



Jira to Jira Sync: How to Synchronize Multiple Jira Instances in 8 Steps



Table of contents

Use Cases For Jira Issue Sync

- Outsourcing Quality Assurance For Developers separated by multiple Jira instances
- Setting up a Jira to Jira sync while excluding confidential information from unauthorized access
- Jira Service Desk Sync to Multiple Development Projects

How to Synchronize Multiple Jira Instances

- Step 1: Establish Workflow With Both Teams
- Step 2: Choose an Appropriate Template for your Jira sync project
- Step 3: Create a New Connection between your Jira instances or projects
 - Continue with the Basic Mode
 - Continue with the Visual Mode
 - Continue with the Advanced ScriptMode
- Step 4: Invite the other team to connect their Jira project
- Step 5: Synchronize a Jira issue
- Step 6: Customize Synchronization Rules Through Scripting
- Step 7: Create automatic synchronization triggers
- Step 8: Handle Sync Errors

Wrapping it Up



How can we integrate multiple Jira instances and projects by way of synchronization?

Jira is hands down one of the best project management and communication tools out there for development teams. But when you work with outside teams, whether they be service suppliers, clients, or outsourced teams, your collaborative Jira process can get a little sticky.

Every team has its own workflows and pipelines within Jira. If you've ever tried onboarding an outside team onto your Jira project - or vice versa - you know how difficult it can get to manage separate workflows.

However, having two completely separate Jira instances or projects is just as much of a hassle. Information gets lost. Issues get confused for other issues. And manually syncing issues from project to project is not a scalable solution.

What if you're working with a large number of teams within Jira? To have teams work together in a more structured manner, we'll have to integrate multiple Jira instances or projects.

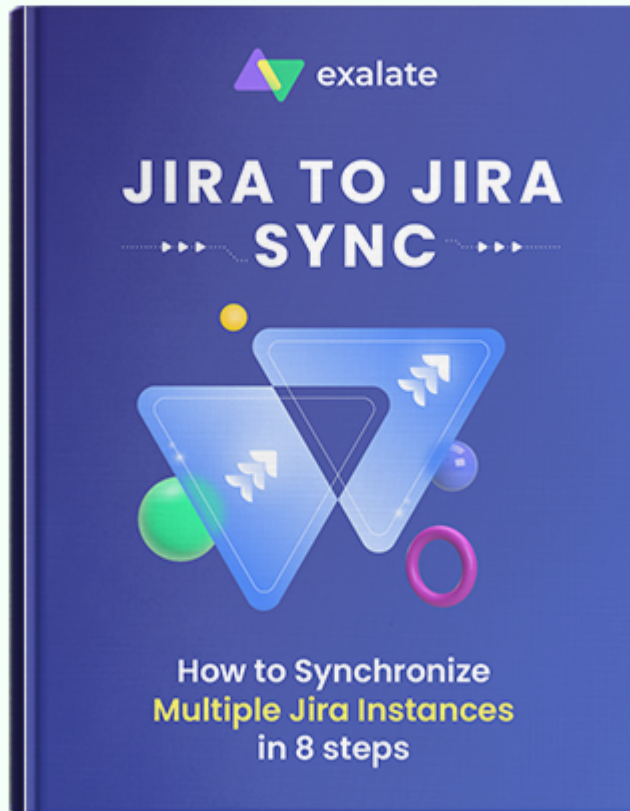
In this guide, I will explain to you exactly how to achieve a Jira to Jira issue sync, step-by-step.

Note: *We will be using the Jira sync tool [Exalate](#) for this. Exalate enables teams to efficiently coordinate their work even when tracked in multiple projects or multiple Jira's. This two-way sync automatically updates issues on both ends when a change is made.*

Here's what we're going to cover in this blog post:

- [Use Cases for Jira Issue Sync](#)
- [How to Synchronize Multiple Jira Instances](#)
 - [Continue with the Basic Mode](#)
 - [Continue with the Visual Mode](#)

- [Continue with the Script Mode](#)



Get the Jira to Jira Sync Guide

Learn how to achieve a seamless integration between multiple Jira instances, step-by-step.

