

# Create issues randomly

## On this page

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

### Use case

In this use case, we will create randomly sub-tasks for specific issues.

Use the following use case to create random issues with your desired specifications!



### Configuration

1

Add a [Create issue](#) post function to the **workflow transition** you want.

2

Issues to be created

1

#### Mode

Choose **Single issue**

2

#### Issue type

Choose **Selected issue type** **Sub-task**

3

#### Parent issue

Choose **Current issue**

The parameters above can be changed to your specific needs, check the related use cases for inspiration.

3

### Summary

Enter a summary according to your needs, like

weekly test

4

### Description

To enrich the issue with further information, you might want to add a short description. It is not mandatory, though.

---

## Further configuration

---

### Fields

You have the possibility to set additional fields on the issue to be created.

Maybe you want to predefine the issue **Assignee** or a **time estimation** for time logging purposes.

Don't want to set the fields? You can also copy existing values from the current issue!

For further information, you might want to take a look at the [Create issue documentation](#).

---

### Issue links

This is an optional parameter where you can define issue links for the issue to be created.

---

### Additional options

If you want to copy issue keys of created issues into the field **Temporary Text 3** as a text list, choose this option.

---

### Conditional execution

To have the issues created randomly, enter the following expression:

```
modulus(round({system.currentDateTime}/random()*10),3) = 0
```

You can read more about the [JWT expression parser functions](#) used in the expression above:

- [modulus\(\)](#)
- [round\(\)](#)
- [random\(\)](#)

---

### Run as

Select which **user** will be used to execute this post function. By default this parameter is set to the **current user**. You can also use field codes to run the function as a dynamic user (e.g. current assignee).

Make sure that the user running the post function has all the **relevant permissions** to perform the actions defined in the configuration (e.g. "Update Issues")!

If you want to keep track the actions being performed automatically, we suggest to create a **dedicated JWT account**, granted all relevant **permissions**, and use it in the Run as parameter to identify which changes have been made with JWT.

---



## Screenshots



## Related use cases

| Use case  | JWT feature | Workflow function | Parser functions  | Label      |
|---|-------------|-------------------|---|------------|
| Clone epic, tasks and sub-tasks                               |             | Create issue      | toString() toStringList() issueKeysToIssueList() replaceFirst() first() fieldValue() textOnStringList() findModify() subtasks() issuesUnderEpic() |            |
| Create an issue in the current project                        |             | Create issue      |   | STAFF PICK |
| Create an issue with a custom summary                         |             | Create issue      | previousValue()   |            |
| Create a sub-task for each component                          |             | Create issue      | toStringList()  |            |
| Create a sub-task for every recurring deadline within a task  |             | Create issue      | addMonths() getMatchingValue() dateToString()   |            |
| Create a sub-task for every sub-task closed                   |             | Create issue      | findReplaceFirst() first() findPattern() toString() sum() toNumber()  |            |
| Create a sub-task for every user in a field and assign them   |             | Create issue      | getMatchingValue() toStringList()   |            |
| Create child issue within Advanced Roadmaps hierarchy         |             | Create issue      |   |            |
| Create issues randomly  |             | Create issue      | modulus() round() random()  |            |
| Create issue under epic                                       |             | Create issue      | toString() epic()   |            |
| Create several issues combining fields                        |             | Create issue      | nthElement() toStringList() modulus() count() ceil()  |            |
| Create several sub-tasks depending on the component           |             | Create issue      |   |            |
| Create sub-tasks depending on selected values in custom field |             | Create issue      | toStringList()  | STAFF PICK |
| Create three issues with individual summaries                 |             | Create issue      | getMatchingValue()  |            |

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

