

# Creating a Precondition and Associating with a Test.

precondition is like a prerequisite for test to run

options:1) add new precondition

2) add existing precondition

By default the preconditions are not added after creating ,so we have to associate with a test in order to do so or we have to create precondition within the test itself.

## Test details

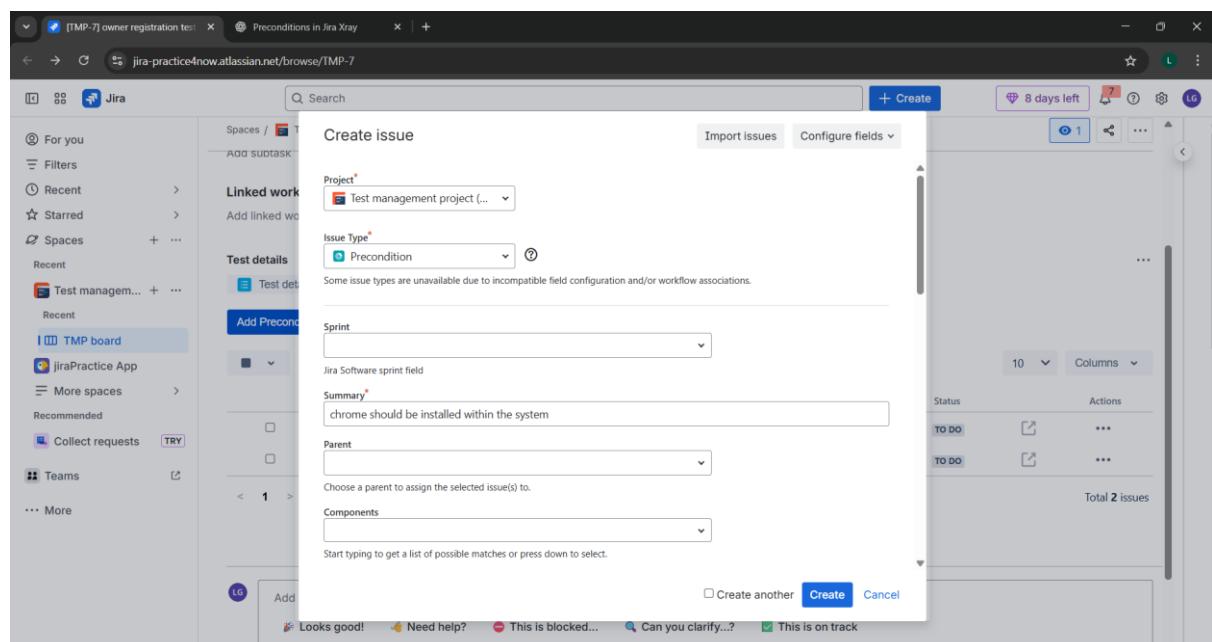


Fig: Creating pre-condition as issue (outside a test)

