

Service Management to Multiple Projects

Last Modified on 01/15/2026 3:37 pm EST

Introduction

In this article, we will look to connect an external-facing service desk with an internal development Jira. The external Jira is configured with a single service desk configuration - providing generic request types such as 'Raise a problem', and 'Submit an idea' ... Some of these requests need to be escalated to the internal development Jira and kept in sync such that the support engineer can keep in touch with the customer whenever development has a couple of questions or whenever a fix is made available.

The Result

Keep it Simple

The case has 30 different products under development. We wanted to avoid customers having to select the right service desk to raise tickets because it is not very nice for the customers to have them browse through different categories, and if they do, they are not always right so we have to recategorize regularly and lastly the service desk is integrated with email - requiring to categorize anyway. Instead of complicating raising issues for the customer, we decided that we wanted a very simple UI for the customer, providing just 2 options.



Raise a problem



Suggest an improvement



Raising an issue is not harder than entering an email.



Help Center / ACME

Raise a problem

Summary

Email notifications are not sent with French profile

Description (optional)

When my French colleague enters a record in our symphony application, no email notification is sent.

It works fine when the profile language is set to English

Thanks for looking into it

Peter

Attachment (optional)

Drop files to attach, or [browse](#)

[Create](#) [Cancel](#)

Simple isn't it?

Involve Development

Support engineers are there to support. They are great at keeping the customer informed and handling all customer concerns, but some of these issues are nasty and need to be addressed by development. There should be a way to dispatch an issue to another team - who can do their magic and provide a fix (or workaround) for the problem.

Adapt the Workflow

First of all, we needed a way to track the issue when it is assigned to development, so we changed the standard development workflow by adding a status 'Waiting for development' and adding a couple of transitions.

- **Escalate** transitions the issue from '**Waiting for support**' or '**Waiting for the customer**' to '**Waiting for development**' indicating that development is working on it.
- **Dev done** transitions the issue from '**Waiting for development**' to '**Waiting for support**' so that the support team can communicate to the customer so they can give it a try.



What Team Should Handle the Problem?

A support issue can be raised against 30 different products, so there should be a way to categorize the issue such that the appropriate team is notified about the request and start handling it. We, therefore, added a simple drop-down list, listing all the potential products. When an issue is escalated to development, the support engineer can provide his assessment of what product and at what priority the problem should be handled.

Escalate

Priority: Critical

Product: CRM Back End

Comment: Respond to customer

🔒 Your comments will not be visible to customers on the portal.

Style: B I U A ³A       

Hi Serge,

Can you have a look at the configuration of the email server - it appears that the notifications are queued but not sent.

Please keep me updated, the customer is a bit anxious

Thanks,
Rob

[preview](#) · [syntax help](#)

[Escalate](#) [Cancel](#)

Delegating to Another Project on Another Issue Tracker

For security reasons, the service desk environment is hosted on another Jira than the development Jira. We didn't want to expose the internal development information to the internet. A configuration is easily made, risking making IP-sensitive information available.

When a support engineer escalates a customer request to development - the request should be copied over to the development Jira (in the right project !) and all information must be kept in sync. Development needs to know which customer has raised the issue, it must be possible to request additional information (such as log files or screenshots), and support must work with the customer once the issue is resolved.

Configuring Connection

Enter Exalate - the flexible Jira Synchronization solution.

For more information on how to set up the synchronization between two Jira instances, please read the [Getting Started guide](#). At a certain stage of the configuration, it is necessary to configure the synchronization such that new issues get created in the appropriate development project.

Exalate provides flexibility through [Groovy scripting](#) and a very open object model. For more information, please read the [issue fields available for synchronization](#) which details what objects and fields are available when a message is being processed.

To ensure that the customer request is raising an issue in the appropriate project, we configured the Create Processor on the development server to look up the project.

```
...
def targetProject = replica.customFields.product?.value?.value
def project = nodeHelper.getProjectByName(targetProject)

issue.projectKey = project ? project.key : "GEN"
...
```

That's it - when a new development project is created, it is sufficient to add the name of the project to the drop-down. No additional configuration is necessary.

ON THIS PAGE

[Introduction](#)

[Product](#)

[The Result](#)

[About Us](#) 

[Keep it Simple](#) 

[Glossary](#) 

[Involve Development](#)

[API Reference](#) 

[Configuring Connection](#)

[Security](#) 

[Pricing and Licensing](#) 

[Resources](#)

[Subscribe for a weekly Exalate hack](#) 

[Academy](#) 

[Blog](#) 

[YouTube Channel](#) 

[Ebooks](#) 

[Still need help?](#)

[Join our Community](#) 

View our Community

[Visit our Service Desk](#)

[Find a Partner](#)