[GET THE EBOOK](#)

Use Cases For Jira Issue Sync

Okay, so Jira to Jira sync sounds convenient, but how exactly can it help your project?

Here I'll dive into a few of the several use cases in which Jira issue sync can come in handy.

Outsourcing Quality Assurance For Developers separated by multiple Jira instances

Many software and game developers outsource quality assurance to third-party teams.

These outsourced quality assurance teams have their own workflows within Jira. Especially since they often have to juggle separate clients with multiple workflow styles.

A QA team needs to constantly share issues and bug reports with the development team. But the QA team doesn't require access to all of the development team's task-related issues and other development notes.

The traditional solution would be to have the dev team access both Jira instances or projects at the same time: their own project, plus the QA team's project. The latter only contains bug reports.

However, constantly needing to keep up with two Jira projects (or instances) for the same development project is a waste of time that bogs down developers.

A [Synchronization tool](#) enables both teams to stay within their own Jira projects and only sync Jira instances that are relevant to the other team. The workflow could be established as follows:

- QA tester finds a bug
- QA tester creates an issue in the QA Jira project
- The tester's team lead vets the issue and syncs it to the dev team's Jira project
- An instance of the bug is now created in the dev team's Jira project
- The Dev team updates the QA team on the status of the bug via their twin issue - all information is synchronized on the QA team's end

With this workflow, only relevant issues get communicated between both teams for a more streamlined approach to quality assurance.

Setting up a Jira to Jira sync while excluding confidential information from unauthorized access

What if you want to sync Jira issues with an outside team without leaking confidential information?

With the traditional method, two separate issues would be created in two separate Jira instances (or projects).

The issue available to the outside team would only contain non-confidential information. However, this method requires dual work every time the issue is updated.

On the other hand, Exalate enables project administrators to sync only specific fields between two Jira projects.

For example, let's say a dev team didn't want to share certain screenshots with the quality assurance team.

But what if they wanted to add these screenshots to the issue to better keep track of the bug on their end?

In this case, the administrator could set it up so that all attachments on the development side do NOT get synchronized to the QA side.

This wouldn't prevent the QA team from uploading attachments for the dev team, enabling them to share relevant information for bug reproduction.

Jira Service Desk Sync to Multiple Development Projects

Let's say you have an IT support team reporting issues with a Jira service desk when angry users come to them with problems.

In this use case, you have two Jira projects (or separate Jira instances):

1. the front-end support project
2. the back-end development project

The usual method to report bugs would be for IT support agents to forward these Jira issues via email.

Now, even if we set aside the fact that this gets developers' emails flooded when they're already busy, email is just not an effective way to share Jira issues. It's a waste of time and information will inevitably get lost both ways.

Instead, we can set up a Jira to Jira integration.

That way, IT support agents can create issues and sync them so that the development team gets instant access to the issues. This, without having to comb through their email first.

This also means that the IT support agent has visibility on the status of the issue when the developers update it, making communication with angry users much simpler.

Note: We've covered more use cases on Jira issue synchronization in the following article on [Making The Case For Synchronization Of Issue Trackers](#)

How to Synchronize Multiple Jira Instances

Now that we've gone over some use cases together, I'll show you how easy it is to set up a Jira to Jira integration.

We use our own tool, [Exalate](#), to establish a Jira issue sync. Simply because it offers the most flexibility. And it's still relatively easy to set up.

Here, I'll guide you through this process step-by-step. (And if you happen to get stuck, or have any questions, use the live chat widget on this page.)

Step 1: Establish Workflow With Both Teams

The very first step you'll need to take - before even installing Exalate into your Jira projects - is to establish the workflow you'll want to be used between both teams.

It's possible for two teams to sync Jira projects without having the same workflow. For example, you could have one project with a workflow like this:

Sync Jira workflows

And have the second team adopt a workflow like this:

Syncing of Jira workflows

No matter what you decide to establish as your workflow on both sides of the Jira project, it's important for you to agree on what information is exchanged. And how changes on one side need to be applied to twin issues on the other side.

This will avoid communication mistakes between the teams and ensure that the efficiency of the workflows is maximized.

Step 2: Choose an Appropriate Template for your Jira sync project

One of the great features that come from the Jira synchronization tool [Exalate](#) is the ability to pick from pre-existing sync templates when setting up a connection between two projects.

