

# Configure the Connection in Script Mode for Azure DevOps

Last Modified on 02/27/2024 10:55 am EST

## Configure the Synchronization (Optional)

### Set up Sync Rules

Once you test the Connection, you can choose to configure the synchronization behavior with the help of Sync Rules (Groovy-based scripts).

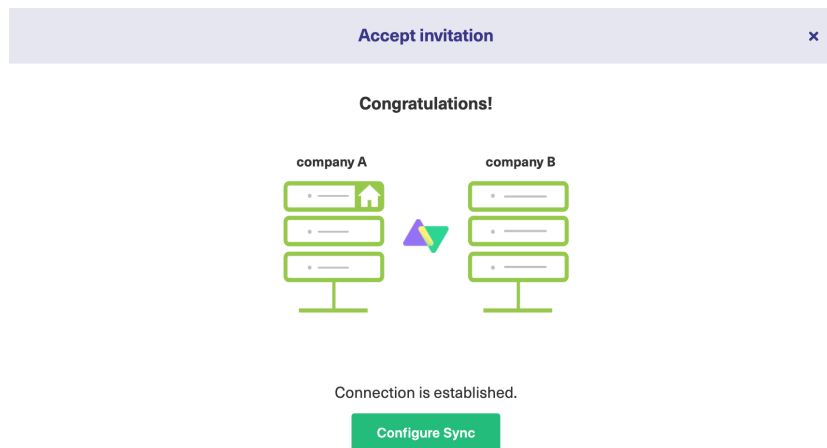
1. Edit your Connection.

There are multiple ways to edit your connection:

- Navigate to **Connections** →  **Edit connection.**

Connection	Issues under sync	Last sync	Status
 Company A to Company B <small>Description of the connection lorem ipsum sit amet</small>	123	Issue TEST-777 46 minutes ago	<span>Active</span>   

- If you just created a connection, select **Configure sync.**



2. Set up **Outgoing sync** and **Incoming sync** in the **Rules** tab.

**Note:** Outgoing sync and Incoming sync scripts:

- are generated by default once the connection has been set up. You can, however, add, edit or remove these scripts to configure your sync according to your requirements.
- are platform specific.
- are present at both the source and the destination instance to give independent control over what data needs to be sent or received.

In the **Outgoing sync**, you can enter scripts to specify what data you will send.

In the **Incoming sync**, you can enter scripts to specify what data you will receive.

```
Outgoing sync
Outgoing sync rules define what information can be sent to the destination instance. Check the documentation for more details.

1 replica.key = workItem.key
2 replica.assignee = workItem.assignee
3 replica.summary = workItem.summary
4 replica.description = nodeHelper.stripHtml(workItem.description)
5 replica.type = workItem.type
6 replica.status = workItem.status
7 replica.labels = workItem.labels
8 replica.priority = workItem.priority
9 replica.comments = nodeHelper.stripHtmlFromComments(workItem.comments)
10 replica.attachments = workItem.attachments
11 replica.project = workItem.project
12 replica.areaPath = workItem.areaPath
13 replica.iterationPath = workItem.iterationPath
14
15 /*
16 Custom Fields (CF)
17 How to send any field value from the source side to the destination side.
18 1/ Add the value to the replica object. Use the Field Name from the field or the API name:
19 https://docs.exalate.com/docs/how-to-sync-work-item-fields-obtained-through-a-rest-api-call
20 2/ Uncomment this next statement out and change accordingly:
21 replica.customFields."CF Name" = issue.customFields."CF Name"
22 */
23
24 // Exalate API Reference Documentation: https://docs.exalate.com/docs/exalate-api-reference-documentation
25
```

```
Incoming sync

1 if(firstSync){
2 // Set type name from source entity, if not found set a default
3 workItem.projectKey = "For test"
4 workItem.typeName = nodeHelper.getIssueType(replica.type?.name)?.name ?: "Task";
5 }
6
7 workItem.summary = replica.summary
8 workItem.description = replica.description
9 workItem.attachments = attachmentHelper.mergeAttachments(workItem, replica)
10 workItem.comments = commentHelper.mergeComments(workItem, replica)
11 workItem.labels = replica.labels
12 workItem.priority = replica.priority
13
14 /*
15 Custom Fields (CF)
16 To add incoming values to an Azure DevOps custom field, follow these steps:
17 1/ Find the Field Name of the CF. You can also use the API name: https://docs.exalate.com/docs/how-to-sync-work-item-fields-obtained-through-a-rest-api-call
18 2/ Check how the value is coming over from the source side, by checking the "Entity Sync Status" of a ticket in sync and then selecting "Show Remote Replica".
19 3/ Add it all together like this:
20 workItem.customFields."CF Name".value = replica.customFields."CF Name".value
21 */
22
23 /*
24 Status Synchronization
25 For Status Syncing, we map the source status, to the destination status with a hash map. The syntax is as follows:
26 def statusMap = [
27 "remote status name": "local status name"
28 ]
29 Go to Entity Sync Status, put in the ticket key, and it will show you where to find the remote replica by clicking on Show remote replica:
30 def statusMap = [
31 "Backlog" : "To Do",
32 "In Progress" : "Doing",
33 "Done" : "Done"
34 ]
35 def remoteStatusName = replica.status.name
36 workItem.setStatus(statusMap[remoteStatusName] ?: "Add a default status in these double quotes")
37 */
38
39 /*
40 Area Path Sync
41 This also works for iterationPath field
42
```

## Automate your Synchronization through Triggers

Specify **Triggers** if necessary.

With triggers, you can set up automatic sync of entities that fit a specific search query.

You can create **Triggers** by editing the connection or by configuring the sync as shown in the previous step. You can also create them on the Exalate Admin Console under the **Triggers** tab.

**Note:** For more information, check out [How to create a trigger](#).

## Publish the changes made to the Connection

Click **Publish** to save a connection.

You can also use these hotkeys to publish a connection:

- **Ctrl + S** on Windows or Linux
- **Cmd + S** on Mac

**Note:** Changes in existing connections are not applied until you update the entity. Please make sure to update the entity, if you want to apply changes.

## What's Next?

Well done! You have successfully completed your "Getting Started" journey. But there's more.

Options you can consider next:

- Sit back and relax! Sync will happen based on the Sync Rules and Triggers you have set up in this step.
- You can also check out the [Configuration Guides](#) to see more synchronization examples.
- You can also check out the [Exalate API Reference Documentation](#) to learn more about how to work with incoming and outgoing sync scripts (aka Sync Rules).

### ON THIS PAGE

#### Product

[Configure the Synchronization \(Optional\)](#)

[Release History](#)

[Publish the changes made to the Connection](#)

[Glossary](#)

[What's Next?](#)

[Security](#)

[Pricing and Licensing](#)

#### Resources

[Academy](#)

[Blog](#)

[YouTube Channel](#)

[Ebooks](#)

#### Still need help?

[Join our Community](#)

[Visit our Service Desk](#)

[Find a Partner](#)