

Prevent creation of issues with a duplicate summary

On this page

[Configuration](#) | [Screenshots](#) | [Related use cases](#)

Use case

This use case ensures, that users **cannot create** new issues if an issue of the same **type** and **summary** already exists in the **project**.



Configuration

1

Add the [Validation based on JQL query](#) to the **Create** transition of the desired workflow.

2

JQL

```
project = %{issue.project.key} AND type = %{issue.issueType} and summary ~ %{issue.summary}
```

3

Validation

Choose **Number of issues returned by the JQL query satisfies a logical expression**

4

Expression

```
#the number of issues returned by the query is temporarily stored in the field  
temporaryNumber1  
{issue.temporaryNumber1} = 0
```

5

Error message

```
There already exists a %{issue.issueType} with the summary: %{issue.summary}
```

Error message with displaying issue key of existing issue

You can also display the **issue key** of the already existing issue so that you can directly navigate to that issue.

Make sure that the parsing mode is set to **Advanced**.

```
"There already exists a " + %{issue.issueType} + " with the summary \" + %{issue.summary} +  
\"\": " + first(fieldValue(%{issue.key}, issuesFromJQL("project = " + %{issue.project.key} + "  
AND type = " + %{issue.issueType} + " and summary ~ \" + %{issue.summary} + "\""))
```

This might not work for issues where the summary contains **special characters** since the default search will be performed as a **fuzzy search**. You can perform an **exact search**, however, if you use the following syntax: `"\"%{issue.summary}\""`

```
project = %{issue.project.key} AND type = %{issue.issueType} and summary ~ "\"%{issue.summary}\""
```



Screenshots

[Update parameters of the 'Validation based on JQL query' Validator for this transition.](#)

Update parameters of the 'Validation based on JQL query' Validator for this transition.

Basic configuration *

Define the required parameters for this validator.

JQL *

```
1 project = %{issue.project.key} AND type = %{issue.issueType} and  
summary ~ "\"%{issue.summary}\""
```

Enter a JQL query and optionally use **field codes**, e.g. `%{issue.summary}`, to insert field values.

Validation *

Number of issues returned by the JQL query satisfies a logical expression

Expression *

```
1 (issue.temporaryNumber1) = 0
```

The **Temporary number 1** field code returns the number of issues returned by the JQL query. The expression must be a **logical expression** that returns **true** or **false**.

> Validation options

Error message

Add an **optional** error message to show when a validation fails.

Location

Show on top

Define where the error message should be displayed.

Message

```
1 There already exists a %{issue.issueType} with the summary:  
%{issue.summary}
```

Enter plain text and optionally use **field codes**, e.g. `%{issue.summary}`, to insert field values.

Add translation

Add **optional** error message translations. Translations will be displayed based on the user's language setting.

> Run as

Validation based on JQL query

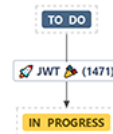


Documentation

Evaluate issues against the output of a JQL query.

Workflow

JWT 3.0.0 Use cases



Related use cases

Use case	JWT feature	Workflow function	Parser functions	Label
Assignee may only have a restricted number of assigned issues		Condition based on JQL query Validation based on JQL query		
Prevent creation of issues with a duplicate summary		Validation based on JQL query		

Prevent issue creation with the same field value



Validation based on JQL
query

STAFF PICK

Logical validator

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