

Read a project property

This function has been **renamed** with the **JWT 3.0** release.

Find the new documentation at:

Copy JWT project property

On this page

- [Purpose](#)
- [Example: Read a project property "StagingSiteURL" and write the value into "Testing Site" custom field](#)
- [Usage Examples](#)
- [Related Features](#)

Purpose

This post-function allows to read the value of a **Project Properties** and write the value into a field of the issue.

Example: Read a project property "StagingSiteURL" and write the value into "Testing Site" custom field

Project key:

Leave this field empty for reading a property in current project.

Name of project property:

Project Properties are stored in project description with the following format:
{property_name=property_value} e.g.
{MaxNumberBugs=5}

Target field to store project property value:

Conditional execution:

Optional boolean expression that should be satisfied in order to actually execute the post-function.
(Syntax Specification)

Field code injector:

Summary - [Text] - %{00000}

Field codes with format %{nnnnn} will be replaced with their corresponding value at runtime.

Field code injector:

Summary - [Text] - %{00000}

Field codes with format %{nnnnn} will be replaced with their corresponding value at runtime.

Testing Site - [URL Field]

1

Leave the field empty for executing the post-function unconditionally.

[Collection of Examples](#)

[Line 1 / Col 1]

Logical connectives: and, or and not. Alternatively you can also use &, | and !.

Comparison operators: =, !=, >, >=, < and <=. Operators in, not in, any in, none in, ~ and !~ can be used with strings, multi-valued fields and lists.

Logical literals: true and false. Literal null is used with = and != to check whether a field is initialized, e.g. {00012} != null checks whether Due Date is initialized.

String Field Code Injector:

Summary - [Text] - %{00000}

Numeric/Date Field Code Injector:

Original estimate (minutes) - [Number] - {00068}

Once configured post-function will look like this:

Conditions 0	Validators 0	Post Functions 6
The following will be processed after the transition occurs Add post function		
1. Project property StagingSiteURL of current project will be stored in field Testing Site .		

JIRA Workflow Toolbox provides an **Expression Parser**, which is used in the main features of the plugin. This parser provides a set of functions for checking the existence of a project property, and reading its value:

FUNCTION	RETURNED VALUE
projectProperty (string property_name) : string	Returns a string with the value of project property with name property_name in current issue's project. Shows an error message if project property doesn't exist. Example: projectProperty("maxNumberOfReopenings") returns "3", provided there is a string {maxNumberOfReopenings=3} in the description of current issue's project.
projectProperty (string property_name , string project_key) : string Available since version 2.2	Returns a string with the value of project property with name property_name in project with key project_key . Shows an error message if project property doesn't exist. Example: projectProperty("maxNumberOfReopenings", "CRM") returns "3", provided there is a string {maxNumberOfReopenings=3} in the description of project with key "CRM".
projectPropertyExists (string property_name) : boolean Available since version 2.2	Returns true only if there is a project property with name property_name in current issue's project, i.e., if project's description contains a string like {property_name=value} . Example: projectPropertyExists("maxNumberOfReopenings") returns true only if there is a string like {maxNumberOfReopenings=x} in the description of current issue's project.
projectPropertyExists (string property_name , string project_key) : boolean Available since version 2.2	Returns true only if there is a project property with name property_name in project with key project_key . Example: projectPropertyExists("maxNumberOfReopenings", "CRM") returns true only if there is a string like {maxNumberOfReopenings=x} in the description of project with key "CRM".

The parser is a common feature, making **Project Properties** available in the main features of the plugin:

- Condition **Boolean Condition with math, date-time or text-string terms**
- Validator **Boolean Validator with math, date-time or text-string terms**
- Post-functions **Set a field as a function of other fields**, **Copy parsed text to a field** and **Mathematical and date-time expression calculator**

Usage Examples

Page: [Using project properties to calculate custom sequence numbers](#)

Related Features

- [Check project property](#)
- [Set or create project property](#)