

Adding basic external script

Last Modified on 03/26/2024 5:55 am EDT

This article shows an example of how you can create an example groovy file, add a script, and call it in the Sync Rules.

1. Make sure you have the **scripts** directory. The directory location depends on the issue tracking platform. Custom scripts can only be deployed on Jira Server/Datacenter and nodes which are deployed through the docker deployment approach.

Platform	location
Jira Server	<jira-home>/scripts
Jira Datacenter	<jira-shared-home>/scripts
Docker based	<p>/opt/<nodename>/data/scripts</p> <p>There could be one of the following values instead of <nodename> :</p> <ul style="list-style-type: none">◦ snownode for Exalate for ServiceNow.◦ adnode for Exalate for Azure DevOps.◦ hpqcnode for Exalate for HP ALM/QC.
Jira Cloud	<p>Jira Cloud, just as any other cloud node, supports a set of specific scripts. Custom scripts cannot be deployed in this environment.</p> <p>Check out List of external scripts for Jira Cloud for more information.</p>

2. Create **BasicFieldSync.groovy** file with the following code, and store it in the right location on your server. There is no need to restart instance/add-on to enable the external script.

```

class BasicFieldSync
{
    static receive(issue,
        replica,
        nodeHelper,
        commentHelper,
        attachmentHelper) {

        issue.summary    = replica.summary
        issue.description = replica.description
        issue.assignee   = nodeHelper.getUserByUsername(replica.assignee?.username)
        issue.reporter   = nodeHelper.getUserByUsername(replica.reporter?.username)
        issue.labels     = replica.labels
        issue.comments   = commentHelper.mergeComments(issue, replica)
        issue.attachments = attachmentHelper.mergeAttachments(issue, replica)
    }
}

```

3. Call the **BasicFieldSync.groovy** script from the Sync Rules.

- Replace the script in the outgoing sync rules (create and change processors) as below:

Existing script

```

issue.summary    = replica.summary
issue.description = replica.description
issue.assignee   = nodeHelper.getUserByUsername(replica.assignee?.username)
issue.reporter   = nodeHelper.getUserByUsername(replica.reporter?.username)
issue.labels     = replica.labels
issue.comments   = commentHelper.mergeComments(issue, replica)
issue.attachments = attachmentHelper.mergeAttachments(issue, replica)

```

New script

```

BasicFieldSync.receive(
    issue,
    replica,
    nodeHelper,
    commentHelper,
    attachmentHelper
)

```

Now you have one file with basic synchronization rules. You can reuse it in outgoing sync processors: [new issues \(create processor\)](#) and for existing issues (change processor). If you add new code into the **BasicFieldSync.groovy**, it is automatically executed in your incoming sync rules (create and change processors).

[API Reference](#)

[Security](#)

[Pricing and Licensing](#)

Resources

[Academy](#)

[Blog](#)

[YouTube Channel](#)

[Ebooks](#)

Still need help?

[Join our Community](#)

[Visit our Service Desk](#)

[Find a Partner](#) 