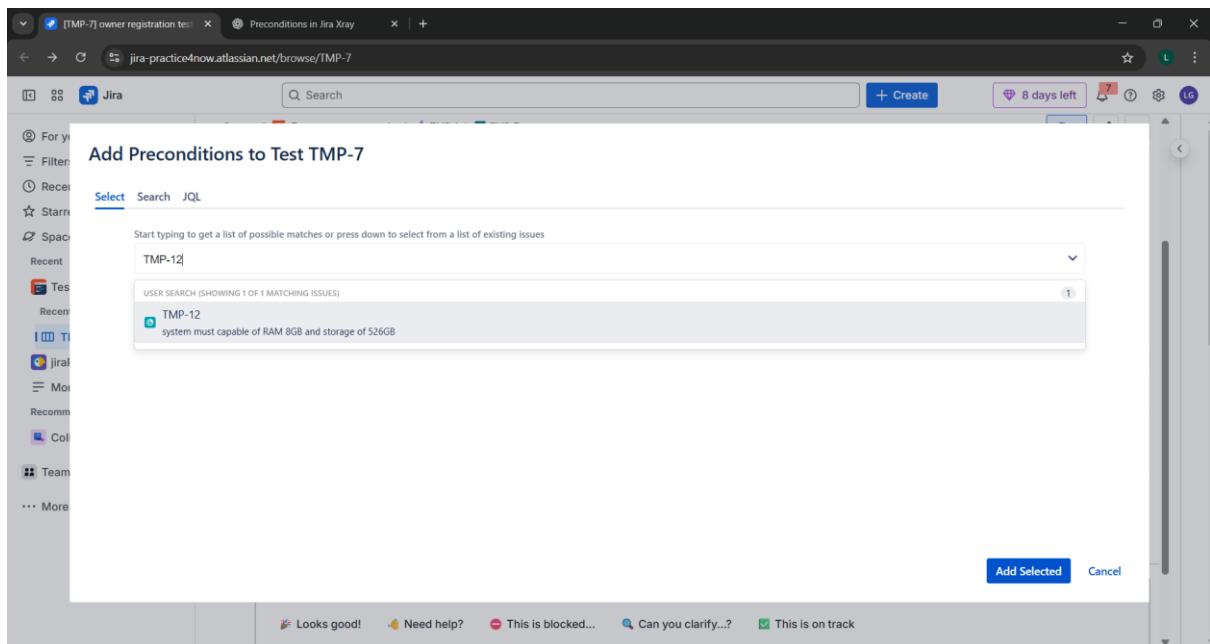


Fig: Associating existed precondition to a test

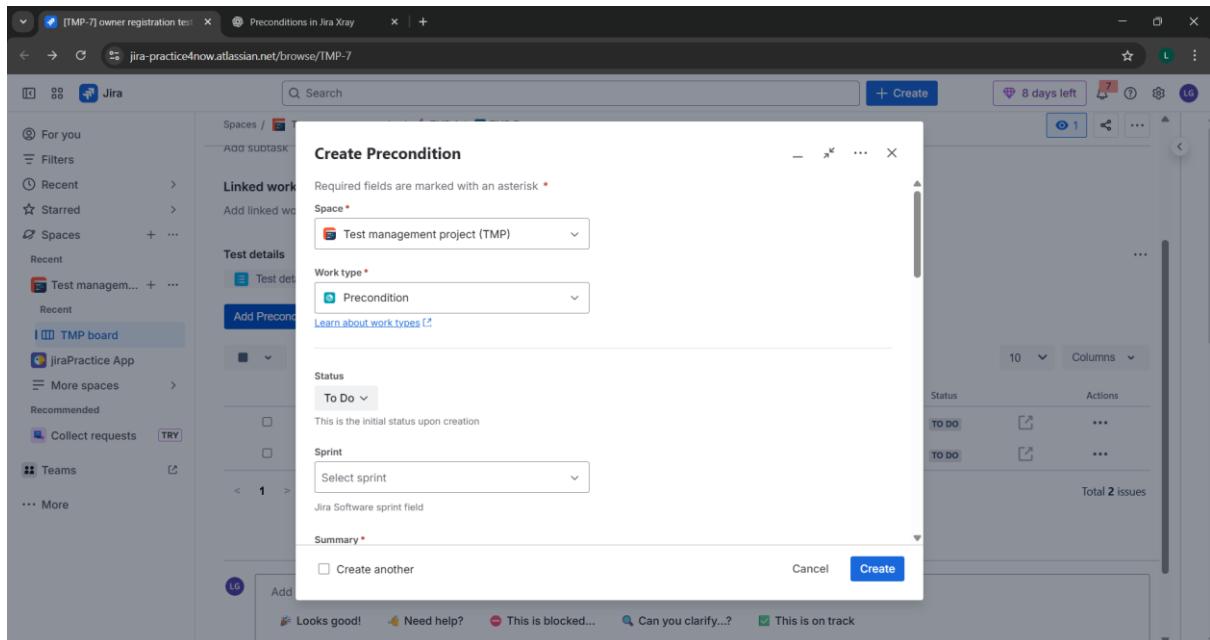


Fig: Creating a Precondition within the test

# Test Set

- Test Set is a Collection of related tests

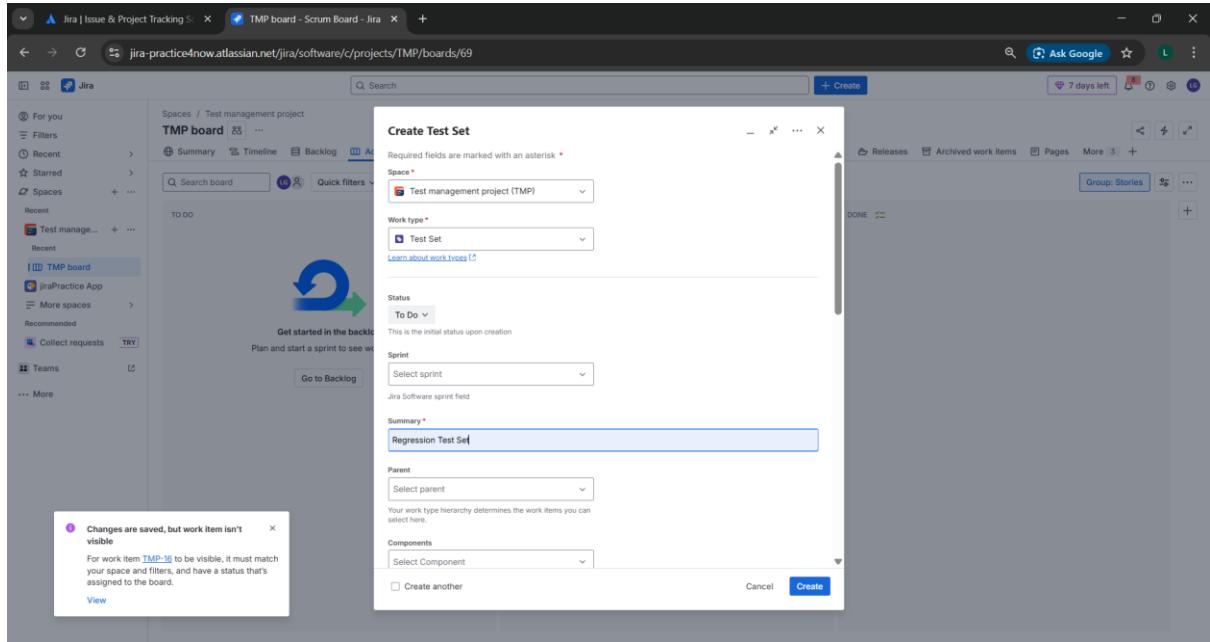


Fig: Creating Test Set issue

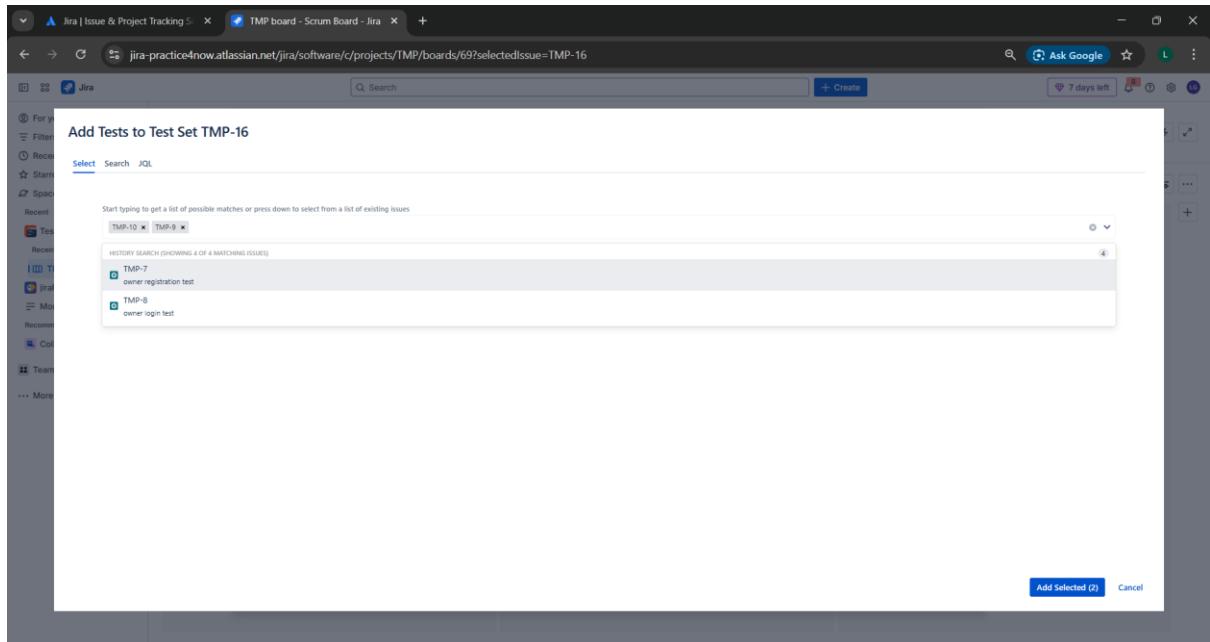


Fig: Adding existing tests to the test set.

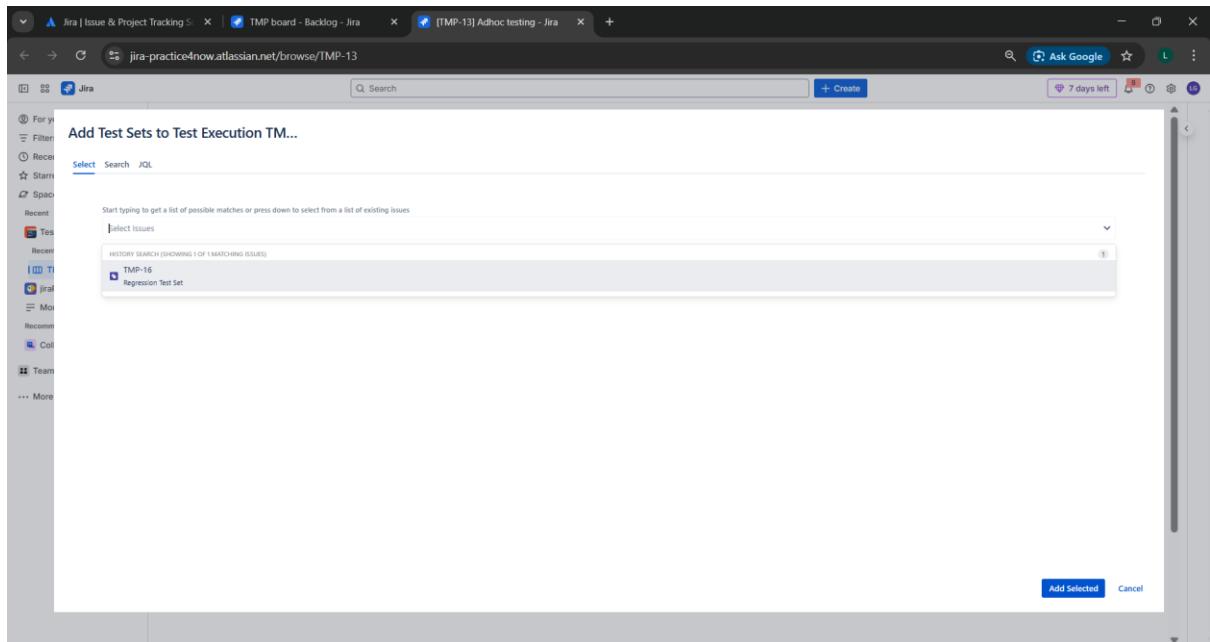


Fig: Adding existing test set to test execution.

