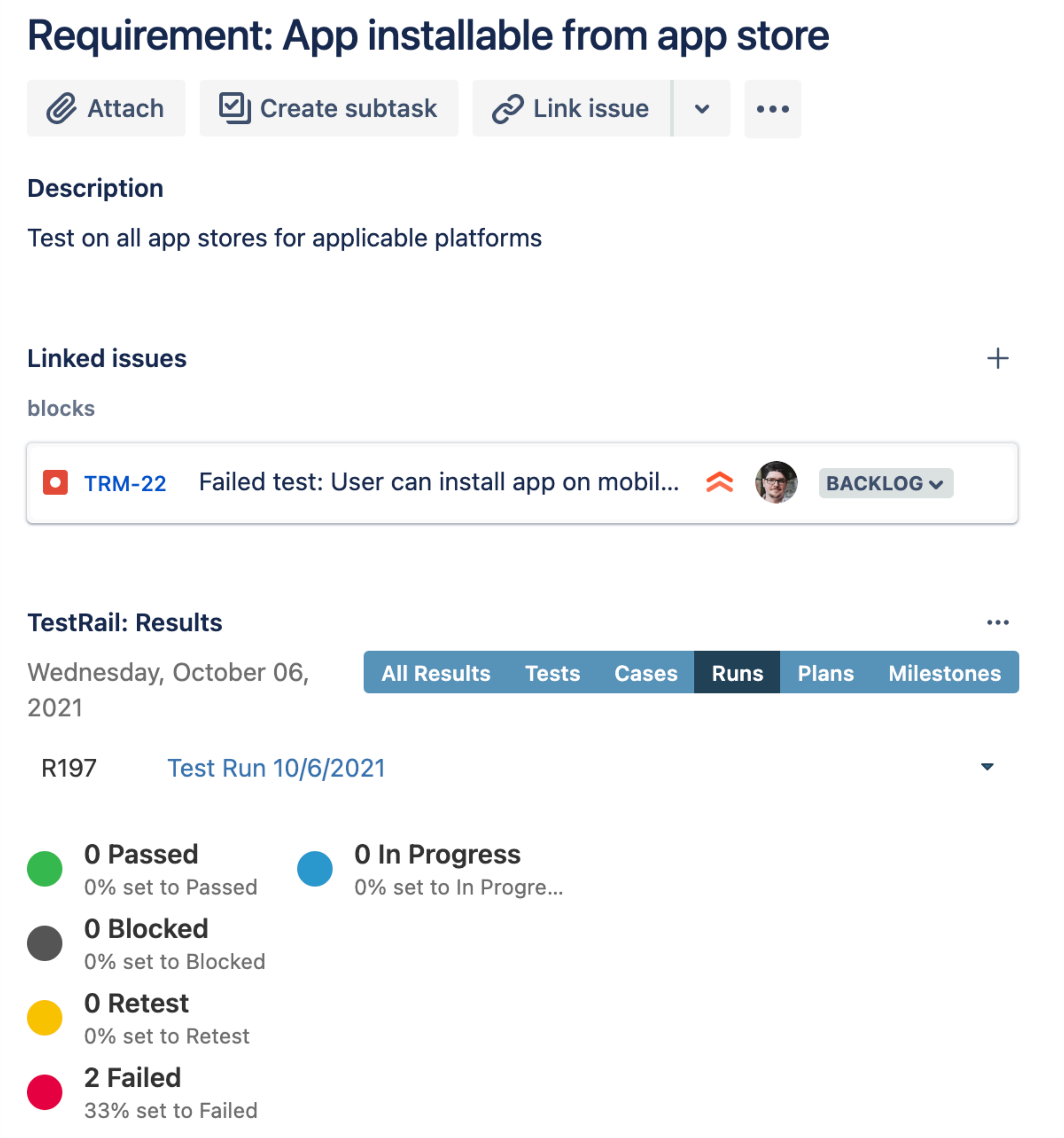
Priority Medium	**Status** Approved
References TRM-42	**Automation Type** Manual

TRM-42: Requirement: App installable from app store

Project TestRail Messenger	**Issue Type** Story	**Status** Ready for Testing
Components –	**Assignee** –	**Priority** Medium
Affects Version –	**Sprint** –	**Epic** –

```
Test on all app stores for applicable platforms
```

If you click the reference hyperlink, it will open the linked issue in Jira — where, assuming you have the TestRail for Jira add-on installed, you will also be able to see information from TestRail.



Requirement: App installable from app store

Description

Test on all app stores for applicable platforms

Linked issues

blocks

TRM-22 Failed test: User can install app on mobil... BACKLOG

TestRail: Results

Wednesday, October 06, 2021

R197 Test Run 10/6/2021

● 0 Passed
0% set to Passed

● 0 In Progress
0% set to In Progre...

