

Field codes

On this page

[Overview](#) | [Field code notation](#) | [What is a context?](#) | [Available field codes](#)

One of the most important features of **JWT** is the easy accessibility to Jira data stored in **system fields**, **custom fields** and a significant number of other **virtual fields** that are made available by the **JWT** implementation.

You can **access**, **validate**, do mathematical **calculations** and **manipulate** the values found in these fields through the use of **field codes**.

A field code is a **unique identifier** (key) that can be used in any **JWT expression editor**. At the same time a field code is a safety feature that makes your expressions **immune to custom field renaming**.



Overview

Field codes will be automatically added to your expression when you insert them anywhere using the expression parser **field code injector**.

Field codes for Jira **standard** or **system fields** will display the attribute in a legible form like `%{issue.summary}`.

All **custom fields** will be notated like `%{issue.cfnnnnn}` where `nnnnn` contains the Jira **custom field ID**.

Once an expression has been saved, the real name will be displayed in the configuration element.

The purpose of using the `cfnnnnn` notation is quite simple - **custom fields can be renamed**.

Your browser does not support the HTML5 video element



Field code notation

Depending on the **context** in which they are being used, field codes will contain a **prefix** following this **notation**: `{origin.field}`

What is a context?

A context basically determines where JWT will pull data from. Available **contexts** (or **origins**) in JWT are:

Context	Example	Description
issue	<code>%{issue.description}</code>	<p>The field value will be retrieved from the issue that currently being processed by a workflow function or automation rule.</p> <p>The most commonly used context in JWT.</p>
parent	<code>%{parent.summary}</code>	<p>The field value will be retrieved from the parent of the issue that is processed by a workflow post function or an automation rule.</p> <p>Only valid for sub-tasks.</p>

seed	<pre>%{seed.issue.summary} %{seed.text} %{seed.number}</pre>	<p>The field value will be retrieved from the element currently being processed by a workflow function or automation rule that is capable of analyzing multiple elements (e.g. Create issue post function or the Create issue action).</p> <p>Elements can be:</p> <ul style="list-style-type: none"> • issues • texts • numbers <p>depending in your configuration. Read more about Seeds.</p>
action	<pre>%{action.status}</pre>	<p>The value will be retrieved from the response of the executed remote action.</p> <div>Currently only available in the Execute remote action post function.</div>
function	<pre>filterByPredicate(subtasks() , {function.issue.dueDate} != null)</pre> <div>This example returns a list of all sub-tasks of the current issue that have a due date set.</div>	<p>The field value will be retrieved from issue(s) returned by some JWT expression parser functions.</p> <p>Mostly these are functions that return an issue list (e.g. subtasks()).</p> <div>Some functions iterate (loop) over all elements returned by a list. As soon as each element of the list is being processed, this element is the temporary context.</div>
created.parent	<pre>%{created.parent.summary}</pre>	<p>The field value will be retrieved from the new parent issue of a sub-tasks being created in a workflow post function.</p> <div>Currently only available in the Create issue post function and only valid for sub-tasks.</div>

Additional contexts for automation rules

An issue is tied to a workflow. This is why most field codes in [JWT workflow functions](#) have their issue context. [Automation rules](#) are not tied to individual issues. This is why sometimes it is necessary to define their context separately.

These additional contexts are available for automation rules:

Context	Description	Example
trigger	The issue, user, version, component or project event that triggers the execution of the rule .	<pre>%{trigger.issue.description}</pre> <p>The description of the issue triggering the automation rule.</p>
selector	The issue currently being processed by a selector (e.g. an issue returned by a JQL query).	<pre>%{selector.issue.cf10021}</pre> <p>The value of the custom field with the ID 10021 from the issue currently being processed by a selector.</p>

The **prefix** is a referential part of the field code and **will be inserted into the expression** whenever you select a field from a dropdown list (as shown below).

Your browser does not support the HTML5 video element

The additional contexts can be used **in combination** with the **standard** contexts (e.g. parent, seed).

