How to Sync Tasks and Subtasks in Jira On-premise

> **This article applies to Exalate Classic only.** If you're using the New Exalate experience, please refer to the New Exalate documentation.

With Exalate, you can sync tasks and subtasks while keeping the parent-child relationship on the remote side. Check the example below. The example below works only for connections in Scripting mode. If you already created a connection in Visual mode, you need to create a new connection in Scripting mode.

Source Side

Outgoing sync

Send the ID of the parent issue.

```
replica.parentId = issue.parentId
```

Destination Side

Incoming sync

Find the local parent issue with the help of the **_parent.Id received from_** the remote side.

To sync the parent link with a subtask, you need to first sync a parent task, and then only the related subtasks. So if you sync subtasks _before_ the parent task, you get an error.

```
if(firstSync && replica.parentId){
    issue.typeName    = "Sub-task" //Make sure to use the right subtask type here.
  def localParent = nodeHelper.getLocalIssueFromRemoteId(replica.parentId.toLong())
  if(localParent){
   issue.parentId = localParent.id
  } else {
      throw new com.exalate.api.exception.IssueTrackerException("Subtask cannot be created: parent issue with remo
te id " + replica.parentId + " was not found. Please make sure the parent issue is synchronized before resolving this e
rror" )
    }
}
```

Syncing Subtasks Automatically after Syncing Parent Task (Jira Cloud)

Outgoing sync

```
// Automatically sync all subtasks when syncing parent task
httpClient.get("/rest/api/3/issue/"+issueKey.id).fields.subtasks?.collect{
  def subTaskKey = new com.exalate.basic.domain.BasicIssueKey(it.id, it.key)
  syncHelper.exalate(subTaskKey)
}
```

Syncing Subtasks in Visual Mode for Jira Cloud and Jira on-premise

With this script, you can specify how the subtask is linked to the parent task.

```
if(firstSync && replica.parentId){
    issue.typeName     = "Sub-task" //Make sure to use the right subtask type here.
 def localParent = nodeHelper.getLocalIssueFromRemoteId(replica.parentId.toLong())
 if(localParent){
  issue.parentId = localParent.id
 } else {
     throw new com.exalate.api.exception.IssueTrackerException("Subtask cannot be created: parent issue with remo
te id " + replica.parentId + " was not found. Please make sure the parent issue is synchronized before resolving this e
rror" )
    }
}
```

ON THIS PAGE