# Creating a test execution and running the added tests.

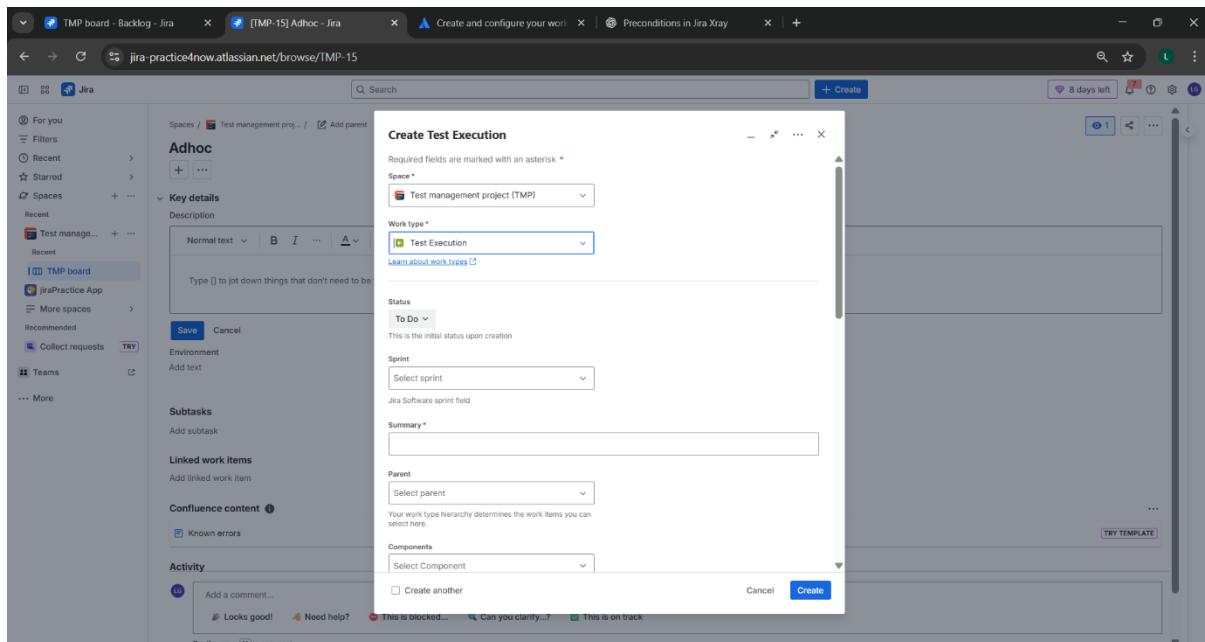


Fig: creating a test Execution

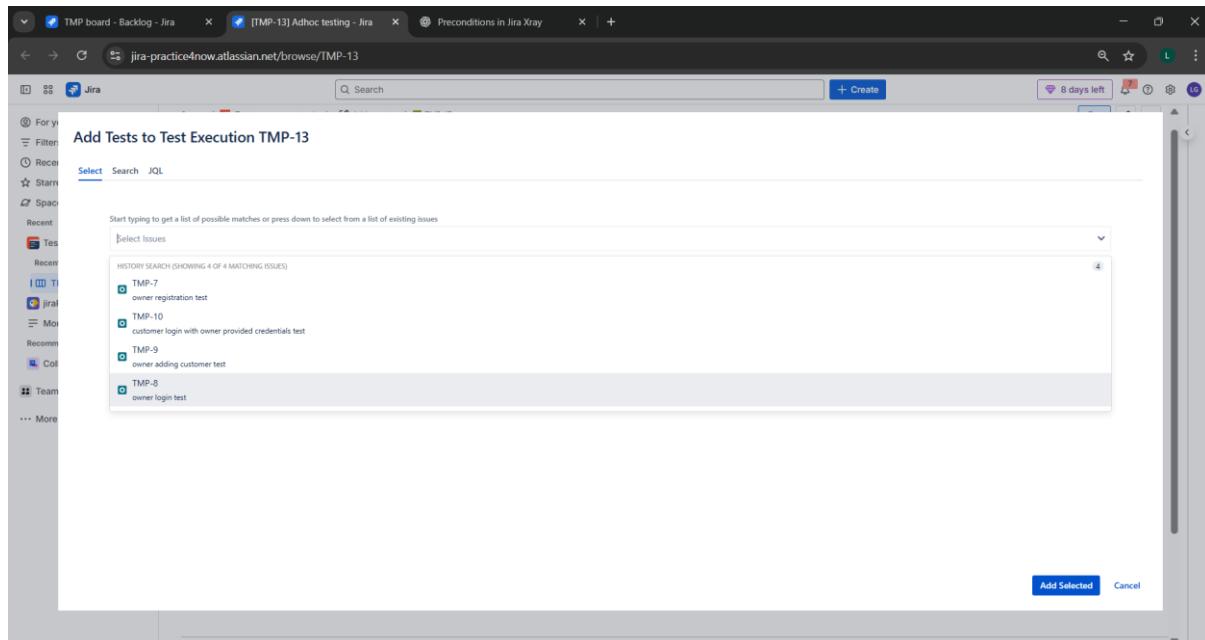


Fig: Adding Existing tests to this test execution

The screenshot shows a Jira board interface. On the left, there's a sidebar with navigation links like 'For you', 'Recent', 'Starred', 'Spaces', etc. The main area displays a test item titled 'Adhoc testing'. Under 'Tests', there is one entry: 'TMP-7 owner registration test' (Test Type: Manual, Status: TO DO). The overall execution status bar at the bottom shows 'TOTAL TESTS: 1' and 'TO DO'. A play button is located in the top right corner of the test item view.

Fig: After adding tests and then click play button to run the step

This screenshot shows the detailed view of the 'owner registration test' from the previous board. It includes sections for 'Test details' (with tabs for 'MANUAL' and 'AUTOMATED'), 'Preconditions' (listing 'TMP-11 - Chrome browser should be installed in the system and must have proper internet connectivity' and 'TMP-12 - system must capable of RAM 8GB and storage of 526GB'), and 'Steps'. The first step is listed with an action ('owner enters email'), data ('owner1@gmail.com'), and expected result ('owner verified'). Below this, the 'Actual Result' section shows 'entered email is verified'. The status bar at the bottom right indicates 'The test step has passed'.