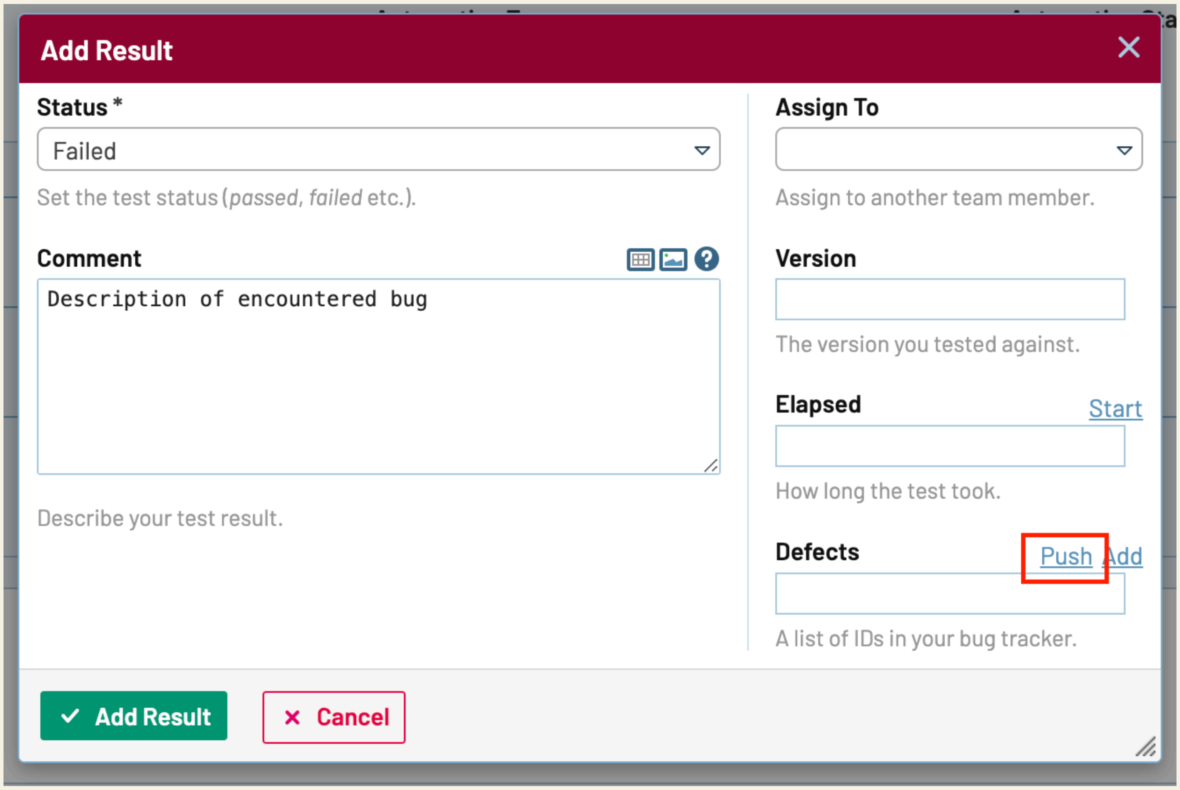
● 0 Blocked
0% set to Blocked

● 0 Retest
0% set to Retest

● 2 Failed
33% set to Failed

Automatically Create and Link Defects

Another great feature of the integration is the ability to create new issues for bugs or defects in Jira without leaving TestRail. When logging a test result in TestRail, simply click the ‘Push’ link to the right of the ‘Defects’ field. This will open a dialog box with a Jira issue description that is automatically pre-populated with data from your testing (including any comments you have added to your test result). You can review the issue description, fill out additional Jira fields, and add attachments to the issue before pushing all information from your TestRail test to Jira for your development team to review.



Add Result

Status *
Failed
Set the test status (*passed*, *failed* etc.).

Comment
Description of encountered bug
Describe your test result.

Assign To
Assign to another team member.

Version
The version you tested against.

Elapsed
How long the test took.

Defects
A list of IDs in your bug tracker.

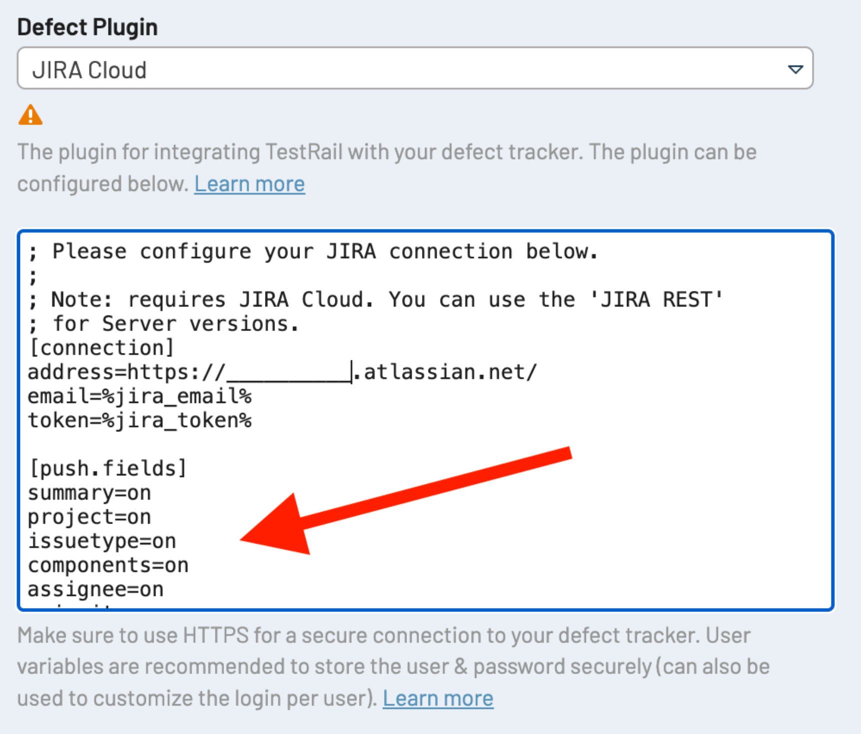
Customize the Integration to Your Workflow

Jira is infinitely customizable, and no two teams use it exactly alike. We’ve made it easy to customize both the Jira fields visible in the window that appears when you hover over linked references and the Jira fields that are available to fill in the push defect window.

To customize these templates, navigate to your TestRail dashboard then click ‘Administration’ in the upper right hand corner. From there, click ‘Integrations’ in the sidebar.

On the top sub-navigation, there will be two tabs called ‘Defects’ and ‘References.’ Start by clicking into Defects and scrolling down to the long-form text box.

In this box you will see two lists of fields under the headers **[push.fields]** and **[hover.fields]**. Push fields are the fields that TestRail users can update when creating new Jira issues from within TestRail, and hover fields are the Jira fields visible when hovering over a linked defect from TestRail. To enable or disable a field in either view, simply set the value to “on” or “off” in the configuration settings window.