Example:

```
%{trigger.parent.summary}
```

Find a full list of automation-only field code here: [Field codes \(automation-only\)](#)

Number vs. text field code notation

Field codes must always be enclosed by **curly brackets {}** but if they are used for **text-strings**, the brackets must be preceded by a **percent sign %**.

- **Numeric fields** can be referenced as numbers using the following notation: `{issue.somenumberfield}`. (❗ no preceding % sign)
 - If a field is not set or does not return a number (e.g. a text field), it is evaluated to `null`.
- **Text fields**: Any field type or data type can be transformed to text, so **any field** can be referenced as a **text-string value** using the following notation: `%{issue.somefield}`.
 - If a field has no value (`null`), an **empty text** will be returned.
- **Cascading Select** fields are treated as text fields, where *i* is the index that represents the level to be accessed. (*i* = 0 is used for base level) are notated as `%{issue.somefield.i}`

A complete list of all available data types can be [found here](#).



Available field codes

Check out the following pages to familiarize yourself with the different **types** of field codes.

Field codes (read / write)

Field codes (read-only)

Field codes (write-only)

Field codes (automation-only)

Seeds

Field name	Field code	Type	Example output	Labels
Action response	<code>%{action.response}</code>	READ-ONLY	<code>{"self": "https://your-domain.atlassian.net/jira/rest/api/3/project/10042","id": 10010,"key": "PROJ"}</code>	
Action response details	<code>%{action.response.JMESPath}</code>	READ-ONLY	PROJ	
Action status	<code>%{action.status}</code>	READ-ONLY	200	
Add to parent's total time spent (minutes)	No field code	WRITE-ONLY		
Add to time spent (minutes)	No field code	WRITE-ONLY		
Affects version/s	<code>%{issue.versions}</code>	READ / WRITE	1.0, 1.1	
Affects version/s with details	<code>%{issue.versions.details}</code>	READ-ONLY	1.0 # First release # RELEASED ON 28/Mar/22 12:00 AM # ARCHIVED	
Assignee	<code>%{issue.assignee}</code>	READ / WRITE	admin.istrator	
Assignee's email	<code>%{issue.assignee.email}</code>	READ-ONLY	user@example.com	
Assignee's full name	<code>%{issue.assignee.displayName}</code>	READ-ONLY	Bob Smith	
Attachments	<code>%{issue.attachments}</code>	READ-ONLY	file1.txt, readme.pdf, screenshot.png	

Attachments (current attachments will be replaced)	No field code	WRITE-ONLY	
Attachments (only new attachments will be added)	No field code	WRITE-ONLY	
Attachments with details	%{issue.attachments.details}	READ-ONLY	file1.txt (text/plain, 5.14 KB), readme.pdf (application/pdf, 179.8 KB), screenshot.png (image/png, 5.449 KB)
Available target statuses	%{issue.status.achievable}	READ-ONLY	In Progress, Resolved, Closed
Available transitions	%{issue.transition.achievable}	READ-ONLY	Start Progress, Resolve Issue, Close Issue
Component/s	%{issue.components}	READ / WRITE	Web Site, Authenticator, Statistics
Component default assignee	%{trigger.component.defaultAssignee}	READ-ONLY	admin.istrator
Component description	%{trigger.component.description}	READ-ONLY	This is a component
Component ID	%{trigger.component.id}	READ-ONLY	10000
Component lead	%{trigger.component.lead}	READ-ONLY	admin.istrator
Component leads	%{issue.components.leads}	READ-ONLY	admin, d.smith
Component name	%{trigger.component.name}	READ-ONLY	Customer Relationship Management Component
Creator	%{issue.creator}	READ-ONLY	admin.istrator
Creator's email	%{issue.creator.email}	READ-ONLY	user@example.com
Creator's full name	%{issue.creator.displayName}	READ-ONLY	Bob Smith
Current date and time	%{system.currentDateTime}	READ-ONLY	19/Mar/20 1:38 PM
Current user	%{system.currentUser}	READ-ONLY	admin.istrator
Current user's email	%{system.currentUser.email}	READ ONLY	user@example.com
Current user's full name	%{system.currentUser.displayName}	READ-ONLY	Bob Smith
Customer Request Type Name	%{issue.customerRequestTypeName}	READ-ONLY	
Date and time of creation	%{issue.created}	READ-ONLY	19/Mar/20 1:38 PM
Date and time of last status change	%{issue.lastStatusChange}	READ-ONLY	19/Mar/14 1:38 PM
Date and time of last update	%{issue.updated}	READ-ONLY	19/Mar/14 1:38 PM