Sync templates establish which issue information gets synchronized, and which does not, as well as what this translates to on the other side.

For instance, you can sync the summary, description, comment, and attachment between twin Jira issues - right out of the box.

So before you set up a connection, make a list of all the relevant information you want to sync from Jira to Jira.

We'll go into more depth on how to change this later.

Jira comes in 2 different deployment models - Jira Cloud and Jira on-premise (Including Jira Data Center and [Jira Server](#)).

Note: *Atlassian announced that there's going to be no more selling licenses or new feature development from February 2, 2021. But the existing Server customers have access to maintenance and support for an additional three years, ending February 2, 2024.*

For the purpose of this document, the following steps are for Jira Cloud.

If you are using Jira on-premise check out this [documentation](#).

Step 3: Create a New Connection between your Jira instances or projects

For this step, you need to have Exalate already installed on both Jira Cloud instances.

For installing Exalate, go to the Atlassian Marketplace and type in "[Exalate Issue Sync for Jira 2 Jira & more](#)".

The screenshot shows the Atlassian Marketplace search results for 'exalate'. The interface includes a search bar with 'exalate' entered, a 'Help' button, and a sidebar with filters. The main content area displays six search results for Exalate integrations. Each result card includes the Exalate logo, the integration name, a brief description, the developer 'iDalko', a star rating, the number of installs/downloads, and a security badge (Cloud Fortified or Cloud Security Participant).

ATLASSIAN Marketplace exalate Help

Show apps for: All Products, Select a hosting type, Sort By

Search results
10 results filtered by "exalate" x

- Exalate Jira Issue Sync & more**
Two-way Jira sync tool. Supports Jira to Jira Integration, ServiceNow, Github, Zendesk, Salesforce, Azure DevOps integration...
by iDalko
★★★★☆ 80 3.7k installs
- Exalate: ServiceNow to Jira Integration**
Experience a seamless, real-time and two-way ServiceNow Jira Integration
by iDalko
★★★★☆ 3 272 downloads
- Exalate: Zendesk to Jira Integration**
Experience a seamless, real-time and two-way Zendesk Jira Integration
by iDalko
- Exalate: Github to Jira Integration**
Experience a seamless, real-time and two-way Github Jira Integration
- Exalate: Salesforce to Jira Integration**
Experience a seamless, real-time and two-way Salesforce Jira Integration
- Exalate: Azure DevOps to Jira Integration**
Experience a seamless, real-time and two-way Azure DevOps Jira Integration

FILTER BY

- Cloud Fortified apps
- Partner supported
- Free for all teams
- Beta version
- Free up to 10 users

CATEGORIES

- Integrations
- Admin tools
- Workflow
- IT & helpdesk
- Project management
- CRM

Click on download and install the app to start a free trial.

Once this is done, you're ready to create a new connection. One team will initiate the connection while the other accepts the invitation. In this example, we'll have a **Blue** Jira instance initiating the connection with the **Green** Jira instance.

For the Blue Jira, click "Apps" in the top menu and click on "Manage your apps". The "Apps" menu should give you several options on the left. If Exalate is installed you can see it in this menu. Click on "Connections".

You'll see any existing connections in this menu if you happen to have any. But if this is your first connection, this area will be empty.

The screenshot shows the Jira 'Connections' page. At the top, there is a navigation bar with 'Apps' selected and a 'Create' button. Below the navigation bar, the 'Connections' section is active. It contains a description: 'Connection defines synchronization behavior, including communication details, sync rules, and scope.' and three buttons: 'Initiate connection', 'Accept invitation', and a refresh icon. Below this is a table with the following data:

Connection	Issues under sync	Last sync	Status
? 3_to_4	1	issue UD-297 2 months ago	● Active
ADO_to_Jira	19	issue UD-328 6 days ago	● Deactivated
? epicsync_to_epic	1	issue CON-9 11 months ago	● Deactivated
gloz_to_glojscrip	3	issue UD-303 2 months ago	● Active

At the bottom of the table, there are links for 'Documentation', 'EULA', 'Support', and 'Report a bug'. Below the links, it says 'Powered by Exalate v. 5.4.9 (Core v. 5.4.9)'.

