

# How to Sync Multiple Projects with Multiple Issue Types in Jira On-premise

Last Modified on 04/09/2024 6:15 am EDT

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

```
1 replica.key      = issue.key
2 replica.project  = issue.project
3 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

```
1 if(replica.type.name == "Bug") {
2   issue.projectKey = "TESTERS"
3   issue.typeName   = "Test"
4 }
5 else{
6   issue.projectKey = "DEVELOPERS"
7   issue.typeName   = "Task"
8 }
```

- 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

```

1 if(replica.project.key == "SD" && replica.type.name == "Bug") {
2   issue.projectKey = "TESTERS"
3   issue.typeName   = "Test"
4 }
5 else{
6   issue.projectKey = "DEVELOPERS"
7   issue.typeName   = "Task"
8 }

```

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

### Incoming sync

```

1 def projectMapping = ["SERVA":"CLOUDA", "SERVB":"CLOUDB"] // ["REMOTE PROJECT KEY": "LOCAL PROJECT KEY"]
2 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.

```

1 // no issue.XXX = YYY before the return below
2 // ...
3 if (
4 [
5 "question",
6 "incident",
7 "changes request",
8 "application upgrades",
9 "os patching"
10 ].any { it.equalsIgnoreCase(replica.type?.name) } ) {
11   issue.projectKey = "WHSUP"
12 } else if ("task".equalsIgnoreCase(replica.type?.name)) {
13   issue.projectKey = "WHADM"
14 } else {
15   return
16 }

```

#### Product

[About Us](#)

#### ON THIS PAGE

[Release History](#)

[Glossary](#)

[Source Side](#)

[API Reference](#)

[Destination Side](#)

[Security](#)

[Pricing and Licensing](#)

#### Resources

[Subscribe for a weekly Exalate hack](#)

[Academy](#)

[Blog](#)

[YouTube Channel](#)

[Ebooks](#)

#### Still need help?

[Join our Community](#)

[Visit our Service Desk](#)

[Find a Partner](#)