Date and time of resolution	%{issue.resolutionDate}	READ-ONLY	19/Mar/20 1:38 PM
Description	%{issue.description}	READ / WRITE	Take your Jira to the next level by using JWT!
Due date	%{issue.dueDate}	READ / WRITE	19/Mar/20
Environment	%{issue.environment}	READ / WRITE	Data Center
Execute transition	No field code	WRITE-ONLY	
Execute transition (delayed execution)	No field code	WRITE-ONLY	
Fix version/s	%{issue.fixVersions}	READ / WRITE	1.0, 1.1
Fix version/s with details	%{issue.fixVersions.details}	READ-ONLY	1.0 # First release # RELEASED ON 28/Mar/22 12:00 AM # ARCHIVED
Issue key	%{issue.key}	READ-ONLY	CRM-25
Issue status	%{issue.status}	READ / WRITE	Open
Issue status (delayed writing)	No field code	WRITE-ONLY	
Issue status category	%{issue.status.category}	READ-ONLY	Done
Issue type	%{issue.issueType}	READ-ONLY	Bug
Jira base URL	%{system.baseUrl}	READ-ONLY	https://www.atlassian.net/jira
Keys of linked issues	%{issue.links}	READ-ONLY	CRM-13, HR-12, SDESK-45
Keys of sub-tasks	%{issue.subtasks}	READ-ONLY	CRM-23, CRM-26, CRM-31
Labels	%{issue.labels}	READ / WRITE	web customer java mobile
Last comment	%{issue.lastComment}	READ / WRITE	Take your Jira to the next level by using xApps!
Last comment's visibility restriction	%{issue.lastComment.visibility}	READ / WRITE	jira-administrators
Last commenter	%{issue.lastComment.author}	READ-ONLY	admin.istrator
New comment	No field code	WRITE-ONLY	
New comment (sends email notifications)	No field code	WRITE-ONLY	
New labels	No field code	WRITE-ONLY	
New watchers	No field code	WRITE-ONLY	
Number of affects version/s	{issue.versions.count}	READ-ONLY	42
Number of attachments	{issue.attachments.count}	READ-ONLY	3

Number of fix version/s	{issue.fixVersions.count}	READ-ONLY	2
Number of labels	{issue.labels.count}	READ-ONLY	3
Number of linked issues	{issue.links.count}	READ-ONLY	3
Number of sub-tasks	{issue.subtasks.count}	READ-ONLY	3
Number of votes received	{issue.votes}	READ-ONLY	3
Original estimate (minutes)	{issue.originalEstimate}	READ / WRITE	120
Previous issue status	%{issue.status.previous}	READ-ONLY	Closed
Previous issue status category	%{issue.status.previousCategory}	READ-ONLY	Done
Priority	%{issue.priority}	READ / WRITE	Blocker
Project category	%{issue.project.category}	READ-ONLY	Archive
Project category (trigger)	%{trigger.project.category}	READ-ONLY	Archive
Project description	%{issue.project.description}	READ-ONLY	This is the CRM project.
Project description (trigger)	%{trigger.project.description}	READ-ONLY	This is the CRM project
Project ID	%{issue.project.id}	READ-ONLY	10001
Project key	%{issue.project.key}	READ-ONLY	CRM
Project key (trigger)	%{trigger.project.key}	READ-ONLY	CRM, HR, SDESK
Project lead	%{issue.project.lead}	READ-ONLY	admin.istrator
Project lead's email	%{issue.project.leadEmail}	READ-ONLY	user@example.com
Project lead's email (trigger)	%{trigger.project.leadEmail}	READ-ONLY	user@example.com
Project lead's full name	%{issue.project.lead.displayName}	READ-ONLY	Bob Smith
Project lead's full name (trigger)	%{trigger.project.leadDisplayName}	READ-ONLY	Bob Smith
Project lead (trigger)	%{trigger.project.lead}	READ-ONLY	admin.istrator
Project name	%{issue.project.name}	READ-ONLY	Customer Relationship Management
Project name (trigger)	%{trigger.project.name}	READ-ONLY	Customer Relationship Management
Project type	%{issue.project.type}	READ-ONLY	Service