Defect Plugin

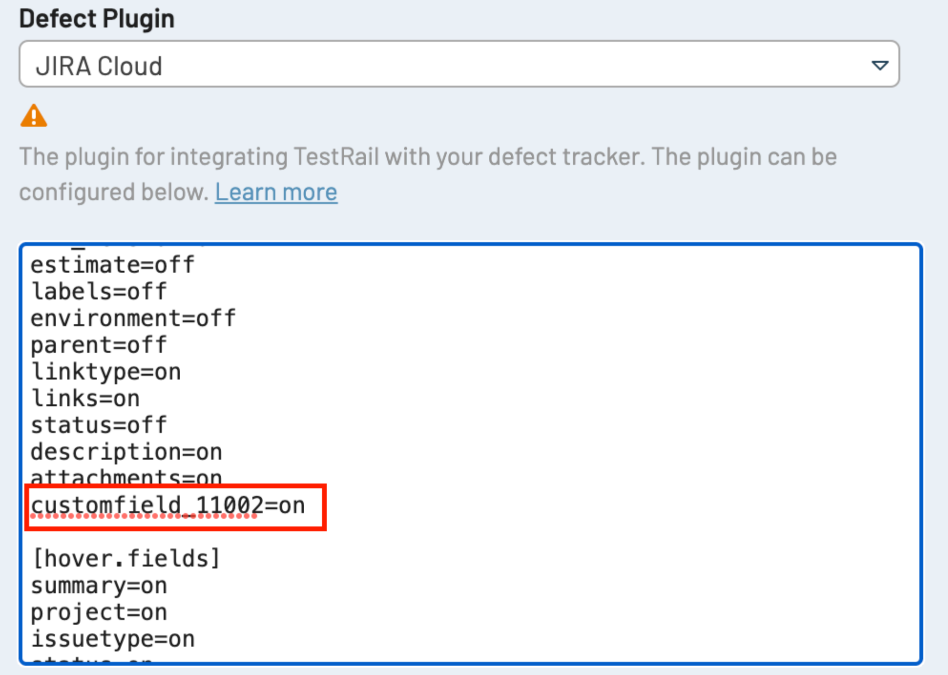
JIRA Cloud

The plugin for integrating TestRail with your defect tracker. The plugin can be configured below. Learn more

```
; Please configure your JIRA connection below.
;
; Note: requires JIRA Cloud. You can use the 'JIRA REST'
; for Server versions.
[connection]
address=https://__________.atlassian.net/
email=%jira_email%
token=%jira_token%

[push.fields]
summary=on
project=on
issuetype=on
components=on
assignee=on
```

Make sure to use HTTPS for a secure connection to your defect tracker. User variables are recommended to store the user & password securely (can also be used to customize the login per user). Learn more

You can also use the above-mentioned approach to add custom fields to TestRail’s Push Defect dialogue to pass additional information along to Jira. To do this, simply add a line with your custom field to the [push.fields] section to add a field:



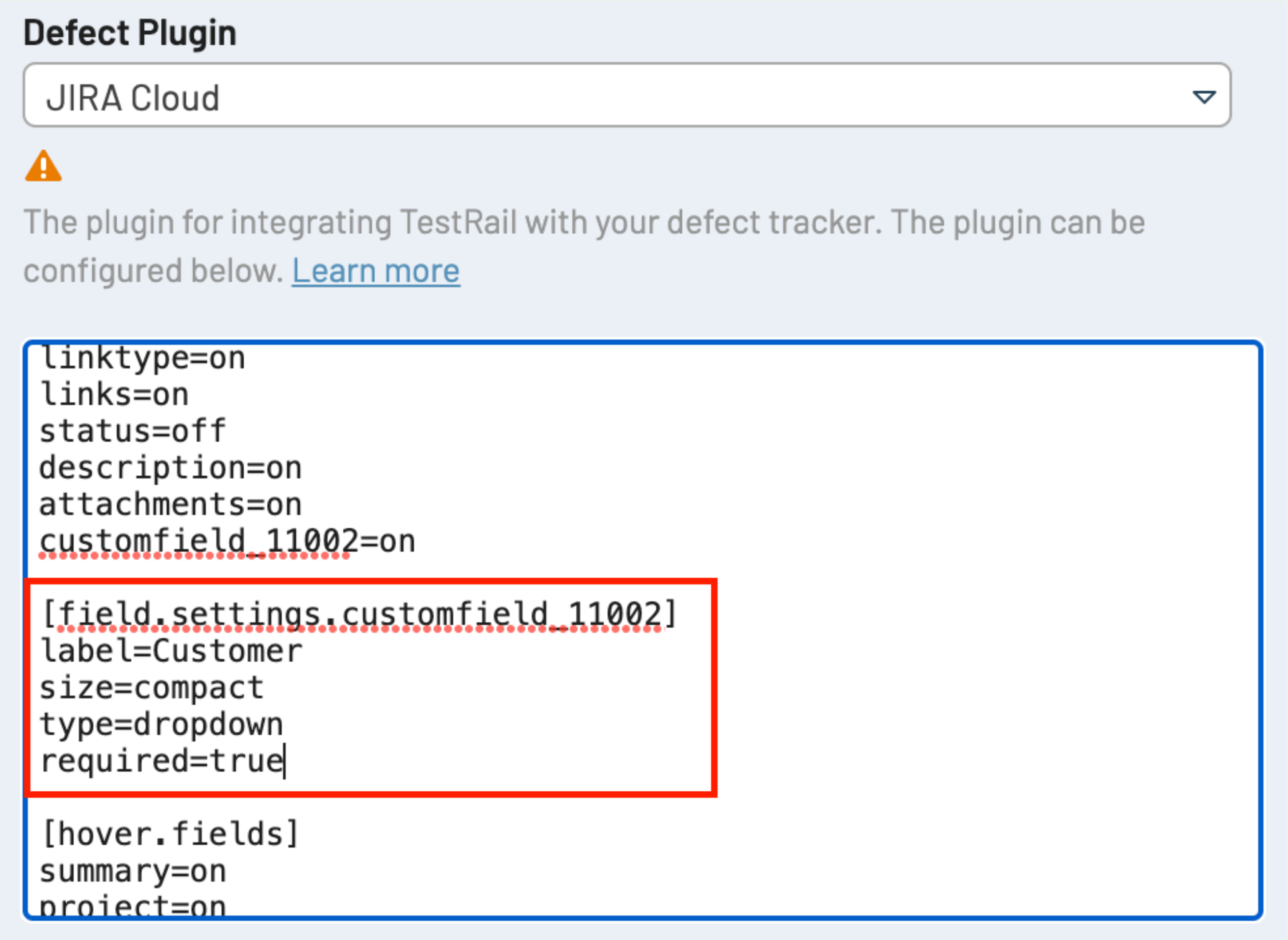
Defect Plugin

JIRA Cloud

The plugin for integrating TestRail with your defect tracker. The plugin can be configured below. Learn more

```
estimate=off
labels=off
environment=off
parent=off
linktype=on
links=on
status=off
description=on
attachments=on
customfield_11002=on

[hover.fields]
summary=on
project=on
issuetype=on
```