Click on the green “Initiate Connection” button.

Now you’ll fill in the URL for the destination instance (Green) you’re synchronizing with. Exalate will let you know if you’ve made a mistake in the URL or if it cannot find the destination.

You can click the “I don’t have a URL” button in case you don’t have the URL of the destination instance and then follow these [steps](#).

Initiate connection ×

Destination instance URL ⓘ

I don't have a URL

Next

At this point, you can choose one of the three ways to continue. Exalate allows you to set up your connection using 3 modes: the Basic, the Visual, and the Script Mode.

Exalate's Basic Mode enables you to set up a connection for a limited set of issue fields like *summary*, *description*, *comments*, *attachments*, and *issue types*.

The sync rules are generated automatically by Exalate and cannot be modified. These connections are recommended for use cases with basic synchronization needs.

Note: *Exalate also has an option where not only Global administrators but also Project administrators can set up Basic connections from the Project settings. This helps establish the connection on a project level. Visit this [page](#) to learn more about how to do it.*

Exalate's Visual Mode gives you an easy way to set your connection up to share what you want, at the time of your choosing. If you prefer, you can use the script mode. Script mode is a bit trickier, but lets you use more advanced programming logic.

Another advantage of the Visual mode is that it is a one-side, one admin control that lets you configure everything together.

The Script mode, by contrast, is disparate, with someone on each side of the connection controlling the sync separately and autonomously. It uses a Groovy scripting engine to allow you to customise your Jira sync however you want.

Note: *If you're already familiar with Groovy, then great! But if you're new to it, then check out chapter 4 of the [Exalate Academy](#) and learn all about it. It's quite straightforward and once you get the hang of the logic, you can play around with your sync flexibly.*



First I'll walk you through the Basic Mode, then we will talk about the Visual mode, and finally the Script Mode. But if you are interested in learning about either of the 2 directly, then you can skip ahead and jump to the relevant sections.

Continue with the Basic Mode

For getting started with the Basic Mode, click "Next" after selecting it. This will take you to a screen where you must select the project on the Blue Jira instance side initiating the connection and hit "Next".

Initiate connection ×

Select a project for the incoming sync

Exalate generates default sync rules to synchronize basic issue fields. By default the following issue data will be synchronized: summary, description, comments, attachments and issue types.

Please select the project where you want to create issues, received from the other side.*


UI Design (UD) ▾

[← Previous](#) [Next](#)


This will redirect you to a screen that will ask you whether you have admin access to the destination instance or not. Click on “Yes, I have admin access” and click “Initiate”.

Initiate connection ✕

Do you have admin access to the destination instance?

 **Yes, I have admin access**

- You will be redirected to the destination instance to establish the connection

 **No, I don't have admin access**

- You will generate an invitation and send it to the destination instance admin to establish the connection

[< Previous](#) Initiate

You will again encounter a screen on the Green Jira instance which will ask you to select a project on that side. After selecting the project, press the “Confirm” button.

This successfully establishes a connection between the 2 Jira instances.

Now you can sync your first issue by directly entering the issue key as shown below, by creating a [trigger](#), or by using the [connect](#) operation.

You can even synchronize your issues in [bulk](#).

Accept invitation ×

Select a project for the incoming sync

After accepting an invitation for a Basic mode connection, Exalate will be able to sync basic issue fields. The sync rules cannot be updated. Exalate will sync the following issue data: summary, description, comments, and attachments.

Issue types are mirrored. This means that tasks will be synced as tasks, bugs as bugs, and improvements as improvements. If an issue type is not included in the basic set, it will be synced as a task.

Please select the project where you want to create issues, received from the other side.*


Content Management (CM) | ▾

Confirm

Once the connection has been established, you can directly see the status of synchronization.

Initiate connection ✕

Congratulations!



BlueJira GreenJira

Connection is established.

