

# Execute remote action

JWT offers the Execute remote action post function to execute predefined or custom actions not only locally on the current Jira environment, but also on any other web service using JSON.



## Configuration

The post function offers two types of configuration. In the [default view](#) you can choose between predefined and directly usable [action templates](#). In the [expert mode](#), actions can be freely defined and configured.

## Wizard



The wizard is the default configuration view of the Execute remote post function. It was built to enable an easy and efficient configuration of common use cases. For those use cases, action templates are shipped with the app, which can be configured in a three-step approach.

You can switch at any time to the Expert mode, where the configuration so far is taken over, using the Expert mode button on the top. Be aware that once you change something in expert mode, you cannot switch back.

## Create remote action

1

### Select action

At the very beginning, you need to choose an [action template](#) that suits your purpose. You can filter those templates by the target service and search across all provided ones. After selecting one, click next to get to the next step.

2

### Select connection

After selecting an action template, you need to choose the remote service, the action shall get executed against. If you have selected an action template that fits to the target platform, you can select next to your custom connection, the current instance additionally.

If you cannot find a connection that fits, you can easily [configure a new one](#).

After selecting one, click next to get to the next step.

3

### Customize your action

In the last step, you can finish your configuration by setting all the required action template parameters using the set value link next to the parameter name. Some templates have optional parameters that can be chosen using the "Add parameter" action.

Each parameter can be set either by selecting a Jira field that contains an according value or by setting a custom value using the "set value manually option" and passing a parser expression

Besides setting the action parameters, you can also configure fields to be updated after the action has been executed. If provided in the action template, you can set a field using a dedicated part of the remote service's response. In every case, you can store the complete response and the status code of the remote service after executing the action.

After setting the action parameters and optionally storing the service response, you can click "Add" to store the post function in your transition.

## Expert mode



The expert mode allows you to act as an expert in many other complex directions and possibilities! You can create your own actions, connections and extend the already created actions!

## Basic configuration

Here you can define the core parts for setting up the communication with the external service. For this, you need the following:

### Action

Within the **action** parameter you can either choose an existing custom action or create a new one. Additionally, you can select and use any action template known from the Wizard configuration.

An action is composed of:

- Method
- REST path
- Parameters
- Action body

Learn more about how to create a [custom action](#)!

### Connection

Within the **connection** parameter, you can choose or define, where your action get executed against. You can either use the default connection configured in the selected action or override it with a different one. Learn more about [creating new connections](#).

You can use a connection for many actions.

## Customize action

### Parameters

You can only query parameters

Option	Description
Name	A proper name to describe the parameter
Value	You can choose between: <ul style="list-style-type: none"><li>• Get field value from current issue</li><li>• Set field value manually (by passing a parser expression) ✓</li></ul>

## Custom headers

You can **edit** the custom headers from the selected connection, as well as **remove** or **add** new headers.

## Fields

Please note that only web services providing JSON are currently supported. To easily navigate in JSON structures, you can use the JMESPath.

Option	Description
Action template specific	Most action templates provide action-specific options to access attributes out of the action body quickly.
Response body	Choose if you want to get the whole body of the response.
Response status	Choose if you want to get only the status of the response.
Set field value manually (parser expression)	Select the field to copy from using an expression.  <div>Learn more about the possibilities of the <a href="#">JWT expression editor</a>.</div>

## Additional options

- **Send email notification:** Notification will be sent according to the associated notification scheme.
- **Update issues immediately:** Choose if you want Jira to fire a separate "Issue updated" event for any field change. This will also result in a dedicated **issue history** entry.

By default, all the additional options are disabled.

## 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 an external link remote action				STAFF PICK
Create an issue comment remote action				STAFF PICK
Create a page remote action				STAFF PICK
Create a project component remote action				
Create a project remote action				STAFF PICK
Create a space remote action				
Create a user remote action				
Create a version remote action				
Get related assets (Insight) remote action				STAFF PICK
Invite a user remote action				
Link an issue remote action				
Log work remote action				
Transition an issue remote action				STAFF PICK

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