Then, add a [field.settings.{custom field identifier}] line. This section is required so the plugin can identify the correct custom field property to push and/or retrieve data from Jira. The following is an example custom field configuration for Jira:



Defect Plugin

JIRA Cloud

⚠

The plugin for integrating TestRail with your defect tracker. The plugin can be configured below. Learn more

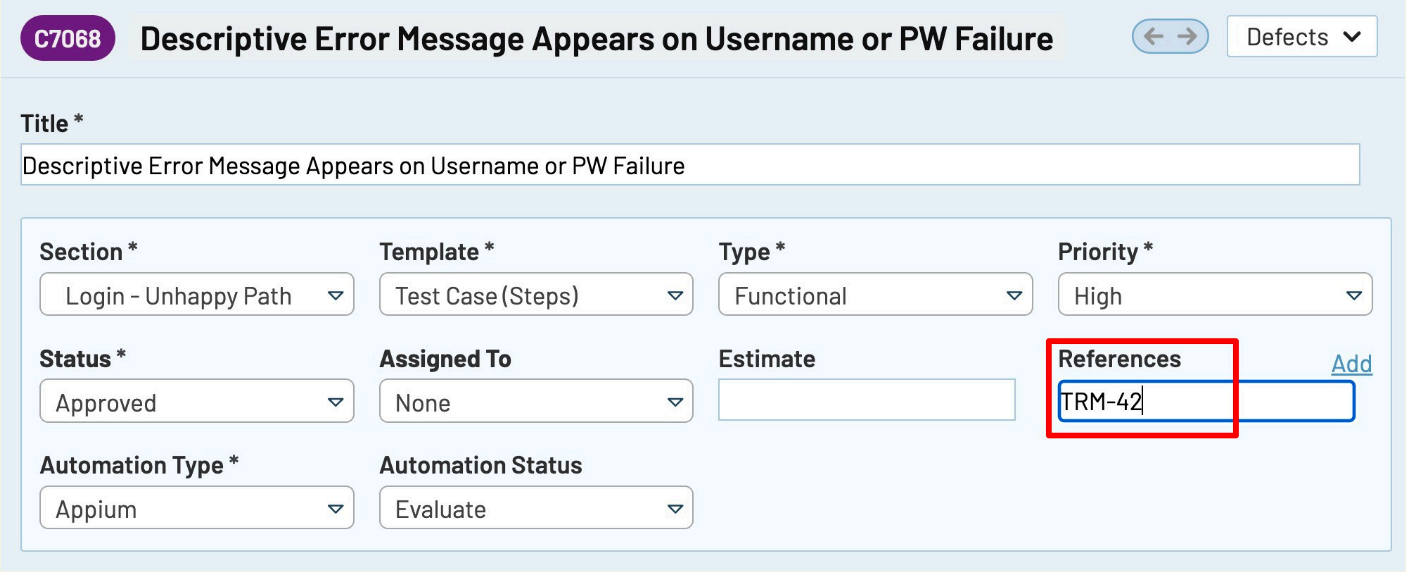
```
linktype=on
links=on
status=off
description=on
attachments=on
customfield_11002=on

[field.settings.customfield_11002]
label=Customer
size=compact
type=dropdown
required=true

[hover.fields]
summary=on
project=on
```

Next, click into References. You will notice that only hover fields are available here — you can adjust these fields' values to customize what is displayed in TestRail when a user hovers over a linked reference.

To learn more about configuring defect fields, including advanced customization options, visit our [documentation center.](#)

Track Test Coverage Using TestRail and Jira

As soon as you have linked requirements or user stories from Jira to test cases in TestRail, you can start generating traceability and coverage reports.



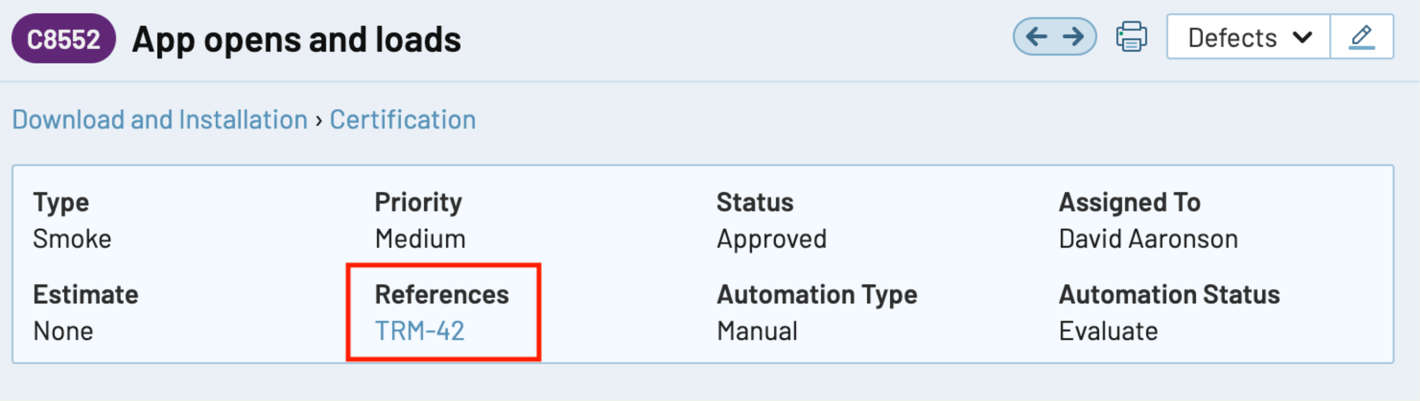
C7068 Descriptive Error Message Appears on Username or PW Failure

Title *
Descriptive Error Message Appears on Username or PW Failure

Section *	Template *	Type *	Priority *
Login - Unhappy Path	Test Case (Steps)	Functional	High
Status *	Assigned To	Estimate	References
Approved	None		TRM-42
Automation Type *	Automation Status		
Appium	Evaluate		

Linking References

Before running a coverage report, ensure all of your test cases are linked to their associated Jira issues. Once your integration is set up, you simply have to enter the ID number of the Jira issue you want to link to your TestRail artifact in the References field and you're all set.