[Configure Sync](#)

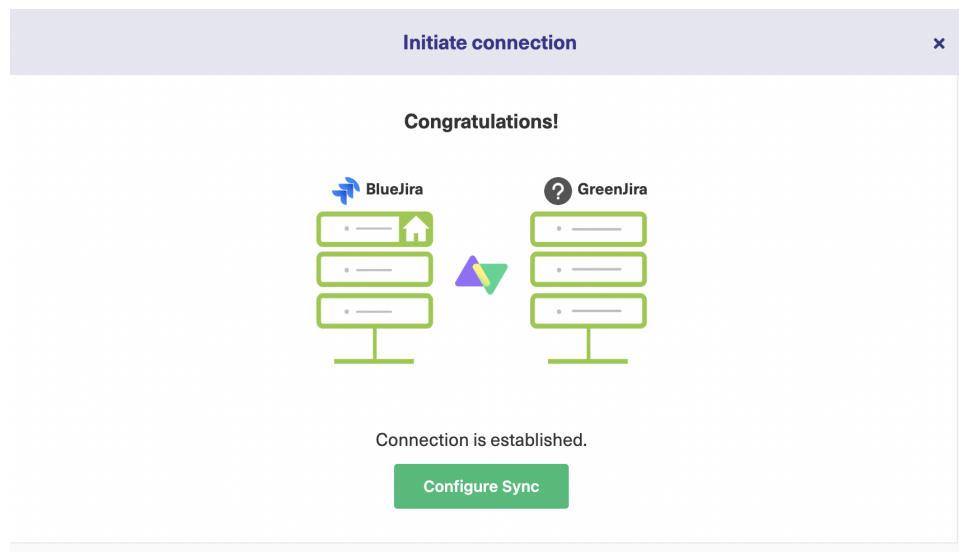
Continue with the Visual Mode

<https://youtu.be/ZWe4ncu8C5k>

If you want to use the visual mode, click it to make sure it is selected in the screen asking you to choose the configuration type, and then click “Next”.

After this, you will be redirected to a screen that will allow you to enter the details of the connection you wish to establish.

After the verification process from the destination instance (for which you need admin access), Exalate will take a few moments to get your connection set up for visual mode. When it is ready, click the “Configure Sync” button.




On the next screen, there are several fields allowing you to control the connection. Each side of the connection has its own separate fields for selecting a project and filtering entities.


The select project dropdown controls which project synchronized items will be taken from and added to.

» **BlueJira_to_GreenJira** ● Active [← Back to Connections](#) [Publish](#)

Scope 1 ————— 2 **Rules**

Define the context for the synchronization: what issues you want to sync and how to start the synchronization process.

 **BlueJira**

 **GreenJira**

Select Project

UI Design (UD) | v

Filter entities

Sync method

Manual → | v

Manual ← | v

Select Project

Software Chasers (SC) | v

Filter entities

[Next](#)

The sync method dropdowns let you choose whether synchronization is automatic, manual, or disabled. There are two of these, one for each direction of the synchronization. In each case, the items are moving from one side of the connection to the other in the direction the small arrow is pointing.

Filters

Select conditions to filter entities for synchronization:

Labels <input type="text" value="Labels"/> v	Issue type <input type="text" value="Issue type"/> v	Status <input type="text" value="Status"/> v
Priority <input type="text" value="Priority"/> v	Summary <input type="text" value="Summary"/>	Description <input type="text" value="Description"/>
Assignee <input type="text" value="Email or Account ID"/> v	Reporter <input type="text" value="Email or Account ID"/> v	

If you click the filter entities button, a pop-up with several controls will appear. You can click the “More” button to expand it and show more controls. The controls show various attributes of your entities.

Making selections in these controls, will filter the entities and determine what gets synchronized. If you want to synchronize all items assigned to a particular person, add them to the “Assignee” dropdown.

To choose entities of a specific priority, select the priority you want from that dropdown. It’s a similar process for all of them. You can combine filters too if you choose.

When you are finished, click “Save” and your filters will be applied.

Don’t forget that different filters are used for each side of the connection, so you’ll need to check the other one, too.

The screenshot shows the configuration page for a connection named "BlueJira_to_GreenJira". At the top, there are navigation buttons: "Back to Connections" and "Publish". Below this, a progress bar indicates two steps: "Scope" (1) and "Rules" (2). The main heading is "Configure the synchronization behavior with the help of field mappings and script rules." There are buttons for "Expand all", "Collapse all", and "+ Add mapping". The main content is a table with columns for "Order", "BlueJira", "Sync direction", and "GreenJira".