Project URL	%{issue.project.url}	READ-ONLY	https://www.decadis.de/x
Project URL (trigger)	%{trigger.project.url}	READ-ONLY	https://www.decadis.de/x
Remaining estimate (minutes)	{issue.remainingEstimate}	READ / WRITE	120
Remaining issues in project	%{issue.remainingIssuesInProject}	READ-ONLY	CRM-1, CRM-2, CRM-3, CRM-4
Remote links	%{issue.remoteLinks}	READ-ONLY	https://www.first-example.com , https://www.second-example.com
Reporter	%{issue.reporter}	READ / WRITE	admin.istrator
Reporter's email	%{issue.reporter.email}	READ-ONLY	user@example.com
Reporter's full name	%{issue.reporter.displayName}	READ-ONLY	Bob Smith
Resolution	%{issue.resolution}	READ / WRITE	Resolved
Security level	%{issue.securityLevel}	READ / WRITE	Classified
Sprint completion date	%{issue.sprintCompletionDate}	READ-ONLY	2020-02-17 07:46
Sprint end date	%{issue.sprintEndDate}	READ-ONLY	2020-02-17 07:46
Sprint ID	%{issue.sprintId}	READ-ONLY	1
Sprint start date	%{issue.sprintStartDate}	READ-ONLY	2020-06-08 12:10
Summary	%{issue.summary}	READ / WRITE	This is my summary
Team name	%{issue.teamName}	READ-ONLY	Team HR
Temporary number	{issue.temporaryNumber1} ... {issue.temporaryNumber5}	READ / WRITE	10
Temporary text	%{issue.temporaryText1} ... %{issue.temporaryText5}	READ / WRITE	This is a text stored temporarily
Total time spent (minutes)	{issue.timeSpent}	READ / WRITE	120
User directory ID	%{trigger.user.directoryId}	READ-ONLY	10001
User directory user	%{trigger.user.directoryUser}	READ-ONLY	admin.istrator
User email	%{trigger.user.email}	READ-ONLY	user@example.com
User full name	%{trigger.user.fullName}	READ-ONLY	Bob Smith
User ID	%{trigger.user.id}	READ-ONLY	10000

User is active	<code>%{trigger.user.isActive}</code>	READ-ONLY	true, false
User key	<code>%{trigger.user.key}</code>	READ-ONLY	admin.istrator
User name	<code>%{trigger.user.name}</code>	READ-ONLY	admin.istrator
Version archived	<code>%{trigger.version.archived}</code>	READ-ONLY	true, false
Version description	<code>%{trigger.version.description}</code>	READ-ONLY	This is version 1.0
Version ID	<code>%{trigger.version.id}</code>	READ-ONLY	10000
Version name	<code>%{trigger.version.name}</code>	READ-ONLY	Version 1.0
Version project	<code>%{trigger.version.project}</code>	READ-ONLY	CRM
Version project ID	<code>%{trigger.version.projectId}</code>	READ-ONLY	10000
Version released	<code>%{trigger.version.released}</code>	READ-ONLY	true, false
Version release date	<code>%{trigger.version.releaseDate}</code>	READ-ONLY	2020-02-17 07:46
Version sequence	<code>%{trigger.version.sequence}</code>	READ-ONLY	1
Version start date	<code>%{trigger.version.startDate}</code>	READ-ONLY	2020-02-17 07:46
Watchers	<code>%{issue.watcher}</code>	READ / WRITE	username1, username2, username3
Workflow scheme	<code>%{issue.workflowScheme}</code>	READ-ONLY	Global scheme

If you still have questions, feel free to refer to our [support](#) team.