C8552 App opens and loads

Download and Installation › Certification

Type	Priority	Status	Assigned To
Smoke	Medium	Approved	David Aaronson
Estimate	References	Automation Type	Automation Status
None	TRM-42	Manual	Evaluate

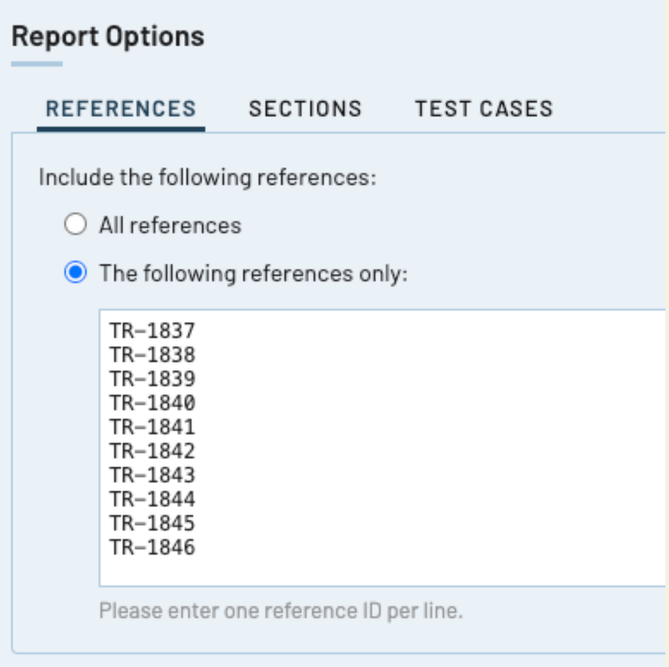
You can also add references to test cases in bulk using the "bulk edit" feature in the test case repository, or create test cases directly from specific user stories in Jira.

Running a Coverage Report

To run a Coverage for References report in TestRail, start by copying the Jira issue IDs for your user stories or requirements. The quickest way to do this is to export the issues from your Jira project to a CSV file. This will allow you to copy the Jira issue IDs into your Coverage for References all at once.

Then, follow these steps to generate your report:

- Navigate to your TestRail project
- Click the "Reports" tab on the top toolbar
- Click "Coverage for References" under "Create Report" on the right-hand toolbar
- Scroll down to "Report Options" then "References"
- Select "The following references only:"
- Paste your list of Jira issue IDs into the text box below



Report Options

Include the following references:

○ All references

● The following references only:

```
TR-1837
TR-1838
TR-1839
TR-1840
TR-1841
TR-1842
TR-1843
TR-1844
TR-1845
TR-1846
```

Please enter one reference ID per line.

Now you're ready to run your report. **This report will:**

- Map all connections between references and test cases
- Allow you to see how many references are linked to each test case
- Show if there are any requirements not covered by any test cases or test cases that don't have any linked references
- Highlight if you missed linking References anywhere

The Coverage for References report makes it easy to evaluate the effectiveness of your testing plans and whether you need to recalibrate your plan for more efficient coverage of your requirements.

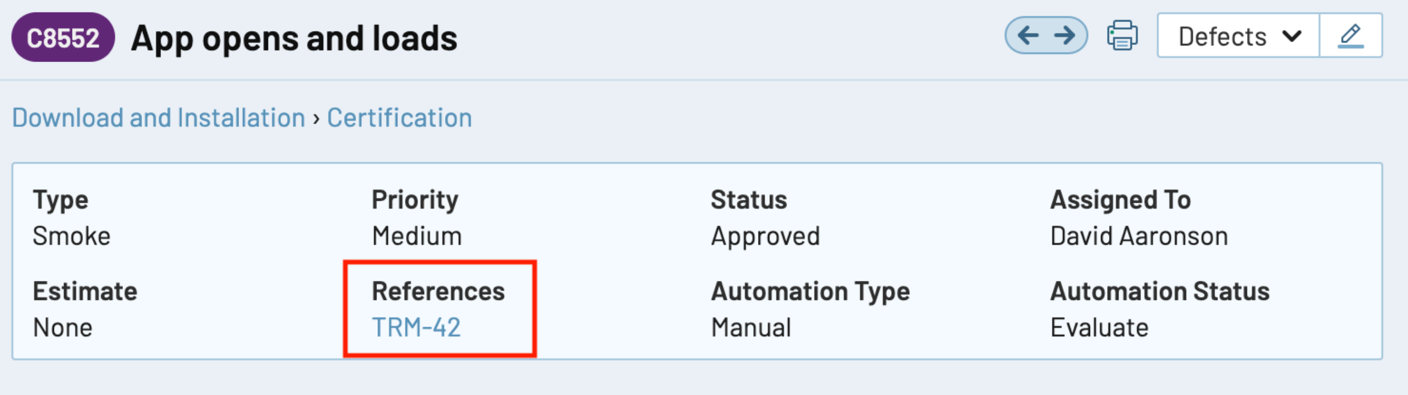
For more information on setting up test coverage with Jira, watch this video!

Build Full Traceability Between Requirements, Testing, and Defects

The key to building traceability between TestRail and Jira is to link references with test cases and link defects with test results. Then, you can easily generate a full traceability matrix.

Linking References

To start building traceability, it is vital to ensure all of your test cases are linked to their associated requirements in Jira. Once your integration is set up, you simply have to enter the ID number of the Jira issue you want to link to your TestRail artifact in the References field and you're all set.



C8552 App opens and loads

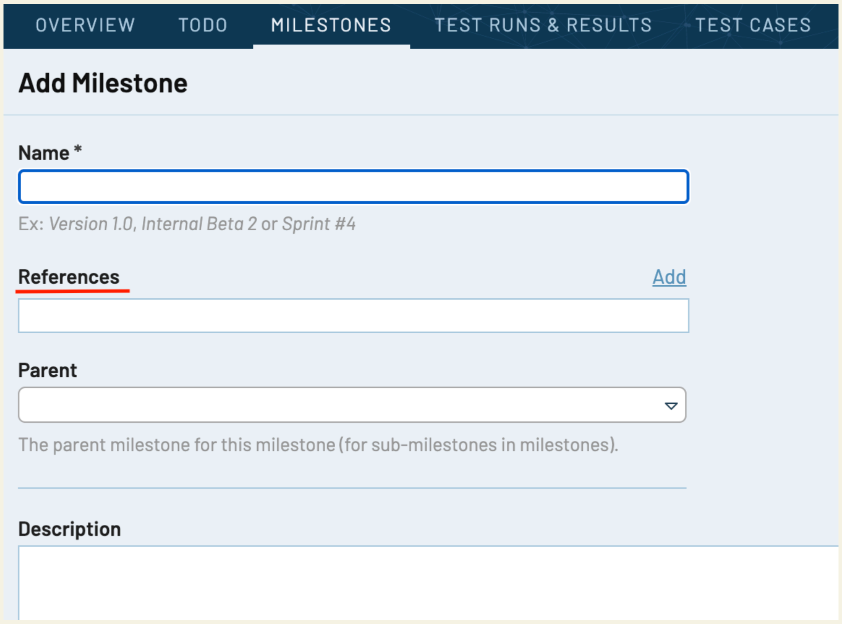
Download and Installation › Certification

