How to Sync Multiple Projects with Multiple Issue Types in Jira Cloud

Last Modified on 01/28/2026 7:12 am EST

In some cases, you may need to sync a certain type of issue into the specific project. Depending on the replica issue type or project key, you can specify the condition in the script. With Exalate, you can configure synchronization of the issue type, depending on the project.

By using synchronization processors, you can define:

- whether you want to send the project key or issue type from the Source Instance.
- what issue type and project are incoming issues created in.

Use Case Example

Imagine you are syncing with your partner. Your partner's instance is sending you some issue data, specified in their **Outgoing sync**. E.g. issue type and project key. You have certain requirements on how these incoming issues should be recorded on your side. **Incoming sync** allows configuring how incoming sync data should be created on your side.

Source Side

Outgoing sync

```
replica.key           = issue.key
replica.project       = issue.project
replica.type          = issue.type
```

Destination Side

- If the issue they sent is of type **Bug,** create an issue with the type **Test** in the project **Testers**. In other cases create an issue with the type of **Task** in the project **Developers**.

Incoming sync

```
if(replica.type.name == "Bug") {
  issue.projectKey   = "TESTERS"
  issue.typeName     = "Test"
}
else{
  issue.projectKey   = "DEVELOPERS"
  issue.typeName     = "Task"
}
```

- If the issue they sent is of type **Bug** from the project **SD,** create an issue with the type **Test** in the project **Testers**. In other cases create an issue with the type of **Task** in the project **Developers**.

Incoming sync

```
if(replica.project.key == "SD" && replica.type.name == "Bug") {
  issue.projectKey   = "TESTERS"
  issue.typeName     = "Test"
}
else{
  issue.projectKey   = "DEVELOPERS"
  issue.typeName     = "Task"
}
```

- You can decide which project to sync the issue with, based on the received issue project key from the source side:

  Incoming sync

```
def projectMapping = ["SERVA":"CLOUDA", "SERVB":"CLOUDB"] // ["REMOTE PROJECT KEY": "LOCAL PROJECT KEY"]
issue.projectKey = projectMapping[replica.project.key]
```

You can use **return** in the Incoming sync rules before any property on the issue variable is changed. For example, you can use **if/else with a return** if you don't want to sync issues that don't match the conditions specified above.

```
// no issue.XXX = YYY before the return below
// ...
if (
 [
 "question",
 "incident",
 "changes request",
 "application upgrades",
 "os patching"
  ].any { it.equalsIgnoreCase(replica.type?.name) }) {
  issue.projectKey = "WHSUP"
} else if ("task".equalsIgnoreCase(replica.type?.name)) {
issue.projectKey = "WHADM"
} else {
  return
}
```

ON THIS PAGE

Source Side

Destination Side