

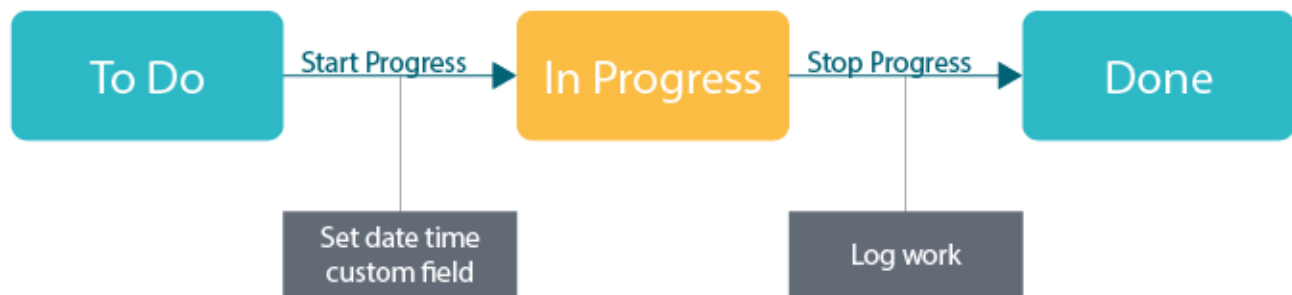
Automatically log work spent in a specific status

On this page

[Prerequisites](#) | [Configuration](#) | [Alternative implementation](#) | [Screenshots](#) | [Related use cases](#)

Use case

In this use case JWT is going to automatically log work as soon as an issue enters and leaves a specific status.



As soon as a user executes the **Start Progress** transition, we're starting our work. As soon as the user executes the **Stop Progress** transition, we stop our work and automatically log the time we worked on the issue.



Prerequisites

In order to make this use case work, we need to create one custom fields to store the time we entered the specific status.

Set up the custom field

1

Create a date time picker custom field

Name: Work Start Time

2

Go to the **Field configuration** and set the visibility of the created custom field to **hidden**.



Configuration

Set up the Start Progress transition



1

Add the [Update or copy field values](#) post function to the **Start Progress** transition.

2

Target issue

Choose **Current issue**

3

Fields

1

Field

Choose **Work Start Time**

2

Value

Choose **Set field value manually**

3

Expression

Enter the following expression:

```
{system.currentDateTime}
```

Set up the Stop Progress transition

1

Add the [Log work](#) post function to the **Stop Progress** transition.

2

Target issue

Choose **Current issue**

3

Time spent

Enter the following expression:

```
(({system.currentDateTime} - {issue.cfnnnnn}) / {MINUTE})
```

cfnnnnn is the field code for the **Work Start Time** custom field we've previously created.

Update the field code to the corresponding field code on your instance!

4

Starting date-time

Enter the following expression:

```
{issue.cfnnnnn}
```

cfnnnnn is the field code for the **Work Start Time** custom field we've previously created.

Update the field code to the corresponding field code on your instance!

5

Work description

You can enter a comment like this:

```
Work started at %{issue.cfnnnnn} and stopped at %{system.currentDateTime}.
```

cfnnnnn is the field code for the **Work Start Time** custom field we've previously created.

Update the field code to the corresponding field code on your instance!

6

Additional options

1

Remaining estimate

Choose **Auto adjustment**

+ Alternative implementation

In the current implementation, the time between entering and leaving the **In Progress** status is being tracked as working time. So **also non-working time will be logged** (e.g. from 5pm to 8am the other day) even though nobody might be working on the ticket.

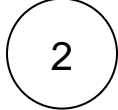
For this, a **JWT calendar** could be used. This way **only the actual work time will be considered**.

Create a new calendar

Head over to the Jira Administration [Manage apps](#) [Jira Workflow Toolbox](#) **Calendars**.

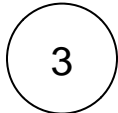


1



2

Hit **Add calendar** in the upper right corner.



3

Name

Enter an appropriate name. For this use case we're going to use the name **working-hours**.



4

Specification

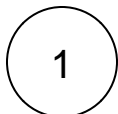
Enter the following schedule:

```
MON-FRI{  
    08:00-17:00;  
}
```

Update the Log work configuration from above

Once the calendar is created, we have to update the configuration of the Log Work post function from above.

We only need to update the **Time spent** expression.



1

Time spent

Enter the following expression:

```
timeDifference({system.currentDateTime}, {issue.cfnnnnn}, "working-hours", LOCAL) / {MINUTE}
```

cfnnnnn is the field code for the **Work Start Time** custom field we've previously created.

Update the field code to the corresponding field code on your instance!

For this expression the [timeDifference\(\)](#) function was used.



Screenshots

Update parameters of the Update or copy field values Function for this transition.

Update parameters of the Update or copy field values Function for this transition.

Basic configuration *

Define the required parameters needed to execute this post function.

Target issue *

Current issue

Select the issues to be updated.

Fields *

Define the fields to be updated.

Field	Value	Write protection
Work Start Time (13200)	<div><div>✓</div><div>123</div><div>{system.currentDateTim ...}</div></div>	<div><div></div><div></div></div>
<div>Add field</div>		

- > Additional options
- > Conditional execution
- > Run as

Update or copy field values



Documentation

Update one or multiple issue fields or copy field values across the same issue or related issues (parent / child, Epic / Story, linked issues and many more).

Workflow

JWT 3.0.0 Use cases



Examples

Pick one of the following example configurations and get started in no time. More details will be provided upon your selection.

- Assign issue to current user
- Keep parent's priority in sync
- Assign important issues to the project lead

Update parameters of the Log work Function for this transition.

Update parameters of the Log work Function for this transition.

Basic configuration *

Define the required parameters needed to execute this post function.

Target issue *

Current issue

Select the issues to log work on.

Time spent *

✓

123

1

timeDifference({system.currentDateTime}, {issue.cf13200}, "working-hours", LOCAL) / {MINUTE}

The expression must return the time to be logged in minutes.

Starting date-time

✓

123

1

{issue.cf13200}

The expression must return the starting date-time of the work log entry. In case no expression is entered, current date-time is used.

Work description

✓

123

1

Work started at %{issue.cf13200} and stopped at %{system.currentDateTime}.

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

Additional options

Remaining estimate

Auto adjustment

- > Conditional execution
- > Run as

Log work



Documentation

Log work on one or multiple issues.

Workflow

JWT 3.0.0 Use cases



Related use cases

Use case	JWT feature	Workflow function	Parser functions	Label
Add percentaged profit margin when closing issue		Log work		

Automatically log work spent in a specific status		Log work	<code>timeDifference()</code>
Log absence time on another issue		Log work	<code>stringToDate()</code>

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