The screenshot shows a test execution interface with the following details:

- Execution Status:** EXECUTING
- Timer:** 00:00:15
- Started On:** 22-Jan-2020 10:46 AM
- Assigned To:** Unsigned
- Executed By:** Lokesha Gobals
- Version:** v1
- Test Version:** v1
- Test Environments:** Not specified

**Test details:**

- Preconditions:**
  - TMH-11 - Chrome browser should be installed in the system and must have proper internet connectivity
  - TMH-12 - system must capable of RAM 8GB and storage of 512GB
- Steps:**

Action	Data	Expected Result
owner enters email	owner@gmail.com	owner verified
Actual Result	entered email is verified	
Actual Result		

**Step State:** PASSED

Fig: Adding manual actual output and saving it then clicking the “green” button

The screenshot shows the Xray Settings page with the following navigation:

- Space settings
- Test management project
- Software space
- Details
- Summary
- People
- Permissions
- Notifications
- Automation
- Features
- Toolchain
- Workflows
- Work items
- Versions
- Components
- Apps
- Xray Settings

The main content area is titled "Defect Mapping". It includes two sections:

- Available Issue Types:**
  - Task
  - Sub-task
  - Story
  - Epic
  - Test
  - Test Set
  - Test Plan
  - Test Execution
  - Precondition
  - Sub Test Execution
- Defect Issue Types:**
  - Bug

A "Save" button is located at the bottom of the mapping section.

Fig: Mapping the Xray’s “Defect” issue type to Jira’s “Bug” issue type

The screenshot shows a failing test step with the following details:

- Action:** owner clicks on ok button
- Data:** None
- Expected Result:** owner should get registered successfully
- Actual Result:** registration is unsuccessful
- Step State:** FAILED

Below the step details, there are buttons for Comment, Defects (Create Defect, Create Sub-Defect, Add Defects), and Evidence.

