

Clone epic, tasks and sub-tasks

On this page

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

Use case

Clone an epic, the tasks under the epic and the sub-tasks of these tasks.



Prerequisites

Set up a custom field

1

Create a **Text (single line)** custom field

Name: Template Issue



Configuration

Add the first Create issue post function

1

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

2

Issue to be created

1

Mode

Choose **Single issue**

2

Issue type

Choose **Parser expression (standard issue type)** and write the following expression:

```
%{issue.issueType}
```

Summary



Enter the following expression with parsing mode set to **Basic text**

CLONE - `%{issue.summary}`

3

Additional options

Update the following fields

1

Epic Name

Choose **Parsed text (basic mode)** and write the following expression:

CLONE - `%{issue.summary}`

2

Resolution

Choose **Parsed text (advanced mode)** and write the following expression:

null

4

Copy remaining fields

Select **From current issue** and **All remaining fields**

5

Issue links

Add a **Relates to** issue link and link it to the **current issue**

6

Additional options

Tick **Store issue keys**

Add a second Create issue post function

1

Add another **Create issue** post function **after the previous post function** in the same **workflow transition**.



2

Issues to be created

1

Mode

Choose **Multiple issues**

2

Expression

Enter the following parser expression with parsing mode set to **Issue list**

```
issuesUnderEpic()
```

3

Issue type

Choose **Parser expression (standard issue type)** and use the following expression:

```
^%{issue.issueType}
```

4

Summary

Enter the following expression with parsing mode set to **Advanced text**

```
^%{issue.summary}
```

5

Description

Enter the following expression with parsing mode set to **Advanced text**

```
^%{issue.description}
```

3

Additional options

Update the following fields

1

Resolution

Choose **Parsed text (advanced mode)** and write the following expression:

```
null
```

2

Epic link

Choose **Field in current issue** and **Ephemeral String 3**

3

Template issue

Choose **Parsed text (advanced mode)** and write the following expression:

```
^%{issue.key}
```

4

Copy remaining fields

Select **From seed issue** and **All remaining fields**

5

Additional options

Tick **Store issue keys**

Add a third Create issue post function

1

Add another [Create issue](#) post function **after the previous post function** in the same **workflow transition**.

2

Issues to be created

1

Mode

Choose **Multiple issues**

2

Expression

Enter the following parser expression with parsing mode set to **Text list**

```
toStringList(toString(textOnStringList(toStringList(replaceFirst(%{00063}, "^([^\s]+,\s)", ""), ","), toString(textOnStringList(fieldValue(%{issue.cfTemplateIssue}, issueKeysToIssueList(^0%)), findModify(toString(subtasks(^1%)), "[^\s]+", ^0% + ":" + ^2%))))), ",")
```

Replace the field code `%{issue.cfTemplateIssue}` with the field code of the Template Issue field in your instance.

3

Issue type

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

4

Parent issue

Select **Set parent manually (parser expression)** and use the following expression.

```
replaceFirst(^%, ".*$", "")
```

5

Summary

Enter the following expression with parsing mode set to **Advanced text**

```
toString(fieldValue(%{00000}, issueKeysToIssueList(replaceFirst(^%, "^.*:", "", ""))))
```

6

Description

Enter the following expression with parsing mode set to **Advanced text**

```
toString(fieldValue(%{00001}, issueKeysToIssueList(replaceFirst(^%, "^.*:", "", ""))))
```

3

Additional options

Update the following fields

1

Assignee

Choose **Parsed text (advanced mode)** and write the following expression:

```
toString(fieldValue(%{00003}, issueKeysToIssueList(replaceFirst(^%, "^.*:", "", ""))))
```

2

Due date

Choose **Parsed text (advanced mode)** and write the following expression:

```
first(fieldValue({00012}, issueKeysToIssueList(replaceFirst(^%, "^.*:", ""))))
```

In this case we cannot use the **Inherit field values** parameter in the post-function for the creation of sub-tasks. Instead, we must use a different expression for each field that we want to copy into sub-task's clones. The expressions to be used are the following ones.

For **Number** and **Date-Time** fields:

```
first(fieldValue({issue.cfTemplateIssue}, issueKeysToIssueList(replaceFirst(^%, "^.*:","")))))
```

For the **rest of the fields**:

```
toString(fieldValue(%{issue.cfTemplateIssue}, issueKeysToIssueList(replaceFirst(^%, "^.*:","")))))
```

Replace the field code `%{issue.cfTemplateIssue}` with the field code of the Template Issue field in your instance.



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()
Create three issues with individual summaries		Create issue	getMatchingValue()

STAFF PICK

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