Order	BlueJira	Sync direction	GreenJira
1	Labels	↔	Labels
2	> Issue type	↔	Issue type
3	> Status	↔	Status
4	> Priority	↔	Priority

If you click the “Expand all” button, some of the entries will open up, and you’ll be able to see how particular values are mapped to each other. By default they are directly copied, but, for example, you could make a low priority entity on one end a high priority entity on the other, depending on the focus of the teams.

Some entries also have a button that lets you expand them individually. Not all entries can be expanded, just those that let you choose from a predefined set of values.

You can reorder entries by dragging them up and down. Each entry has its own edit button and a delete button.

For editing the default mappings, click the “Next” button to go to the rules screen (after configuring the sync scope). Or you can simply edit an existing connection by navigating to “Connections” on the main “Apps” submenu, then click on “Edit Connection” and then “Rules”.

For every connection, you can see a list of mappings that show which fields match each other. Since this is a Jira to Jira connection, the fields match identically named fields on the other side, but this may not be the case for you.

Click an entry’s edit button to bring up its edit mapping screen. There are various controls available.

There’s a sync direction button in the middle. This works like the ones on other screens, but here you can apply it on a field-by-field basis, so you can have specific fields mapped in one direction only if you want.

The screenshot shows the 'Edit mapping' interface. It features two columns for source and target instances, labeled 'BlueJira' and 'GreenJira'. Each column has a dropdown menu currently set to 'Labels'. Between these columns is a 'Sync direction' dropdown menu with a bidirectional arrow icon. Below the columns are two dropdown menus for handling 'If no matching value', both set to 'Do nothing'. At the bottom of the form are 'Cancel' and 'Save' buttons.

You can adjust the mapped field on each side of the screen. Perhaps you want the priority to be passed to the description, so the other team can prioritize it themselves while still seeing how the other team categorizes it.

You can also decide what happens if there isn't a value that matches. You can do nothing, report an error, or set a default value. That's useful if you want a way of spotting problems or absent data.

If no matching value *

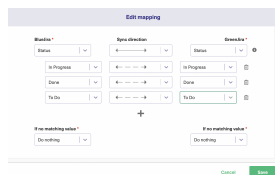
Do nothing | v

Set a default value

Report an error

Do nothing

Click the “Save” button when you’ve finished, or the “Cancel” button to ignore your changes.



For fields with multiple pre-set mappings, such as “Status” shown in the screenshot above, the screen has a list of how the individual options map to each other. There’s also a plus button, so you can add a new option mapping if you want.

Again here, you can choose the sync direction for each specific option, as well as map it to a different option, or delete it altogether.

Back on the rules screen, if you want to add a new mapping, click the “Add mapping” button at the top right. This works much like the edit mapping screen, but you have to choose the values to map from the drop-down boxes on the left and right. Once you’ve chosen them, you’ll see similar options to the other screens and can save your choices by clicking “Save”.

Add script

Use scripting to add an advanced rule. [Check examples.](#)

```
1 // sync value from "remote side text custom field" to the local "text custom field"
2 //BlueJira.issue.customFields."text custom field".value = GreenJira.issue.customFields."remote side tex
3 // Set a fixed value "Exalate" in the custom field with name "My custom field"
4 //BlueJira.issue.customFields."My custom field".value = "Exalate"
5
6 // sync value from "remote side select list custom field" to the local "select list custom field"
7 //BlueJira.issue.customFields."select list custom field".value = GreenJira.issue.customFields."remote s
8 // Set "Red" as a value in the custom fields "My select list"
9 //BlueJira.issue.customFields."My select list".value = "Red"
10
11 // labels sync
12 //BlueJira.issue.labels = GreenJira.issue.labels
13
14
```

Cancel Save

If you click the small arrow to the right of the button, you'll also see an option to add a script. If you click this, a screen will pop up allowing you to add rules using scripting. These can be simple or advanced, though if you learn how it works, you can do things that aren't possible otherwise.

When the add script pop-up first appears you'll see some sample text, as in the screenshot above. This will disappear when you click it to make your own entries.