Fig:Failing a step

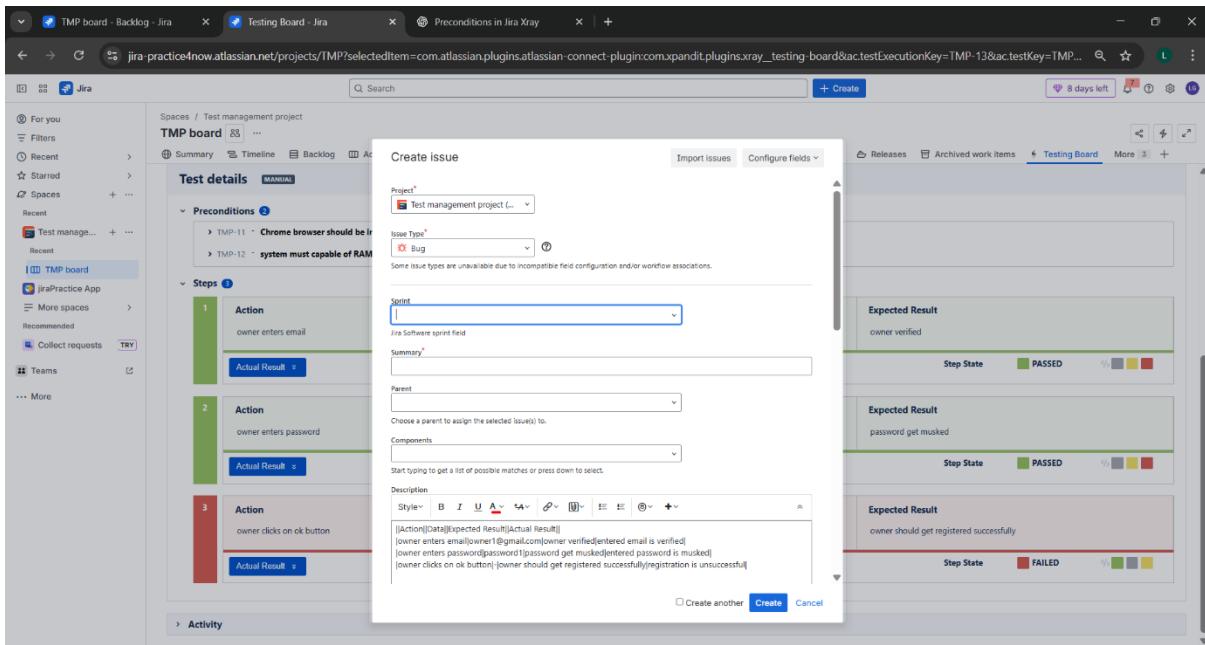


Fig: Creating a Defect within the failed step

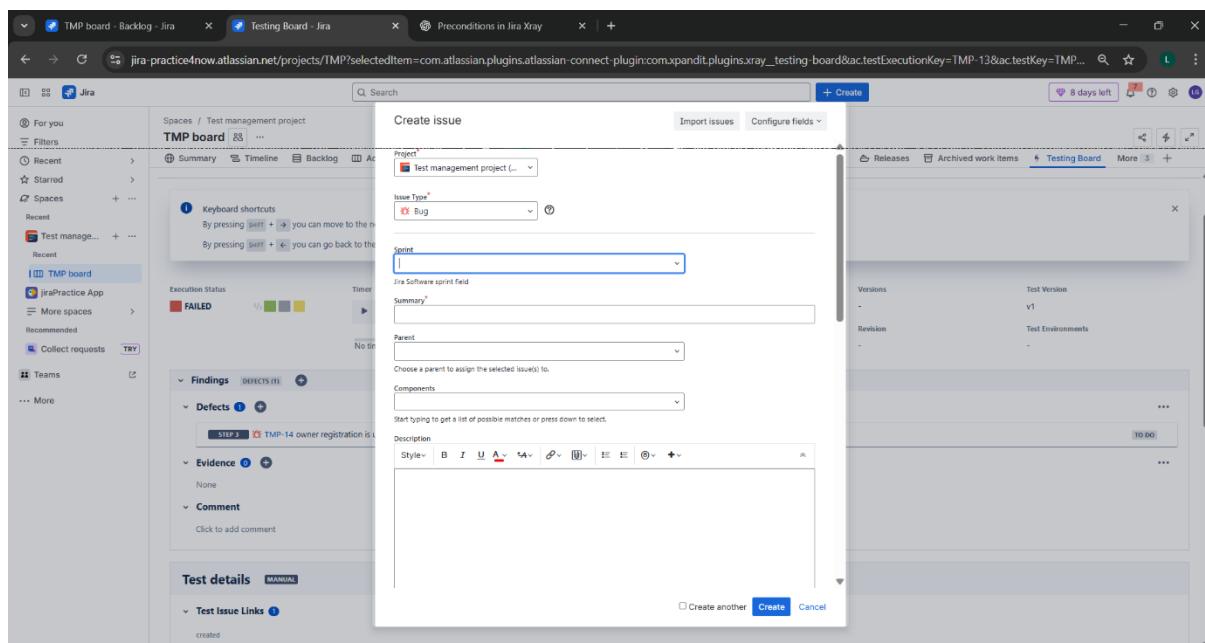
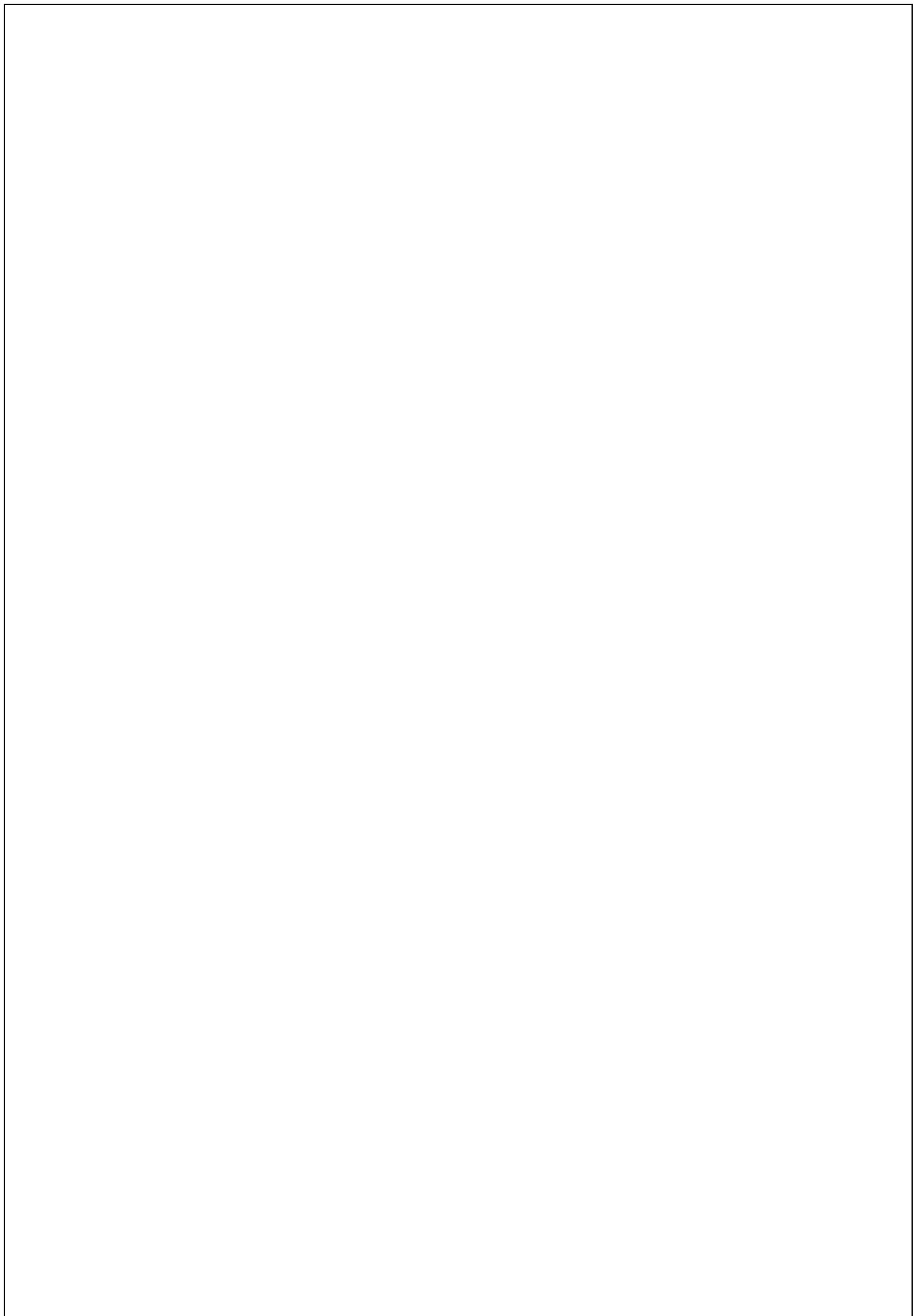


Fig: Creating a Defect issue outside the failed step



# Test Plan

- Test Plan is like placing the related test issues under one umbrella
- Where we can add test, test sets and test execution
- We test plan for a sprint, version, release or for a specific testing

