

[Xray] How to create a use a parameterized project in Jenkins

Sometimes, it is useful/necessary to turn the builds in Jenkins more flexible/dynamic. For that, in the Jenkins job, you can configure it as a parameterized project.

Example using a Freestyle project and a multipart endpoint

- Create a freestyle project
- In the job configuration page check the option "This project is parameterized" and click in the option "Add Parameters". There is a drop-down option to configure parameterized build options (like a boolean attribute, or a string attribute).
- Example:

The screenshot shows the Jenkins job configuration page for a Freestyle project. The "General" tab is selected. The checkbox "This project is parameterized" is checked and highlighted with a red box. Below it, there are two "String Parameter" sections. The first section has a "Name" field with the value "ProjectKey" and a "Description" field. The second section has a "Name" field with the value "ProjectKey" and a "Description" field. At the bottom, there is a "Save" button, an "Apply" button, and an "Add Parameter" button highlighted with a red box.

- In the previous example, I just have created two parameters that will be used to define the description of the Test Execution and in which project the new Test Execution will be created based in the parameters that will be passed when running the job.
- In the Post-Build: Xray: Results Import task, you configure where the parameters will be read and used, using the syntax `${parametername}` (`${Description}` and `${ProjectKey}`)

The screenshot shows the "Xray: Results Import Task" configuration page. The "Jira Instance" is set to "Docker2" and the "Format" is set to "JUnit XML multipart". Under the "Parameters" section, the "Import to Same Test Execution" checkbox is checked. The "Execution Report File (file path with file name)" is set to "JUnit2.xml". The "Test Execution fields" dropdown is set to "JSON Content". The JSON content is displayed in a text area, showing a JSON object with fields for "project", "summary", "issuetype", "labels", and "description". The "key" field is set to "\${ProjectKey}" and the "description" field is set to "\${Description}", both highlighted with red boxes. At the bottom, there are "Save" and "Apply" buttons.

- Save it
- Now you will notice that the "Build Now" text has changed to "Build Now with Parameters", and clicking on the link provides you the option to set those parameters for the build. Example:

Project JUnit2

This build requires parameters:

Description: Description for the parameterized project in Jenkins

ProjectKey: CALC

Build

- Console Output in Jenkins

Console

Started by user unknown or anonymous

Running as SYSTEM

Building in workspace C:\Program Files (x86)\jenkins\workspace\JUnit2

Starting XRAY: Results Import Task...

#####

Xray is importing the Feature Files

#####

Starting to import results from JUnit2.xml

Response: (200) {"testExecIssue":{"id":"12648","key":"CALC-2253","self":"http://cda-support-docker:7221/rest/api/2/issue/12648"}}

Successfully imported JUnit XML multipart results from JUnit2.xml

XRAY_ISSUES_MODIFIED: CALC-2253

XRAY_TESTS:

XRAY_TEST_EXECS: CALC-2253

XRAY_IS_REQUEST_SUCCESSFUL: true

XRAY_STATUS_RESPONSE: {"testExecIssue":{"id":"12648","key":"CALC-2253","self":"http://cda-support-docker:7221/rest/api/2/issue/12648"}}

Finished: SUCCESS

- New Test Execution created in Jira (CALC-2253)

Test Execution for Execution

Details

Type: Test Execution

Priority: Lowest

Affects Version/s: None

Component/s: None

Labels: Android, label1

Test Plan: None

Test Environments: None

Test required: Test

Status: Open (View Workflow)

Resolution: Unresolved

Fix Version/s: None

Description

Description for the parameterized project in Jenkins

Tests

Overall Execution Status

1 PASS 1 FAIL

Total Tests: 2

Rank	Key	Summary	Test Type	#Req	#Def	Test Sets	Assignee	Status
2	BMP-3	XrayUnitImport	Generic	0	0			PASS
1	BMP-2	XrayUnitImport2	Generic	0	0			FAIL

Showing 1 to 2 of 2 entries

Related Documentation

- [Integration with Jenkins](#)