Using Automation and Reports with Issue Translation for Jira Service Management

Custom fields created by the app

As a base for using Jira's built-in automation and reporting functionality, the app provides a custom field that stores the **"Customer Language"** for every support case in your service desk. The Customer Language is determined in the first customer request.

You can get three more custom fields when you enable the experimental features language sentiment and key phrases:

- **Initial Sentiment**
- **Last Sentiment**
- **Key Phrases**.

These custom fields are not available by default.

Use Cases

The three following use cases leverage the Project Automation feature which has been included as in the core of Jira Service Desk cloud. To access it:

- Navigate to the Service Desk project where you want to set the rule
- Enter Project Settings
- Scroll towards the bottom of the left panel, until you find the **Project Automation** option
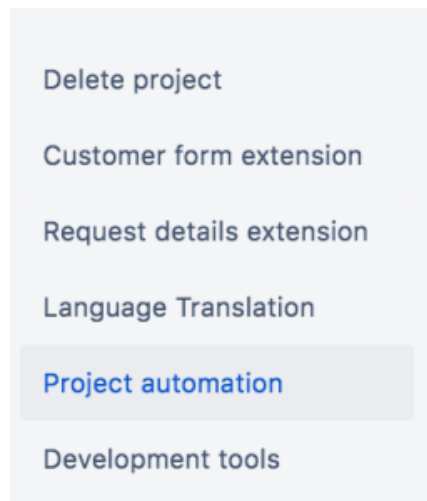
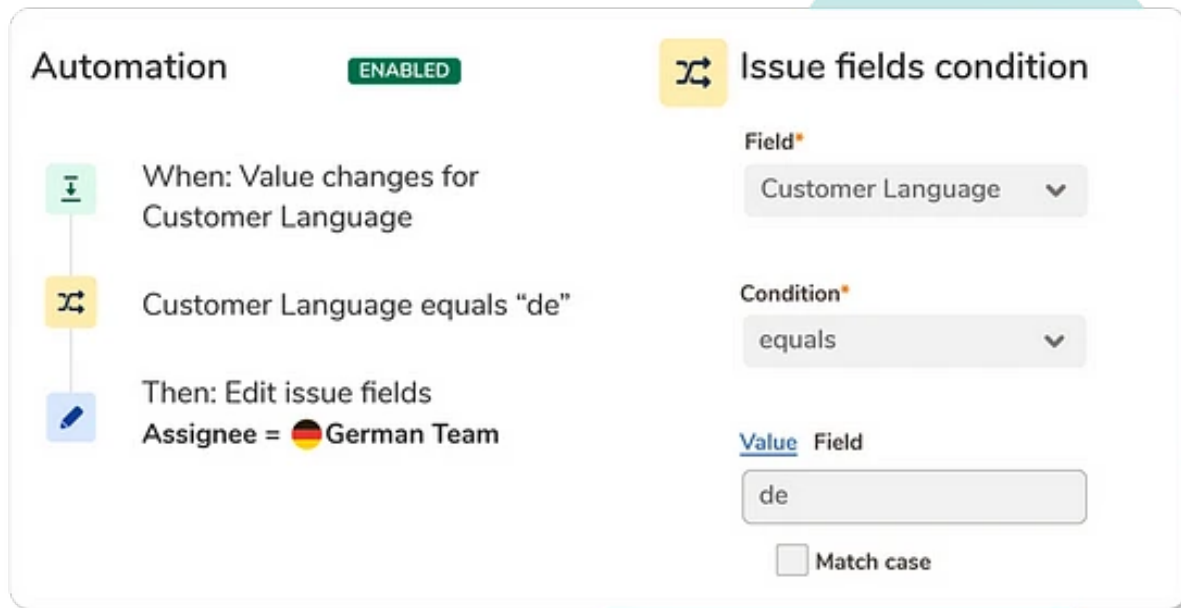- On the automation screen, click on Create rule, at the right top
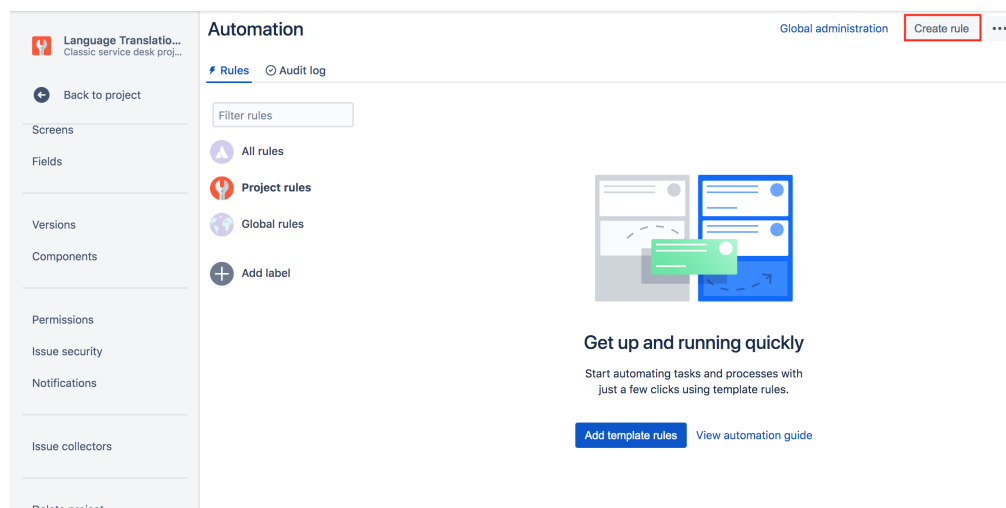
Automation should not be mistaken with **Legacy Automation,** which contains legacy automation rules in Jira Service Desk cloud.

