Jira Cloud remove issue links when removed on remote site

When issue links are removed on the remote site you want them to be removed on your side as well.

This documentation will show you how to do this.

To sync over issue links you first need to add "replica.issueLinks = issue.issueLinks" in your outging sync.

Outgoing Sync

```
replica.issueLinks = issue.issueLinks
```

To receive the issue link you need to add "issue.issueLinks = replica.issueLinks" in your incoming sync.

First we need to import "jsonSlurper"

Import json Slurper

```
 import groovy.json.JsonSlurper
```

Add this code to your script.

What we do is we create a function that will check if your link exsists within a list of issueLinks.

if they don't exsists within that list it means the issue link is removed on the remote side. so we'll add that issueLink to the toDelete variable.
In the delete() function we'll retrieve the issueLinkd if we have that ID whe can use a httpClient to remove the issueLink from our issue.

Last we loop over our toDelete list and apply the delete() function to remove the issueLinks.

Incoming sync

```
// import jsonSlurper
import groovy.json.JsonSlurper

// Check if a link exists in a list of links
def contains(link, remoteLinks) {
    for(_link in remoteLinks) {
        def localLinkId = nodeHelper.getLocalIssueKeyFromRemoteId(_link.otherIssueId).id
        if(localLinkId == link.otherIssueId) {
            return true
        }
    }
    return false
}

// A list that will hold all links we wish to delete
def toDelete = []
// We loop over the replica issueLinks and resolve all links to a local issue key

def issueLinks = []
if ((replica.issueLinks.isEmpty()) && (!previous.issueLinks.isEmpty())){
    issueLinks = previous.issueLinks
}else{
    issueLinks = replica.issueLinks
}

for(remoteLink in issueLinks) {
    def localIssueId = nodeHelper.getLocalIssueKeyFromRemoteId(remoteLink.otherIssueId).id

    // If the local issue link is not present in the remote, then we should delete it
```

```
    for(link in issue.issueLinks) {
        if(!contains(link, replica.issueLinks)) {
            toDelete += link // so we add it to the list of items we wish to delete
        }
    }

}
// sett issue.issueLinks = replica.issueLinks
issue.issueLinks = replica.issueLinks

def getLocalIssueLinksId(otherId){
    def restApiCall = httpClient.get("/rest/api/3/issue/${issue.key}")
    def jsonText = new groovy.json.JsonOutput().toJson(restApiCall)
    def jsonParsed = new JsonSlurper().parseText(jsonText)

    if (jsonParsed == null){
        return
    }

    def issueLinks = jsonParsed.fields.issuelinks

    for (int i = 0; i < issueLinks.size(); i++) {
        // TODO: check that inwardIssue exist if not use outwardIssue
        if(issueLinks[i]?.inwardIssue != null && issueLinks[i].inwardIssue.id == otherId.toString()) {
            return issueLinks[i].id
        }
        if(issueLinks[i]?.outwardIssue != null && issueLinks[i].outwardIssue.id == otherId.toString()) {
            return issueLinks[i].id
        }
    }

    return
}

def deleteRequest(path) {
    path = path.toString()
    def responseJson = (new JiraClient(httpClient)).http(
        "DELETE",
        path,
        ["sysparm_display_value": ["true"]],
        null,
        ["Content-Type":["application/json"]]
    ) { response ->
        if (response.code >= 400) debug.error ("DELETE ${path} failed: ${response.code} ${response.body}".
toString())
        else { if (response.body != null && !response.body.empty) (new JsonSlurper()).parseText(response.body)
else null }
    }
}

// delete an issue link
def delete(link) {
    def otherIssueId = link.otherIssueId
    def issueLinkId = getLocalIssueLinksId(otherIssueId)

    if(issueLinkId == null) {
        debug.error("Unable to find issue link ${issueLinkId} for issue '${issue.key}'")
    }

    deleteRequest("/rest/api/3/issueLink/${issueLinkId}")
}

// loop over all issuelinks we wish to delete, and delete them
for (link in toDelete) {
    delete(link)
}
```

Questions

Recent Questions

Ask a question

1. 0
votes
Lifecycle blocking error
   - **1** answer
   - Jozsef Lehocz
   - May 13, 2024
   - Space: Exalate
   - exalate
   - blocking
   - error
   - upgrade
2. 0
votes
How to sync issues from two Jira cloud sites to one? Is this possible?
   - **1** answer
   - Vedant Kulkarni
   - Mar 27, 2024
   - Space: Exalate
   - jira-cloud
   - connector-cloud-jira
3. 0
votes
Issues are not syncing to local destination desk
   - **1** answer
   - John Lombardo
   - Mar 21, 2024
   - Space: Exalate
   - exalate
   - connector-cloud-jira
   - jira-cloud
4. 0
votes
Not getting the Custom Field sync value from JIRA to Zendesk
   - **1** answer
   - Dinesh Gupalan
   - Mar 18, 2024
   - Space: Exalate
   - connector-onpremise-jira-zendesk
5. 0
votes
Need to split source comment into multiple destination comments. (The entered text is too long.)
   - **1** answer
   - Kyle K
   - Feb 14, 2024
   - Space: Exalate
   - exalate
   - jira-cloud
   - comments
6. 0
votes
exalate plugin of jira server
   - **1** answer
   - Sreenivasaraju P
   - Feb 08, 2024
   - Space: Exalate
   - exalate
7. 0
votes
How to have keys match between jira cloud instances
   - **2** answers
   - Chris Matthews
   - Feb 07, 2024
   - Space: Exalate
   - connector-cloud-jira
8. 0

votes

how decrease dimension of /var/atlassian/application-data/jira/data/exalate

- **1** answer
- Francesco Doricchi
- Dec 14, 2023
- Space: Exalate
- exalate

9. 1
vote

How to Impersonate Attachments in Jira Cloud?

- **0** answers
- Valeriia Solianikova
- Dec 12, 2023
- Space: Exalate
- exalate
- jira
- attachments

10. 0
votes
status

How to sync field that has been added to an existing connection?

- **2** answers
- Daniel
- Dec 03, 2023
- Space: Exalate
- connector-onpremise-jira
- connector-cloud-salesforce

Ask a question