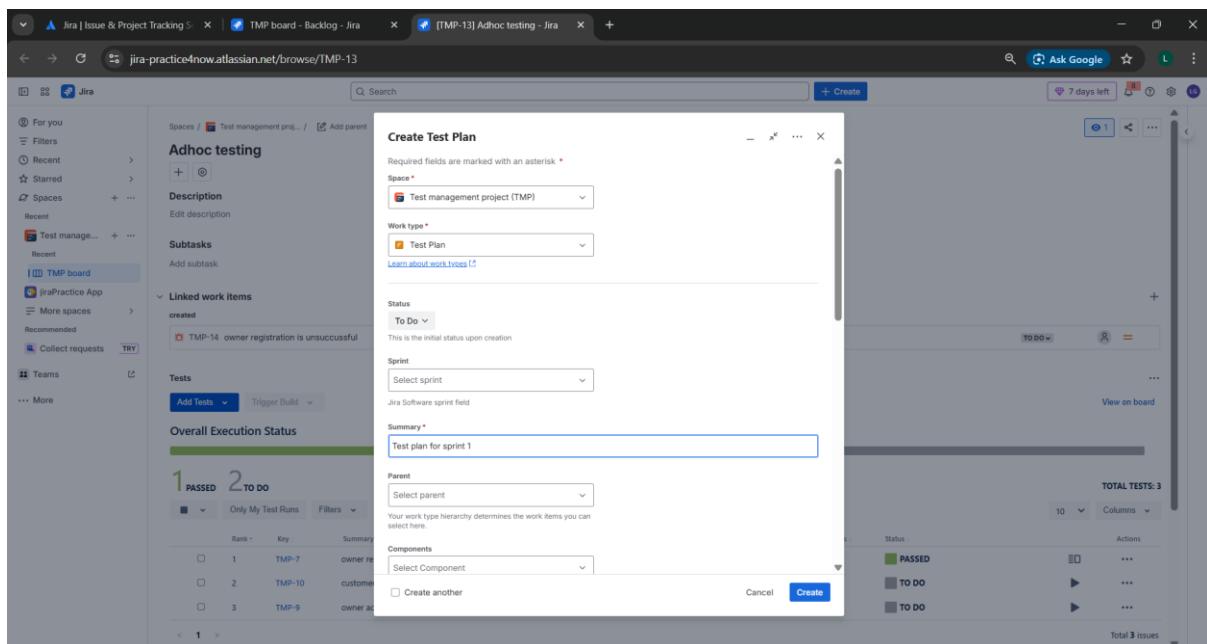


Fig: Creating a Test Plan issue

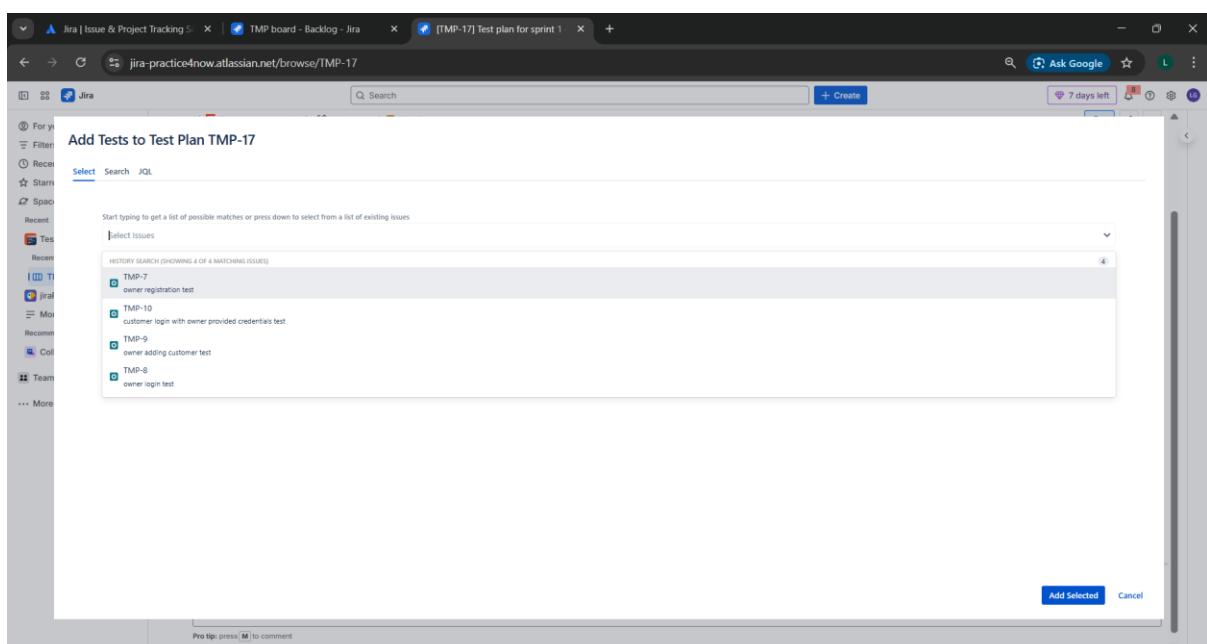


Fig: Adding existing tests to test plan

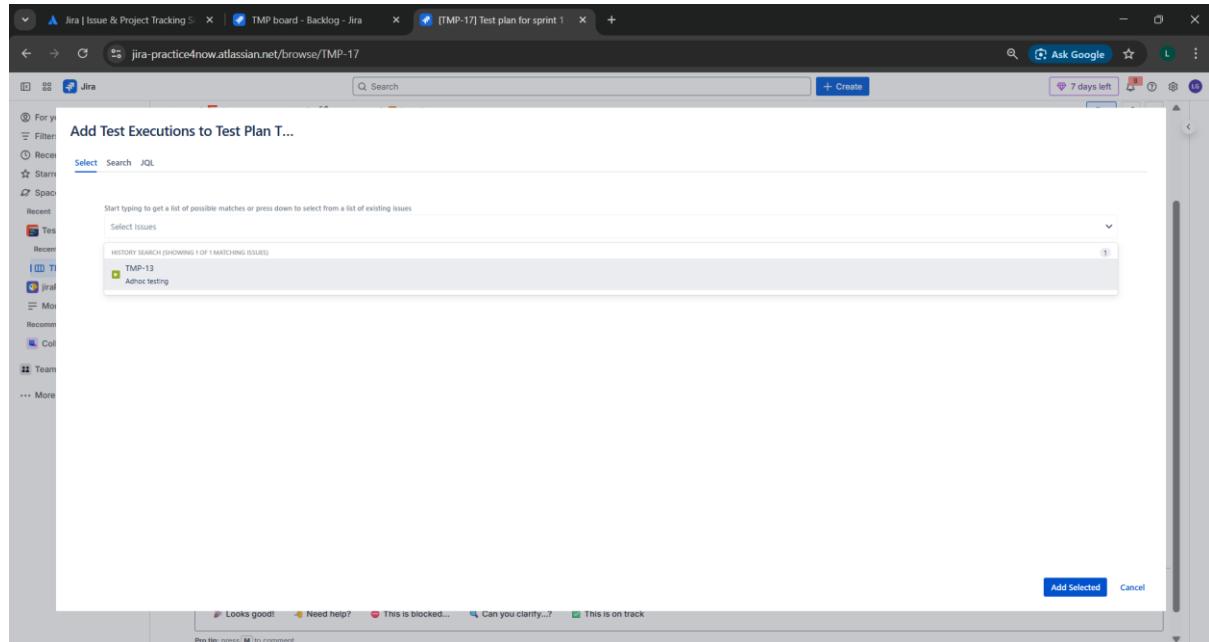


Fig: Adding Existing test executions to the test plan.

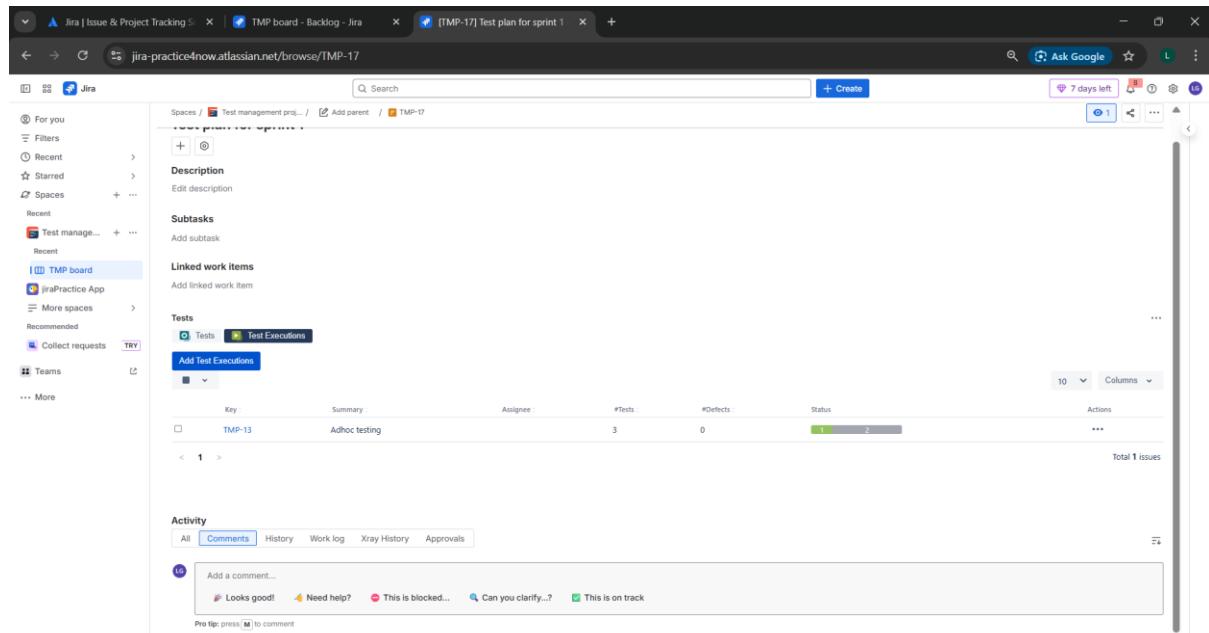


Fig: Tracking the statuses of the executing tests easily by means of a specific test plan