The same automation functionality can be added to Jira Service Desk Server and Data Center with the Automation for Jira app from the Atlassian Marketplace.

1. Reassigning issues based on the customer language

Business case

If you have native speakers of a language in your service desk team, you probably want them to own tickets in that language and override automated translations.

You can do this by:

* Adding this language as "agent language" in settings and letting them handle the conversations in that language.
* With the built-in automation feature you can reassign an incoming request after the Jira Service Desk Language Tools determined the language of the request.

Step-by-step Guide

Step 1: Select customer language as trigger

* Select the trigger **Field value changed**

ⓘ Rule details

▶ New trigger
   Select an event or schedule.

○ Add component

▶ **New trigger**

Triggers start the execution of a rule. Triggers can listen for events or be scheduled to run.

All triggers ⌄ | Start typing to filter components |

Recommended

⬇ **Field value changed**
   Rule is run when an issue's field value changes. POPULAR

💬 **Issue commented**
   Rule is run when a new comment is added to an issue.
   POPULAR

＋ **Issue created**
   Rule is run when an issue is created. POPULAR

⤴ **Issue transitioned**
   Rule is run when an issue is transitioned through its workflow. POPULAR

Issue triggers

⬇ **Field value changed**
   Rule is run when an issue's field value changes.

👤 **Issue assigned**
   Rule is run when an issue is assigned to a user.

- search for the field **Customer Language** and select it

⬇ **Field value changed** ✏

This rule will trigger when the value of the fields selected below changes. Learn more.

Fields to monitor for changes *

| customer   ⌄ |

| Customer Language |

| All issue operations   ⌄ |

Optionally select which operations this trigger will execute for. Leave blank for all operations.

- In the second dropdown, choose the option to **Edit issue**

⬇ **Field value changed** ✏

This rule will trigger when the value of the fields selected below changes. Learn more.

Fields to monitor for changes *

| Customer Language ✕   ⌄ |

You can also match field names using regular expressions.

For

| Edit issue ✕   ⌄ |

Optionally select which operations this trigger will execute for. Leave blank for all operations.

