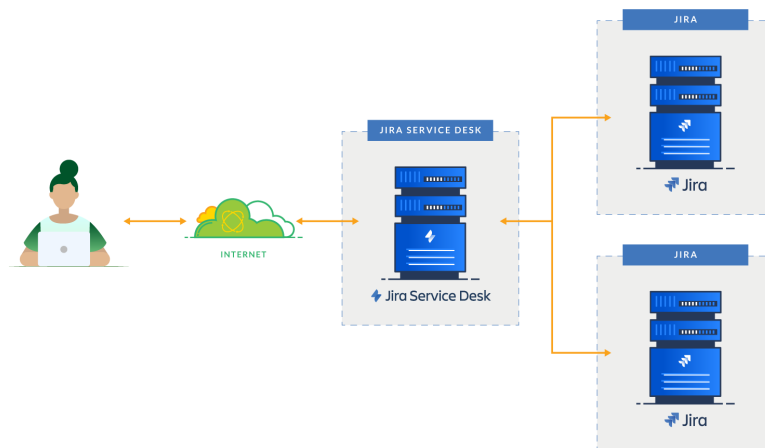


# Unidirectional Sync from Service Desk

Last Modified on 03/25/2024 10:22 am EDT

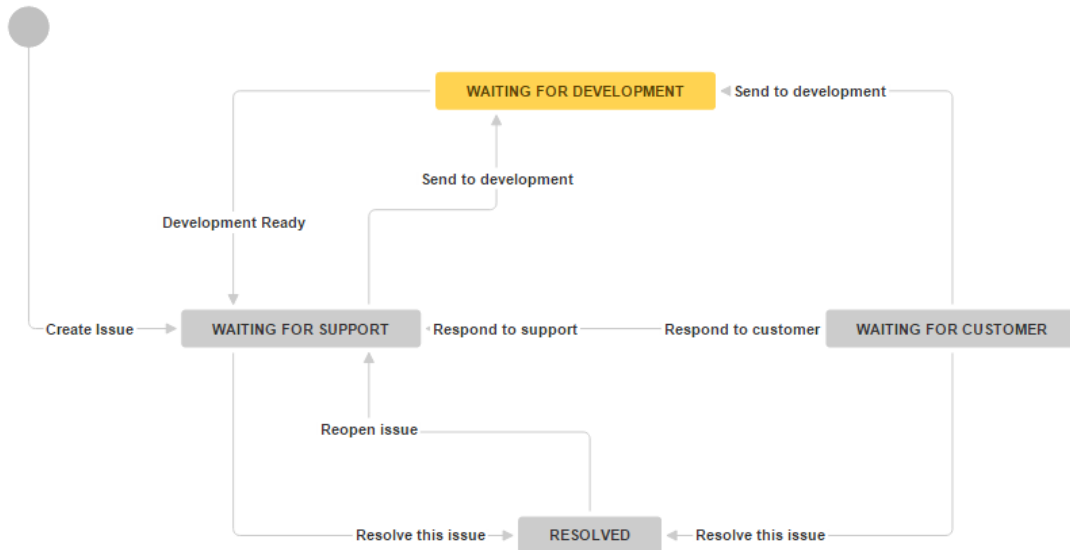
**Warning:** Despite our best efforts, code can change without notice due to a variety of factors. If you encounter an issue in any of the code shown here and find that a specific block of code is not correct, or is causing errors, please check with the [Community](#) to find an updated version.

This use case shows how you can connect 2 Jira instances such that one instance is used for servicing customers, and the other for tracking development tasks.



## Jira Configuration

To support the **Send to development** function, the workflow of the Service desk configuration should be slightly changed.



You need to add an *Exalate post function* to the **Send to development** transition using the previously created connection **DeskDev**.

Workflows

/ JIRA Service Desk IT Support Workflow generated for Project DESK (Draft)

Edit View Properties

Transition: Send to development

Delete ?



Screen: Comment

Triggers 0 Conditions 0 Validators 0 Post Functions 6

The following will be processed after the transition occurs

[Add post function](#)

1. Set issue status to the linked status of the destination workflow step.
2. Add a comment to an issue if one is entered during a transition.
3. Update change history for an issue and store the issue in the database.
4. Re-index an issue to keep indexes in sync with the database.
5. Escalate this issue towards '**DeskDev**'
6. Fire a **Generic Event** event that can be processed by the listeners.

**Note:** The *Exalate post function* is the last in the list as the synchronization needs to take all the changes (such as comments) with it.

# Sync Rules Setup

## Service desk Configuration

### Outgoing sync

```
replica.key      = issue.key
replica.assignee = issue.assignee
replica.reporter = issue.reporter
replica.summary  = issue.summary
replica.description = issue.description
replica.comments = issue.comments
replica.resolution = issue.resolution
replica.status   = issue.status
replica.project  = issue.project
replica.attachments = issue.attachments
```

### Incoming sync

```
if (firstSync) {
  return // ignore incoming requests to create issues
} else {
  issue.summary    = replica.summary
  issue.description = replica.description
  issue.comments += replica.addedComments
  issue.attachments += replica.addedAttachments

  //When the development issue is resolved, trigger the development ready transition

  if (replica.resolution != null) {
    issue.doTransition = "Development Ready"
  }
}
```

## Development JIRA Configuration

### Outgoing sync

```
replica.key      = issue.key
replica.assignee = issue.assignee
replica.reporter = issue.reporter
replica.summary  = issue.summary
replica.description = issue.description
replica.comments = issue.comments
replica.resolution = issue.resolution
replica.status   = issue.status
replica.project  = issue.project
replica.attachments = issue.attachments
```

### Incoming sync

```
issue.projectKey = "ACME"
issue.typeName = "Task"
issue.summary = replica.summary
issue.description = replica.description
issue.comments += replica.addedComments
issue.attachments += replica.addedAttachment

//When the remote issue is in status 'waiting for development' and the local issue has been closed, reopen it

if (replica.status.name == "Waiting for development" && issue.resolution != null) {
  issue.doTransition = "Reopen"
}
```

Have more questions? [Ask the community](#)

#### Product

#### ON THIS PAGE

[About Us](#)

[Release History](#)

[Jira Configuration](#)

[Glossary](#)

[API Reference](#)

[Security](#)

[Pricing and Licensing](#)

#### Resources

[Academy](#)

[Blog](#)

[YouTube Channel](#)

[Ebooks](#)

#### Still need help?

[Join our Community](#)

[Visit our Service Desk](#)

[Find a Partner](#)