

Triage issues created by email

On this page

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

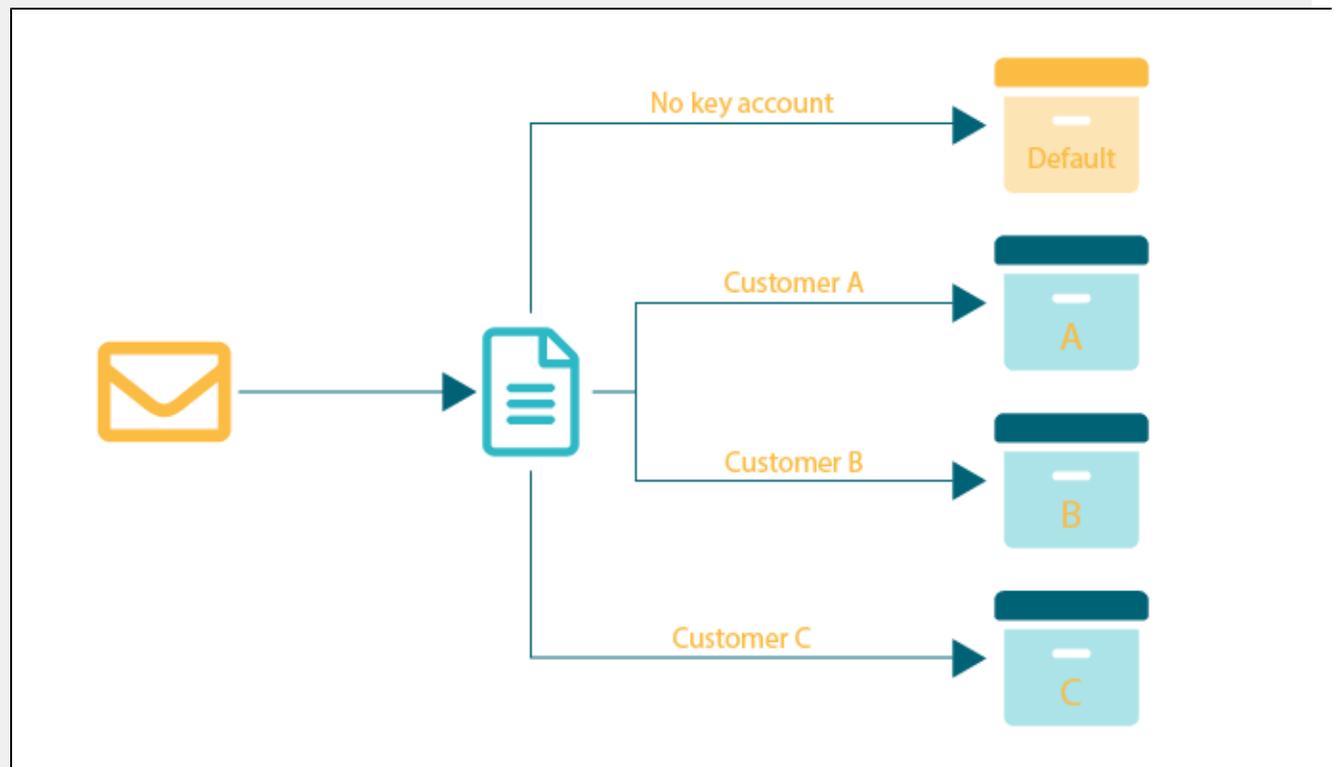
Use case

Jira can be perfectly used to **communicate with your customers**, e.g. by using Service Management or dedicated projects.

It is also possible to **create issues via email**. But in this case the created issues **always belong one specific project**.

In the following use case, **emails sent by one of our key account customers will be automatically moved to the corresponding project**.

Every key account customer is part of the group **key-account** and every key account company has one dedicated project within Jira with a project key corresponding to the email's domain.



Make sure this post function is the last one on your post functions list.



Configuration

1

Add the [Move issue](#) post function to desired workflow transition.

2

Project

Choose **Set project manually (parser expression)**

3

Expression

Set it to **advanced text** and use the following expression:

```
toUpperCase(toString(findPattern(%{issue.reporter.email}, "(?<=. *@)([a-zA-Z0-9_-]+(?:\\.\\.)+)")))
```

This expression grabs the domain from the reporter's email address and transforms it to upper case:

Email address	Return value
admin@atlassian.com	ATLASSIAN
doe@realcompany.com	REALCOMPANY

So for every key account there must be one dedicated project with the corresponding project key.

The used expressions are

- [toUpperCase\(\)](#)
- [toString\(\)](#)
- [findPattern\(\)](#)

4

Issue type

Choose **Retain the issue type**

5

Status

Choose **Retain the status**.

6

Conditional execution

Enter the following expression

```
isInGroup(%{issue.reporter}, "key-account")
```

By using the conditional execution, the created issue will **only be moved to the correct project, if the reporter is a key-account customer.**

The used expression is [isInGroup\(\)](#).



Screenshots

Update parameters of the Move issue Function for this transition.

Update parameters of the Move issue Function for this transition.

Basic configuration*

Define the required parameters needed to execute this post function.

Project*

Set project manually (parser expression) ▼

Select the project the issue will be moved to.

Expression*

```
1 toUpperCase(toString(findPattern(%{issue.reporter.email}, "(?<=.+@)([a-zA-Z0-9_-]+(?:\\.\\.)+)")))
```

The expression must return a **project key** or a **project name**.

Issue type*

Retain the issue type ▼

Select the issue type the issue will be moved to.

Status*

Retain the status ▼

Select the status the issue will be moved to.

Fields

Optionally define further fields to be updated. By default, all other field values will be retained.

Field Value Write protection

No fields configured!

Add field ▼

> Additional options

Conditional execution

The post function will only be executed if the following condition (a logical expression) is met.

Expression

```
1 isInGroup(%{issue.reporter}, "key-account")
```

The expression must return **true** or **false**.

> Run as

Move issue

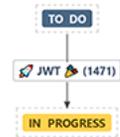


Documentation [↗](#)

Move an issue to other projects, statuses or issue types.

Workflow

JWT 3.0.0 Use cases



Examples

Pick one of the following example configurations and get started in no time. More details will be provided upon your selection.

- Move an issue to another issue type
- Move an issue to another project
- Triage issues created by email



Related use cases

Use case	JWT feature	Workflow function	Parser functions	Label
Move an issue to another issue type		Move issue		
Move an issue to another project		Move issue		
Triage issues created by email		Move issue	toUpperCase() toString() findPattern() isInGroup()	STAFF PICK

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