                                              Cancel   **Save**

- Save the trigger to return to the rule menu

Step 2: Specify the customer language in the condition

- Choose to create a **New condition**
- Choose the recommended option **Issue fields condition**

New condition 🗑

Actions will only execute if all conditions preceding them pass.

Recommended

⇄ **Issue fields condition**
Check whether an issue's field meets a certain criteria
POPULAR

All components

{} **Advanced compare condition**
Compares two values:
{{issue.status.name}} equals
Done

⊷ **Affected services condition**
Check if the affected services field matches certain criteria

⇄ **If / else block**
Perform different actions using if, else-if and else to control the flow.

📎 **Issue attachments**
Checks if issue attachments exist or don't exist

- Select again the **Customer language** field
- Activate the condition **equals**
- Include the **value** of the language that you want to reassign. In this example, it's **de** for German.

Automation   NEW

ⓘ Rule details

⤓ **When: Value changes for**
Customer Language

⇄ **If: Customer Language equals**
de

○ Add component

⇄ **Issue fields condition** 🗑

Checks whether an issue's field meets a certain criteria. Learn more.

Field *
Customer Language ▾

Condition *
equals ▾

Value   Field

de

☐ Match case

- Save the condition

Step 3: Define to whom the issue will be reassigned as the action

- In the component list, select a **New Action**

⚡ New action 🗑

Actions perform changes to a system.

All components ⌄	assign

Recommended

✏️ **Edit issue**
Update certain issue fields.
POPULAR

All components

👤 **Assign issue** Select a user to assign the issue to.	✏️ **Edit issue** Update certain issue fields.

- Select **Assign Issue. You will see multiple options in the dropdown. The two more interesting are:**
  - **Specify user**. Select this action if there is a single user who can speak the selected language
  - **Users in a group**. Select this action if there is more than one user who can speak the selected language, i.e. german-agents. Make sure to maintain this group.

👤 Assign issue 🗑

There are numerous ways to assign issues to users – from specifying the user, to copying from other issues and fields. You can also assign to users in a list based off smart criteria.

Learn more about the different options for assigning issues.

Assign the issue to

User in a group	⌄

User in a group
User in a role
A user in a defined list
Another field value or comment
Automatic
Copy from another issue
Smart value
Specify user
Unassigned

The first 50 group members will be used.

Cancel **Save**

Now the rule has been completed. To start using it, simply:

- Give it a name
- Publish it

Automation NEW   Return to list

German tickets to Jaime   **Add component**

ⓘ Rule details

⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯⎯   New condition
                                            Actions will only execute if all
When: Value changes for   conditions preceding them
Customer Language   pass.   OR   German tickets to Jaime

                                            New action
                                            Actions perform changes to a
If: Customer Language equals   system.   Turn it on
de
                                            Branch rule / related issues
                                            Branch rule and run conditions
Then: Edit issue fields   & actions for these issues.
Assignee, **Advanced**

Add component

Instructions for Server

2. Sending an E-Mail-Notification on the use of rarely used languages

For some rarely used languages that you want to monitor, you can use an automation rule to send you an alert email is someone uses it.

In the following example, we use the Automation Lite for Jira app for cloud. If you don't have it yet, you can get it on the marketplace. It's free. Unfortunately for server, the trigger that we use in this example is a PRO feature. You can try it for 4 weeks.

Cloud:

Server:

3. Escalating a service request with low language sentiment (experimental)

Using the Microsoft Azure language analytics the app provides you with a language sentiment on each customer request and the following responses. Language sentiment comes with a value between 0 and 100 where 0 is worst and 100 is best.

We recommend to experiment with this feature, if it the results are meaningful in your environment, as we made the experience that this is not always the case.
Anyway, when you want to use this feature, you have to switch in on in the app settings of the project.

Then you can do things like this:

Cloud:

Server:

4. Creating a queue for a specific language

Alternatively, you can also easily create a service desk queue, based on the language detection feature of Jira Service Desk Language Tools: one queue for incoming German requests, one for Spanish, etc.

The following screencasts show how to do this.

Cloud:

Server: