Create issue

The **Create issue** post function automatically creates a **single standard issue** or **sub-task** in any given **project**. The issue type can be selected from all available issue types and sub-task issue types in the system (excluding next-gen projects).

Additional **field values** can be **inherited** or manually set using parser expressions.

Configuration

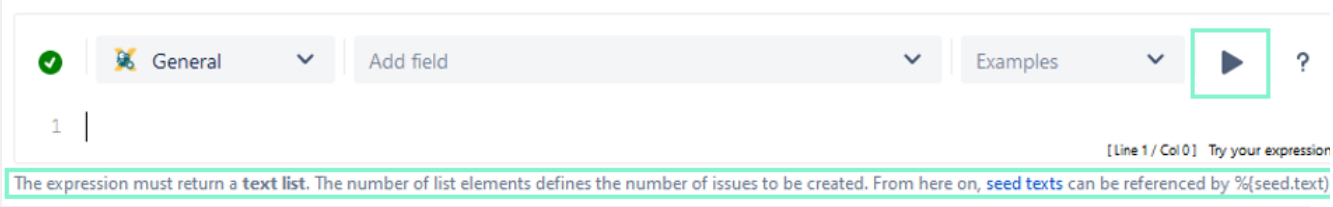
Mode

The **mode** parameter defines **how many** issues you want to create. You can either choose to create just one single issue or multiple ones.

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. In this mode, **only one single issue will be created**.
Multiple issues based on a text list	Determining the number of issues is done using a parser expression. This parser expression has to return a **text list**. The number of list elements defines the number of issues to be created. You can access the respective value using %{seed.text} within all subsequent parameters (except "Conditional execution"). Learn more about seeds.
Multiple issues based on a number	Determining the number of issues is done using a parser expression. This parser expression has to return a **number**. This number defines the number of issues to be created. You can access the current value using %{seed.number} within all subsequent parameters (except "Conditional execution"). Learn more about seeds.
Multiple issues based on an issue list	Determining the number of issues is done using a parser expression. This parser expression has to return an **issue list**. T his number of list elements defines the number of issues to be created. You can ccess the current value using %{seed.issue.somefield} or %{seed.parent.somefield} within all subsequent parameters (except "Conditional execution"). Learn more about seeds.

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 to be created.

The following options are available:

- Selected issue type

- Parser expression (standard issue type):

  > Expects an **issue type ID** or a field code like **`%{issue.issueType.id}`**

- Parser expression (sub-task issue type):

  > Expects a **sub-task issue type ID** or a field code like **`%{issue.issueType.id}`**

> In case a sub-task is chosen, the respective parent has to be selected.

Parent issue (only for sub-tasks)

Choose the parent issue of the sub-task to be created. The following options are available:

- Current issue
- Parent issue
- Selected issue
- Seed issue's parent (**only available** if mode "Multiple issues based on an issue list" was selected)
- Parser expression:

> The input has to be an **issue key**

> The parent issue is only shown when a sub-task issue type is selected.

Project (only for standard issue types)

The project the issue will be created in. It comes with the following options:

- Current project
- Selected project
- Seed issue's project (**only available** if mode "Multiple issues based on an issue list" was selected)
- Parser expression:

  > The input has to be a **project ID** or a field code like **%{issue.project.id}**

> Please note that the project selection does not show up when a sub-task issue type is selected.

Summary

Enter plain text and optionally use field code, e.g. `%{issue.summary}` or functions provided by the selected parsing mode.

Description

Enter plain text and optionally use field code, e.g. `%{issue.description}` or functions provided by the selected parsing mode. You can enrich the styling of your description by using the available markdowns.

Additional fields

In addition to the summary and description of an issue, any other field supported by JWT for Jira Cloud can be set when creating the issue.

After selecting a field and clicking the **Add** button, you can select in the **popup** how to set the value. There are three or - when dealing with selectable fields - four options:

- Copy field from current issue
- Copy field from seed issue (**only available** if mode "Multiple issues based on an issue list" was selected)
- Set field value manually - Read more about the JWT expression editor
- The option **Selected value** is available for the following fields types:
    - Jira Software related fields like **Sprint** or **Epic**
    - User related fields like **Assignee** or **Reporter**
    - Version related fields like **Affects-** or **Fix version/s**
    - Component/s
    - Labels
    - Priority
    - Resolution
    - Security level
    - and all option based custom fields that are supported

Issue links

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 to be created to the current issue.
Parent of current issue	Link the issue to be created to the parent of the current issue.
Parent of new issue	Link the issue 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 to be created to the related Epic of the current issue.
Seed issue	Link the issue(s) to be created to a seed issue. **Only available** if mode "Multiple issues based on an issue list" was selected.
Select issues manually (parser expression)	Link the issue to be created to the issues returned by the parser expression (General mode or Jira expression mode)

Condition

You can **optionally** specify a logical expression or a Jira expression depending on the chosen Parsing mode to define the circumstances (or conditions) under which the link should be created.

Run as

Select the user that will be used to execute the post function. By default, it is set to the current user that executes the transition.

The following options are available:

Option	Description
Selected user	Select a specific Jira user.
User in field	Select the field containing the user that will be used to execute the post function.

The configured user must have all necessary permissions to transition the target issue.

Conditional execution

You can **optionally** specify a logical expression or a Jira expression depending on the chosen Parsing mode 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

Use cases and examples

Use case	JWT feature	Workflow function	Parser functions	Use case description	Complexity
Create an issue in the current project		Create issue		Create an **issue** in the **current project** and additionally set a **summary.** This use case comes in handy if you quickly need to create i.e. a new bug which **relates to the current issue**	BEGINNER
Create an issue with a summary to check for attachment type		Create issue	matches()	Creating many issues and adding a summary and a description can be a bit frustrating and time-consuming. To avoid such things, the following use case shows you how to create a sub-task with a summary to check for attachment type in the parent issue.	INTERMEDIATE
Create a simple sub-task		Create issue		Create a **sub-task,** set the **summary** based on the **parent's component,** and set the **assignee** to the **c urrent user**.	BEGINNER
Create a story in an Epic		Create issue		Link your Epic each time you create a story.	BEGINNER

Create a sub-task for each component		Create issue	toStringList() getMatchingValue()	Create a sub-task for each selected component in the current issue.	BEGINNER
Create a sub-task for each user selected in a User Picker field		Create issue	jiraExpression()	Create automatically a sub-task for each selected user in a User Picker (multiple users) field of the transitioned issue.	BEGINNER
Create a sub-task for high priority issues		Create issue		Create a **sub-task only if** the **priority** of the current issue is **"High".**	INTERMEDIATE
Create a sub-task linked to issues with a specific priority		Create issue		Create sub-tasks and link them to the parent or current issue that has a specific priority of your choice.	INTERMEDIATE
Create a sub-task mentioning the assignee when a high priority task is ready for review		Create issue		Keep your team on track and up to date by creating a sub-task mentioning the assignee's full name and with issue links linked to the appropriate issue whenever a **high priority** issue has been moved to the status "**Review**"	BEGINNER
Create multiple sub-tasks with different summaries and descriptions		Create issue	nthElement()	Create multiple **sub-tasks** with different **summaries and descriptions**.	INTERMEDIATE
Create two sub-tasks when a user story is being approved		Create issue		When a **story** is **approved**, **two sub-tasks** for Development and QA will be created.	INTERMEDIATE

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