Type Smoke	**Priority** Medium	**Status** Approved	**Assigned To** David Aaronson
Estimate None	**References** TRM-42	**Automation Type** Manual	**Automation Status** Evaluate

You can also add references to test cases in bulk using the "bulk edit" feature in the test case repository, or create test cases directly from specific user stories in Jira.

Plan Your Testing and Link Epics

Once you have all of your test cases in your testing repository, it's time to plan your testing. The highest level of organization in TestRail are called milestones, which are used as containers for testing activities related to a specific, time-bound goal (like a particular development sprint or product release). If you use Jira epics to group test cases, we recommend linking your TestRail milestones to your Jira epics to build traceability. You can add your Jira epic's issue ID while adding new milestones or edit your milestone later.



Add Milestone

Name *

Ex: *Version 1.0, Internal Beta 2 or Sprint #4*

References

Parent

The parent milestone for this milestone (for sub-milestones in milestones).

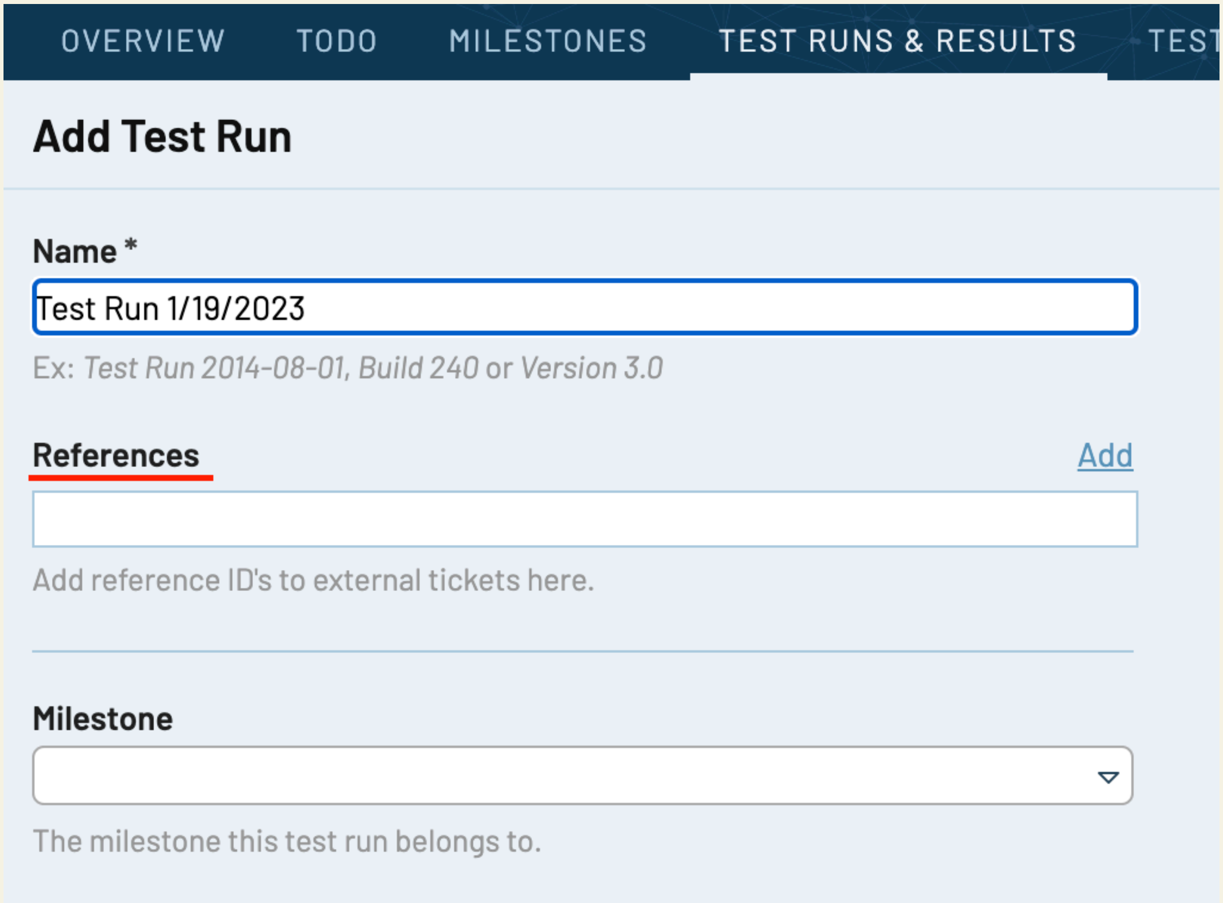
Description

Now, time to populate your milestone with test runs and plans.

In TestRail, test runs contain the actual list of tests your team is performing; you add test cases to a test run to generate and execute your tests.

Test plans allow you to group multiple related test runs, and even allow you to generate many test runs at once if you want to test against multiple configurations. Configurations typically represent environments you need to test your projects against, like different operating systems, web browsers, or devices.

You can further help align development sprints with test cycles by linking test runs, test plans, and/or milestones to Jira artifacts, such as epics. This can be done by entering the Jira issue ID in the References field on test runs, plans, and/or milestones.



Add Test Run

Name *

Test Run 1/19/2023

Ex: *Test Run 2014-08-01, Build 240* or *Version 3.0*

References

Add reference ID's to external tickets here.