# Test Parameter

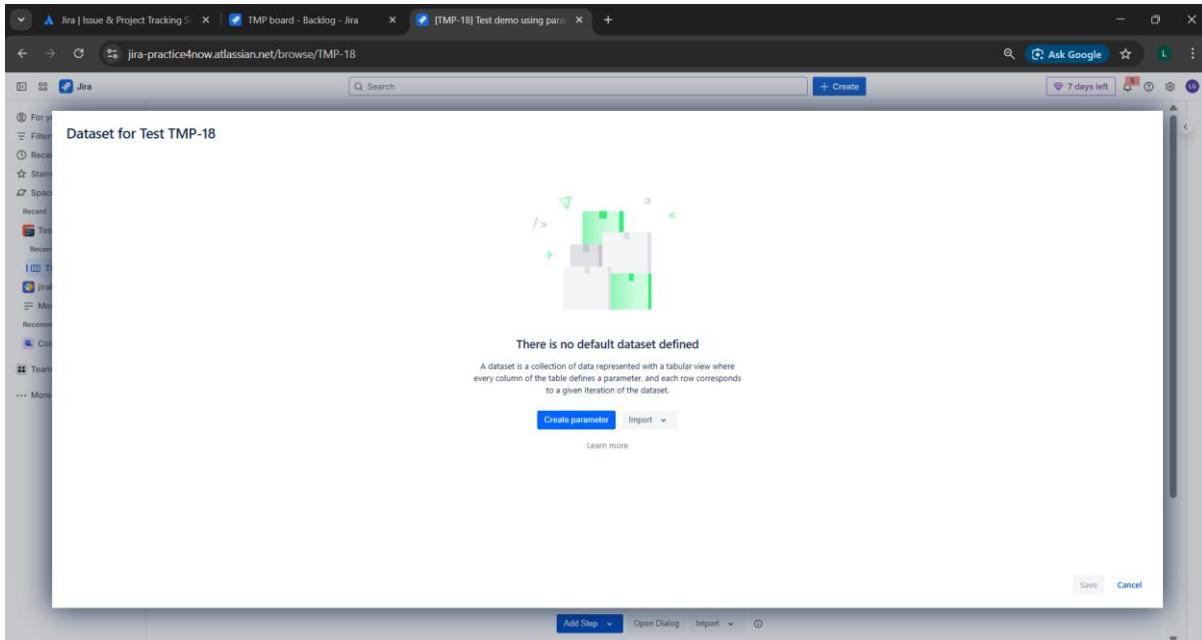


Fig: After creating a test then click on dataset to add parameter data

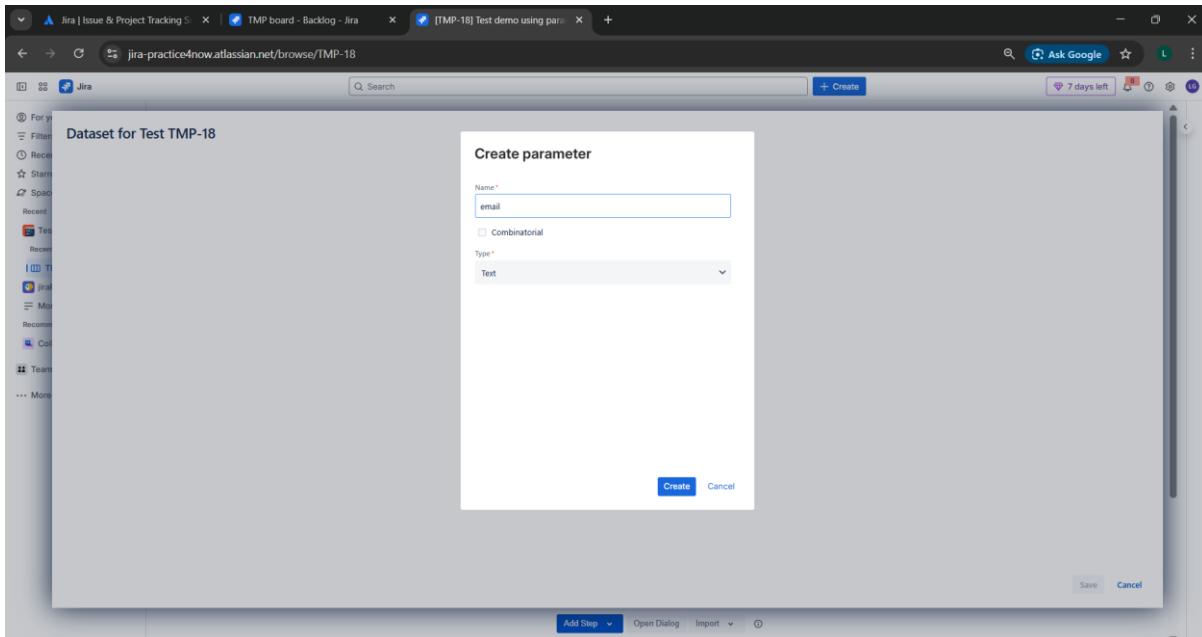


Fig: Creating Text type parameter.

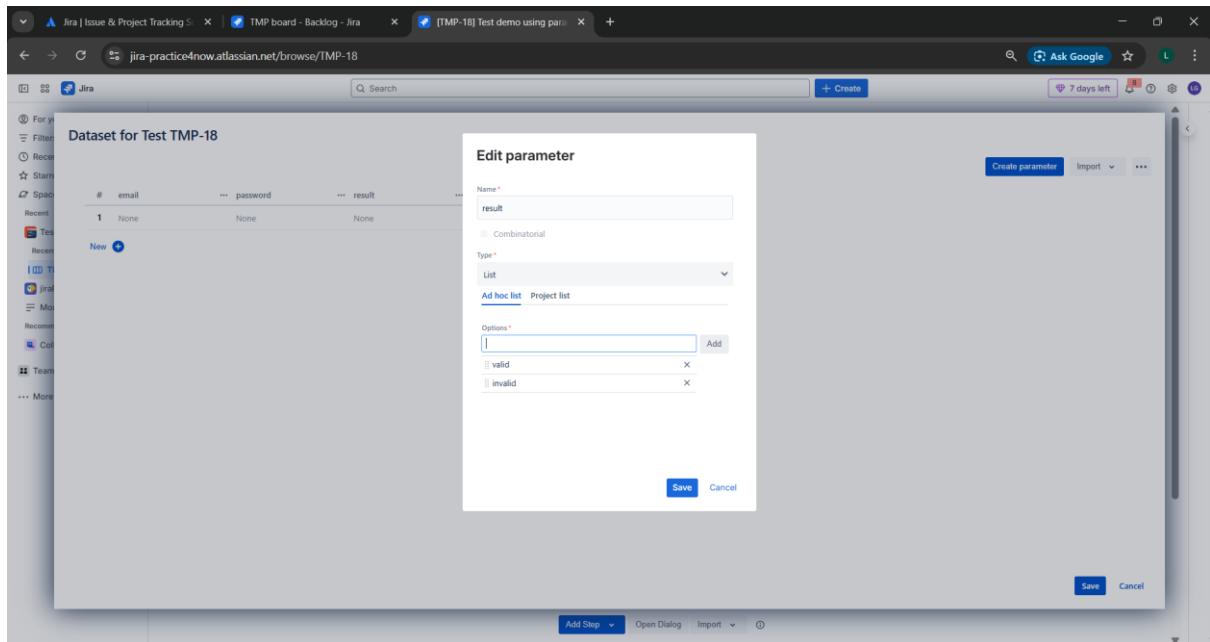


Fig: Creating List type parameter.

