

Add comment

This post function automatically creates a comment in the **current issue**. The comment text can be composed of free text in combination with various fields using their field codes.

The comment author can be selected among various user fields and the visibility can be restricted by project role, user group, or Service Management visibility.



Configuration

Comment's text

Define the content of the comment by choosing one of the following options:

Option	Description
Add comment manually (parser expression)	Compose your comment text as a combination of free text and field codes.
Copy field value from current issue	Select any field. The value of the field will be the content of your comment.

Comment visibility

Define who will be able to view the comment. Choose among the following options:

Option	Description
Unrestricted (all users)	The comment will be visible to all users who have access to the issue.
Restricted to project role	The comment will only be visible to users in a certain project role. e.g., Administrators, Developers...etc.
Restricted to group	The comment will only be visible to users in a certain group.
Customer facing comment for Jira Service Management projects	The comment will be visible to Service Management customers.
Internal comment for Jira Service Management projects	The comment will only be visible to all Service Management agents.

Additional options

By default, **no additional email** will be sent for this comment.

If you check this option, Jira will **send email notifications** according to the associated notification scheme.

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
Add comment with the request participants' display name		Add comment	toString() textOnStringList() toStringList() userDisplayName()	
Alert the assignee of important issues		Add comment		
Alert the reporter		Add comment		
Add internal Jira Service Management comment		Add comment		

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