

Read user property

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

Find the new documentation at:

[Copy JWT user property](#)

On this page

- [Purpose](#)
- [Example: Read user property and store in virtual field](#)
- [Usage Example](#)
- [Related Features](#)

Purpose

This post-function allows to read the value of a **user property** and store the value in field of the issue.

Example: Read user property and store in virtual field


In the following screenshot we show how to read a current user's property called **"Time Logged Last Day (minutes)"**, and store the content into virtual field **"Ephemeral number 1"**:

Name of property to be read:	Time Logged Last Day (minutes){00001}
Field code injector: Summary - [Text] - %{00000} ? Field codes with format %{nnnnn} can be inserted in the field, and they will be replaced with their corresponding field value at runtime.	
Target field:	Ephemeral number 1 - [Number]
Property owner:	Current user - [User]
Additional options:	<input type="checkbox"/> Update issue immediately after field writing. A specific entry will be created in issue history for this field writing.
Conditional execution:	1
Optional boolean expression that should be satisfied in order to actually execute the post-function. (Syntax Specification)	
Leave the field empty for executing the post-function unconditionally. Collection of Examples [Line 1 / Col 1]	
<p><u>Logical connectives</u>: and, or and not. Alternatively you can also use &, and !.</p> <p><u>Comparison operators</u>: =, !=, >, >=, < and <=. Operators in, not in, any in, none in, ~ and !~ can be used with strings, multi-valued fields and lists.</p> <p><u>Logical literals</u>: 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.</p> <p>String Field Code Injector: Summary - [Text] - %{00000}</p> <p>Numeric/Date Field Code Injector: Original estimate (minutes) - [Number] - {00068}</p> <p>Check Syntax</p>	

Once configured, post-function looks like this:

The following will be processed after the transition occurs

[Add post function](#)

1. Read user property **Time Logged Last Day (minutes)** from user in **Current user** and write its value into field **Ephemeral number 1**. 

FUNCTION	RETURNED VALUE
userProperty (string propertyName , string userName) : string Available since version 2.1.34	Returns the value of the user property with name propertyName which belongs to user with user name userName . If the user doesn't have the property, null will be returned.
userProperties (string propertyName , string list userNames) : string list Available since version 2.1.34	Returns the list of values of the user property with name propertyName in all the users whose names are contained in userNames . The output will contain as many strings as users have the property set.

JIRA Workflow Toolbox provides an **Expression Parser**, which is used in many features of the plugin. This parser provides functions for reading the value of a user property.

These functions can be used in the following 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 Example

Page: [Copy the value of a user property into a user picker](#)

Related Features

- [Set or create user property](#)