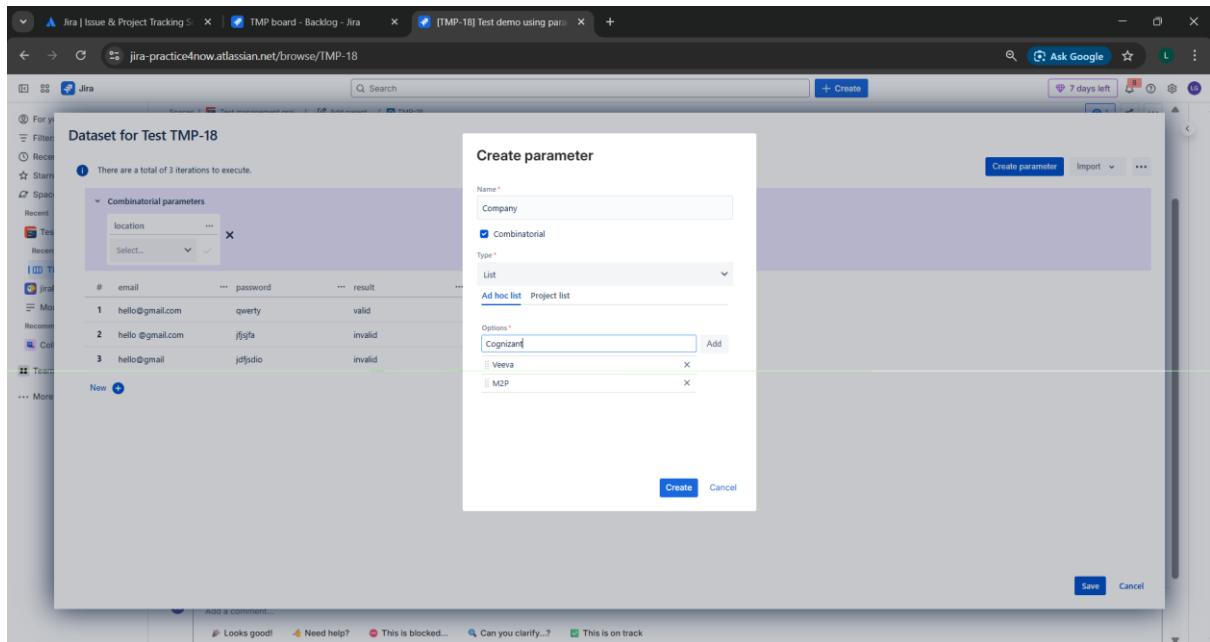


Fig: Creating Combinatorial parameter

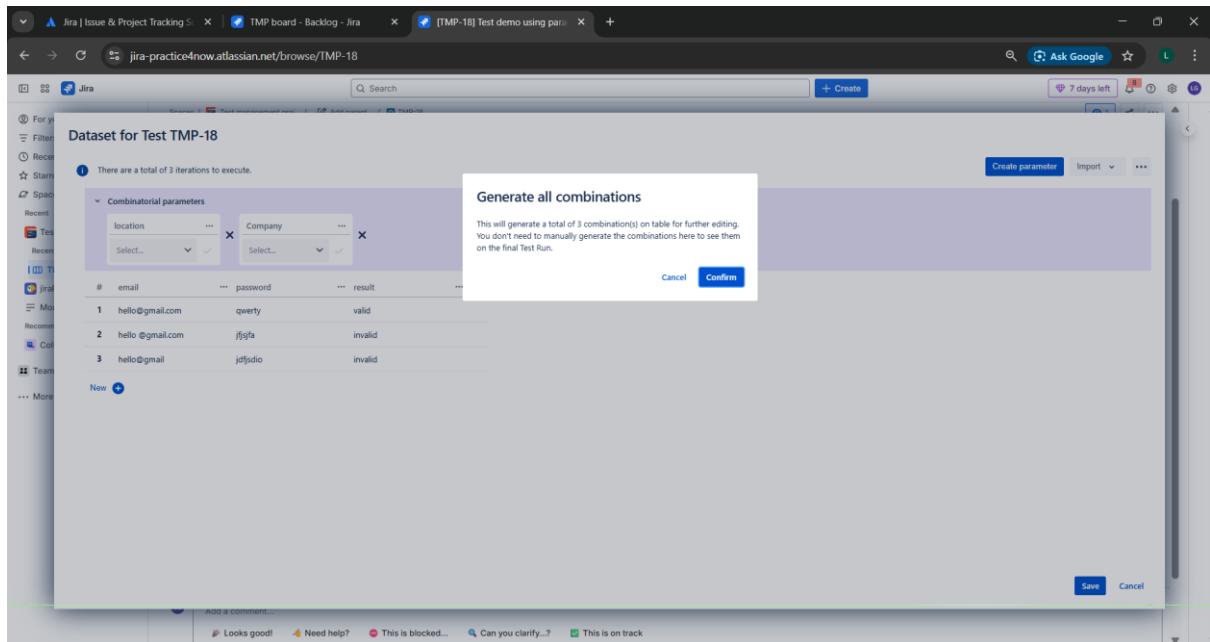


Fig: Generating all combinations of combinatorial parameters

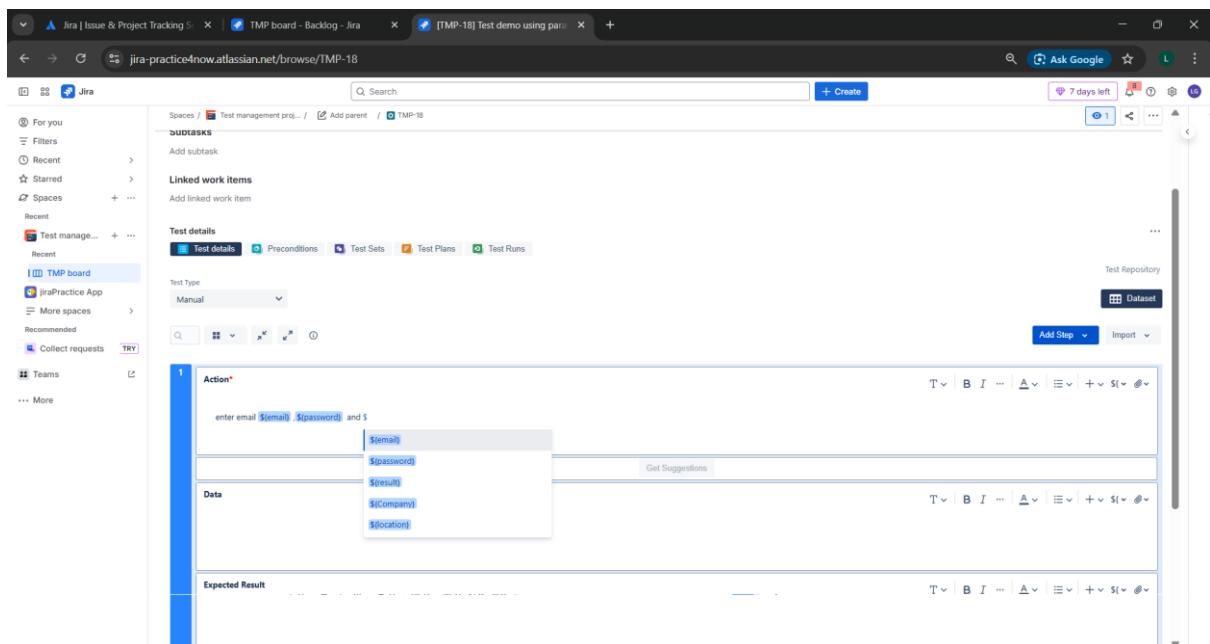


Fig: Adding Dataset related parameterized data for the step in a test