Milestone

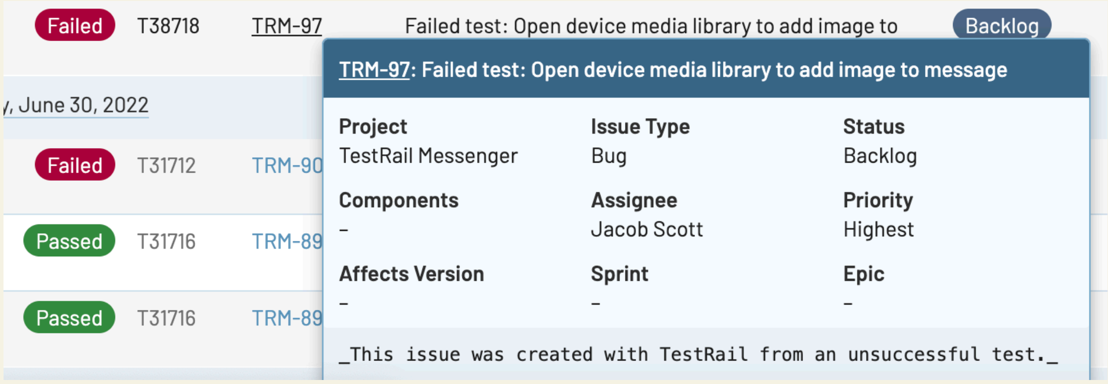
The milestone this test run belongs to.

Run Tests and Report Defects

Now it's time to test! When your team encounters bugs, make sure they either push a defect to Jira while logging their test result (as outlined above), or enter a Jira ID into the defect reference field if the issue already exists or your tester prefers to log defects directly in Jira. Once you have pushed a new defect or linked an existing defect ID, you can hover the defect reference to see Jira fields from TestRail. You can also edit that Jira data directly from TestRail in case you need to update the issue status or another Jira field.

Review Results

Once you have started to report your test results within test runs, you can navigate to your milestone to view all test results under that milestone. You can click “Defects” in the right-hand toolbar to see all defects logged under the milestone, and can hover over the defect ID links to view the defects’ Jira details directly from TestRail.



Failed T38718 TRM-97 Failed test: Open device media library to add image to Backlog

, June 30, 2022

Failed T31712 TRM-90

Passed T31716 TRM-89

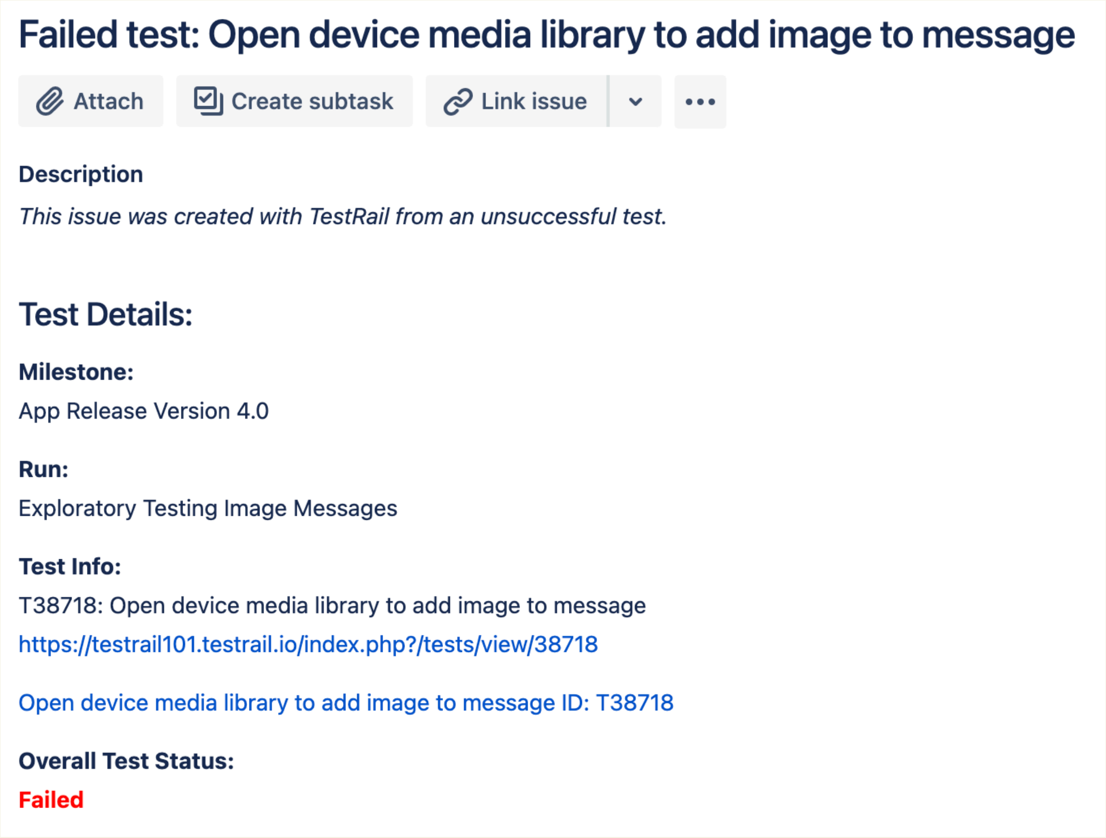
Passed T31716 TRM-89

TRM-97: Failed test: Open device media library to add image to message

Project	Issue Type	Status
TestRail Messenger	Bug	Backlog
Components	**Assignee**	**Priority**
–	Jacob Scott	Highest
Affects Version	**Sprint**	**Epic**
–	–	–

This issue was created with TestRail from an unsuccessful test.

Additionally, assuming you have installed the TestRail Integration for Jira app, you can view the status and details of tests against a Jira issue or epic from within Jira.



Failed test: Open device media library to add image to message

Description

This issue was created with TestRail from an unsuccessful test.

Test Details:

Milestone:
App Release Version 4.0

Run:
Exploratory Testing Image Messages

Test Info:
T38718: Open device media library to add image to message
https://testrail101.testrail.io/index.php?/tests/view/38718

Open device media library to add image to message ID: T38718

Overall Test Status:
Failed

Generate Traceability Reports

Now that you're at the end of your test cycle, it's time to assess if your testing goals were achieved by generating a traceability report. There are two types of traceability reports available in TestRail:

- **Results Comparison for References**
  Shows the results for references in a coverage & comparison matrix
- **Defects Summary for References**
  Shows a summary of defects logged during testing and the latest test status, grouped by references and test cases in a coverage matrix

Much like the Coverage for References report we generated in the section above, you'll most likely want to first export your list of requirements as issues from your Jira project to CSV. Then, follow these simple steps to generate your reports:

- Navigate to your TestRail project
- Click the "Reports" tab on the top toolbar
- Click "Results -> Comparison for References" or "Defects -> Summary for References " under "Create Report" on the right-hand toolbar
- Scroll down to "Report Options" then "References"
- Select "The following references only:"
- Paste your list of Jira issue IDs into the text box below
- Run your report

Now, using the power of Jira and TestRail, you've achieved full traceability! These reports will produce matrices of results, coverage, and defects as pertaining to your requirements in Jira. For more information on building requirements traceability with Jira, **watch this video.**

FAQs and Troubleshooting

Do you need to be logged into your TestRail account to view TestRail results within Jira?

