

Prevent setting due dates outside business hours

On this page

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

Use case

Setting a due date of an issue to a date outside of the company's business hours can cause confusions.

In the following use case the transition will be prevented if the issue's due date is not set accordingly.



Configuration

Assumption: A custom JWT calendar called **"my_calendar"** has been defined as follows:

```
MON - THU {  
  08:00 - 15:00,  
  16:00 - 19:30;  
}  
  
FRI {  
  08:00 - 15:00;  
}
```

1

Add the [Logical validator](#) to the desired workflow transition. You might want to place it within the create transition to prevent the creation of the issue.

2

Expression

Add the following expression:

```
withinCalendar({issue.dueDate}, "my_calendar", LOCAL)
```

3

Error message

Optionally, enter an error message something like this:

```
Unfortunately you're trying to execute this transition outside of our business hours.  
Please try it again on:  
Mon - Thu 08:00-15:00, 16:00-19:30  
Fri 08:00-15:00.
```



Screenshots

Update parameters of the 'Logical validator' Validator for this transition.

Update parameters of the 'Logical validator' Validator for this transition.

Basic configuration *

Define the required parameters for this validator.

Expression *

```
1 withinCalendar(issue.dueDate, "my_calendar", LOCAL)
```

The expression must return `true` or `false`. The condition is met if the `logical expression` returns `true`.

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 Unfortunately you're trying to execute this transition outside of
2 our business hours.
3 Please try it again on:
4 Mon - Thu 08:00-15:00, 16:00-19:30
5 Fri 08:00-15:00.
```

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.

Logical validator

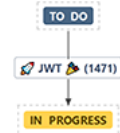
Documentation



Evaluate a logical expression that returns true or false - a most universal and powerful function to compare field values across any kind of issue relation, count sub-tasks or links.

Workflow

JWT 3.0.0 Use cases



Related use cases

Use case	JWT feature	Workflow function	Parser functions	Label
Block an Epic's transition depending on linked issues status and due date	 	Logical validator Logical condition	count() filterByPredicate() linkedIssues()	
Block an Epic's transition until all the issues under that Epic are resolved		Logical validator Logical condition	count() issuesUnderEpic() filterByResolution()	
Block a transition until all sub-tasks have certain fields populated	 	Logical validator Logical condition	count() filterByPredicate() subtasks()	STAFF PICK
Ensure that all issues linked with a certain issue link type have "Due Date" field set	 	Logical validator Logical condition	count() linkedIssues() fieldValue()	
Ensure that an issue has at least one attachment		Logical validator Logical condition		STAFF PICK

Evaluate Assets objects	 	Logical validator Logical condition	findPattern() findReplaceAll() replaceAll() toStringList()
Halt a transition if an element is not contained in a list	 	Logical validator Logical condition	toStringList()
Hide transition to a previous status		Logical condition	
Make "Time spent" field required		Logical validator	
Prevent a closed issue from being reopened after resting 1 week closed	 	Logical validator Logical condition	isInRole()
Prevent a transition depending of the number of components	 	Logical validator Logical condition	numberOfSelectedItem()
Prevent external users from creating issues		Logical validator	matches() toString() textOnStringList() toStringList() userDisplayName()
Prevent issue creation if description contains less than 100 characters		Logical validator	replaceAll() length()
Prevent issue creation with the same field value (Boolean)		Logical validator Validation based on JQL query	count() issuesFromJQL()
Prevent setting due dates outside business hours		Logical validator Logical condition	withinCalendar()
Prevent the creation of sub-tasks unless the parent issue is in a certain status		Logical validator	
Reject duplicated file names in attachments	 	Logical validator Logical condition	count() toStringList() distinct()
Restrict issue creation per issue type and project role		Logical validator	isInRole()
Set a condition in a global transition which only applies in a certain status		Logical condition	
Validate a Select List (cascading) custom field		Logical validator Logical condition	

STAFF PICK

Validate only issue links created in transition screen		Logical validator	<code>count()</code> <code>transitionLinkedIssues()</code> <code>filterByProject()</code> <code>filterByIssueType()</code>
Validation based on the value of a date type project property		Logical validator Logical condition	<code>stringToDate()</code> <code>projectProperty()</code>

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