How to Sync Multiple Zendesk Tickets to One Jira Cloud Issue

Last Modified on 01/15/2026 3:42 pm EST

You can sync multiple Zendesk tickets into one Jira Cloud issue by using the Exalate Connect API.

Zendesk

Outgoing sync

In Zendesk, you need to make a custom field and assign a Jira issue key you want to connect to.

In this example, we use `ABC-123` as the default issue key.

```
replica.customFields."Target Issue Key" = issue.customFields."Target Issue Key"
```

Jira Cloud

Incoming sync

After you Exalate the ticket in Zendesk, this script is executed in Jira Cloud. It connects the corresponding tickets.

```
def remoteIssueUrn = replica.customFields."ABC-123"?.value
if(remoteIssueUrn && firstSync){
 def localIssue = httpClient.get("/rest/api/2/issue/"+remoteIssueUrn)
 if(localIssue == null) throw new com.exalate.api.exception.IssueTrackerException("Issue with key "+remoteIssueUrn+" was not found")
 issue.id = localIssue?.id
 issue.key = localIssue?.key
 return;
}
```

Video Tutorial

Also, check out our video on how to sync multiple Zendesk tickets into one Jira issue:

Important: When creating a multi-ticket to a single issue-type of synchronisation, it is possible this may clear data on the target twin. This is a known issue caused by the fact the entity object used to apply onto the underlying issue is not initialized with the right data. We are working to fix this issue, but until then the following can be applied to circumvent the problem, depending on the connector:

Jira Cloud and Zendesk:

```
def fieldValues = [:]
def issueKey = new BasicIssueKey(localWorkItem.id, localWorkItem.id, "issue", fieldValues)
return new scala.Tuple2(issueKey, scala.collection.JavaConverters.asScalaBuffer(traces))
```

GitHub:

```
def fieldValues = ["repo":"org/repo"]
def issueKey = new BasicIssueKey(localWorkItem.id, localWorkItem.id, "issue", fieldValues)
return new scala.Tuple2(issueKey, scala.collection.JavaConverters.asScalaBuffer(traces))
```

Salesforce:

```
def fieldValues = [:]
def issueKey = new BasicIssueKey(localWorkItem.id, localWorkItem.id, "Case", fieldValues)
return new scala.Tuple2(issueKey, scala.collection.JavaConverters.asScalaBuffer(traces))
```

This ensures that Exalate only creates the twin related metadata, so that any future sync transactions are applied.

> Have more questions? Ask the community