Yes. Users who wish to view TestRail results and information within Jira will also need to have TestRail accounts. TestRail pricing is simple and transparent and you only pay for the users you need — **learn more about our pricing and plans here.**

Is it possible to have two requirements covered by one test case or vice versa?

Yes — you can link multiple references to the same entity in TestRail by adding them in a comma-separated list.

If we want to do multiple runs over time (i.e. checking your login checks every sprint) do we need to go and change the references every time?

Not usually! Typically when you run a regression test, you won't create a new requirement or user story. As long the requirements being tested for regressions aren't changing themselves, you don't need to change references set in the test cases and can execute each test case as often as you need while maintaining full traceability.

In addition to maintaining traceability between test cases/tests and your original requirements, you may want to show that a particular instance of a regression test was run for a given release, as part of a specific epic, or against a certain version of your software. To do this, you can use the reference field on the test run, plan, or milestone that the test is a part of. This allows you to extend traceability all the way from the original requirement through to the test case and each instance of that test over time.

If you include screenshots/images in a TestRail bug report, will they be included when pushing to Jira?

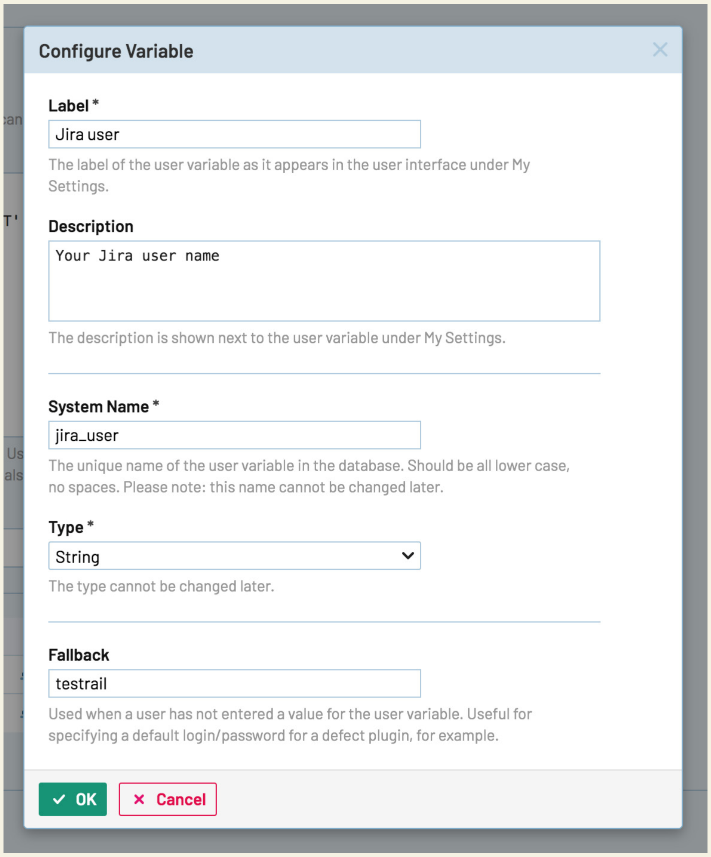
Yes — the information you capture about the bug in your test results and all associated evidence submitted through the Push Defect dialog will be saved and visible in Jira.

How can my testers log defects from TestRail using their own Jira accounts rather than the admin user?

User variables settings allow your TestRail users to use their own Jira account to push bug reports to Jira from TestRail — rather than all defects being logged by a generic "QA" account or similar — boosting visibility and traceability.

You can add user variables under Administration > Integration. Once you've defined user variables, users can enter their details for the variables under My Settings (via the My Settings link in the upper right corner of TestRail's user interface).

For more information on configuring user variables, please refer to this [documentation.]()



Configure Variable

Label *

Jira user

The label of the user variable as it appears in the user interface under My Settings.

Description

```
Your Jira user name
```

The description is shown next to the user variable under My Settings.

System Name *

jira_user

The unique name of the user variable in the database. Should be all lower case, no spaces. Please note: this name cannot be changed later.

Type *

String

The type cannot be changed later.

Fallback

testrail

Used when a user has not entered a value for the user variable. Useful for specifying a default login/password for a defect plugin, for example.

Can I use TestRail to integrate with my test automation tool or CI/CD platform to show traceability with automated tests?

Yes — you can integrate TestRail with your test automation framework using the TestRail CLI or TestRail API.

If you document test cases for your automated tests in TestRail, you can link your cases to requirements in Jira to demonstrate traceability with your test automation.

Once you have uploaded test automation results to TestRail, you can review your results and create or link defects in Jira like you might do for tests you are running manually. This allows you to surface potential issues discovered by test automation more quickly and ensure full traceability across all of your testing efforts.

[Learn more about the TestRail API here]()

[Learn more about the TestRail CLI here]()

What is the best way to map TestRail Milestones to Jira?

Ultimately, the answer to this questions depends on how your team uses Jira. We typically recommend connecting your TestRail Milestones to Jira Epics via the reference field. If your team uses sprints, you can use the start and end date for the milestone to align your testing cycle timeline.

You can watch a demonstration on how to connect Milestones and Epics in our “Full Traceability for All with the Jira Integration” **on-demand webinar here.**

Can TestRail be integrated with Jira at both an instance level and a project level?

Yes — you can create a global integration to connect TestRail to all projects in your instance, or create specific integrations on a per-project basis. However, to integrate TestRail with multiple Jira instances, you will need to create separate projects in your TestRail instance; you can connect the same TestRail project to multiple projects in the same Jira instance, but you cannot connect multiple Jira instances to the same TestRail project.