Create issue

This post function automatically creates one or multiple **standard issues** or **sub-tasks** in any given **project**.

The issue(s) that will be created can be entirely customized - from dynamically determining the **issue type** to **presetting** or **inheriting field values**.

Configuration

Mode

The **mode** parameter defines **how many** issues you want to create. The following modes are available:

Option	Description
Single issue	If you select this mode, you will be dynamically guided to ultimately specify what **issue type** will be created and **where** it will be created. This may include a multi-layered selection process. You can either pick the issue type from a **list** of **available issue types** or manually define the type using a parser expression. Depending on each selection, additional parameters will be shown. These parameters depend on the context or your previous selection - for example: • a **parent issue** needs to be chosen, when a sub-task is being created • a **project** needs to be selected, in which the issue will be created In this mode, **only one single issue will be created**.
Multiple issues	If you want multiple issues to be created at once, you will be given various options to specify • **how many** issues will be created • which **issue types** they will have and • **where** they will be created Determining the **number** of issues is done using a parser expression. The output will depend on the selected parsing mode. The following options are available: • **Number**: The expression must return a **number** that defines the number of issues to be created. The most obvious way to achieve this is by simply adding a number. • **JQL**: The number of issues returned by the **JQL query** defines the number of issues to be created. • **Issue list**: The number of list elements defines the number of issues to be created. • **Text list**: The number of list elements defines the number of issues to be created. When using an **expression** to set the parameter, make sure that the **output** returns a **valid value**. If you are unsure of the result, make sure to **try the expression** using the **quick preview** in the app.

Issue type

Select the issue type of issue(s) to be created. The following options are available:

Option	Description
Selected issue type	Select the **issue type** for the issue to be created. Depending on the selection **additional** parameter will be available. In case you choose a **standard issue type**, the project you want to create the issue in has to be defined. The following options are available: • Current project • Selected project • Seed issue's project (**only available** if multiple issues were selected and JQL mode or Issue list mode was used to set the value) • Set project manually (parser expression) If you want to dynamically set the project using an expression, make sure it returns a **project name** or **project key**. In case you choose a **sub-task**, the respective **parent** has to be selected and the following options are available: • Current issue • Parent of current issue • Seed issue (**only available** if multiple issues were selected and JQL mode or Issue list mode was used to set the value) • Set parent manually (Parser expression) If you want to dynamically set the parent using an expression, make sure it returns a **single issue key**. Especially when using the JQL mode or Issue list mode, that return a list of multiple issues.
Parser expression (standard issue type)	If you want to dynamically set the issue type using an expression, make sure it returns an **issue type name**.
Parser expression (sub-task issue type)	If you want to dynamically set the sub-task issue type using an expression, make sure it returns a sub-task **issue type name**.

Fields

Summary

Define the summary of the issue(s) to be created.

Learn more about the possibilities of the JWT expression editor.

Description

Define the description of the issue(s) to be created.

Learn more about the possibilities of the JWT expression editor.

Additional fields

You can **optionally** set **additional field values.** After clicking the **Add** button you can define how to set the values.

The available options may vary depending on your previous selection(s).

Option	Description
Automatic	This option is **only available** for the field **Assignee**.
Selected value	Select a specific value from the dropdown list.
Copy field value from current issue	Copy the value from a field in the current issue. Select the field you want to copy the value from.
Copy field value from epic issue	Copy the value from the related Epic issue. Select the field you want to copy the value from.
Copy field value from seed issue	Copy the value from a seed issue. This option is **only available** if multiple issues were selected and JQL mode or Issue list mode was used to set the value).
Copy field value from parent of new issue	This option is **only available** if you create a sub-task under a new parent. **Only available** if the issue to be created is a sub-task.
Set field value manually (parser expression)	Select the field to copy from using an expression. Learn more about the possibilities of the JWT expression editor.

Copy remaining fields

Set this parameter if you want to **optionally** copy or clone all (remaining) field values from an issue.

The available options may vary depending on your previous selection(s).

Option	Description
Don't copy fields	By default **no** other field values will be copied.
From current issue	All other field values will be copied from the current issue.
From parent of current issue	All other field values will be copied from the parent of the current issue.
From parent of new issue	All other field values will be copied from the parent of the sub-task to be created. **Only available** if the issue to be created is a sub-task.
From seed issue	All other field values will be copied from a seed issue. This option is **only available** if multiple issues were selected and JQL mode or Issue list mode was used to set the value).
From epic issue	All other field values will be copied from the related Epic issue.

Once some of the options are selected, a checkbox with options to copy all the remaining fields, the attachments, the comments or the issue links of the selected issue will appear.

Issue links

Issues to be linked

Optionally define issue links to be created for the new issue(s). You have the option to link the issue that you are currently creating **to** various other issues.

The available options may vary depending on your previous selection(s).

Option	Description
Current issue	Link the issue(s) to be created to the current issue.
Parent of current issue	Link the issue(s) to be created to the parent of the current issue.
Parent of new issue	Link the issue(s) to be created to the parent of the sub-task to be created. **Only available** if the issue to be created is a sub-task.
Epic of current issue	Link the issue(s) to be created to the related Epic of the current issue.
Seed issue	Link the issue(s) to be created to a seed issue. This option is **only available** if multiple issues were selected and JQL mode or Issue list mode was used to set the value)
Select issues manually (parser expression)	The issues will be set using an expression. Make sure to read all about the JWT expression editor and the available List functions which are extremely powerful when it comes to defining a set of specific issues (e.g. fieldValue(), issuesFromJQL()).

Condition

You can **optionally** define a condition which **must** be met for the link to be created. The expression can be set using the Logical mode.

Additional options

The only available additional option is **Store issue keys**.

If this option is checked, all **issue keys from the issue(s) to be created** will be stored in the field **temporary text 3** as a comma separated list.

Conditional execution

You can **optionally** specify a logical expression to define the circumstances (or conditions) under which the post function should be executed.

The result of the logical expression must return a boolean value of either:

- `true` the post function will be executed
- `false` the post function will **not** be executed

Using the **conditional operator,** even complex or multi-layered conditions can be constructed.

Make sure to learn more about defining logical expressions and browse through the various **examples** here: Logical mode

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.

Use cases and examples

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

If you still have questions, feel free to refer to our support team.