Adding lines here lets you map fields onto each other. In the example above, you can see the text ***BlueJira.issue.labels = GreenJira.issue.labels***. BlueJira and GreenJira are the names of the Jira instances being used. This line maps the issue labels field from GreenJira directly onto BlueJira. It's a straightforward copy.

If you wanted to change to something different you could type ***BlueJira.issue.labels = GreenJira.issue.comments***, or, if you wanted to use a specific value instead, ***BlueJira.issue.labels = 'from Jira synchronization'***.

You can use comments to quickly enable or disable lines by adding "//" to the start of any line you want. Commented code is ignored until you uncomment it by removing the slashes. You can of course also use comments to make comments letting you know what the script does.

Click "Save" when you are happy with your changes.

Finally, the "Publish" button will save all your changes and apply them to your connection. When synchronization next takes place, Exalate will use your new rules.

Continue with the Advanced ScriptMode

If you decide to go with Script mode rather than Visual mode, here's what to do.

First, you need to select it and click “Next”. You will then be asked to enter the connection details as shown below.

Initiate connection ✕

Connection information

Local instance short name*

Remote instance short name*

Connection name*

Description

[< Previous](#) [Next](#)

After which you select a project on the side initiating the connection and click on “Initiate”.

Initiate connection ✕

Select a project for the incoming sync

Exalate generates default sync rules to synchronize basic issue fields. By default the following issue data will be synchronized: summary, description, comments, attachments and issue types.

Please select the project where you want to create issues, received from the other side.*

UI Design (UD) ▾

[← Previous](#)

[Next →](#)

Then, click on “Copy Invitation Code” and click on “Go to remote”.

Initiate connection ×

On the “GreenJira” side, you (or their application administrator) need to **Accept the Invitation.**

Use the following invitation code:

[Copy invitation code](#)

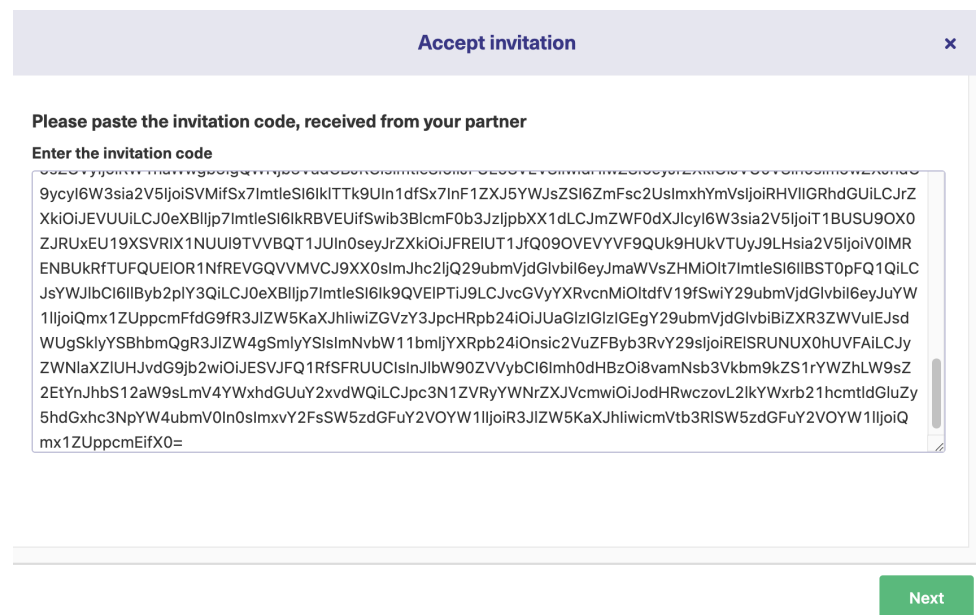
Done

Note: You can also choose to activate or deactivate the connection. When the connection is deactivated, this will pause synchronization. But changes still get queued for later updates. This means that pausing the connection will not make you lose any data.

Step 4: Invite the other team to connect their Jira project

On the second team's side (Green Jira instance), you'll need to go through the same menu. But, this time, click on "Accept Invitation" in the Connections menu.

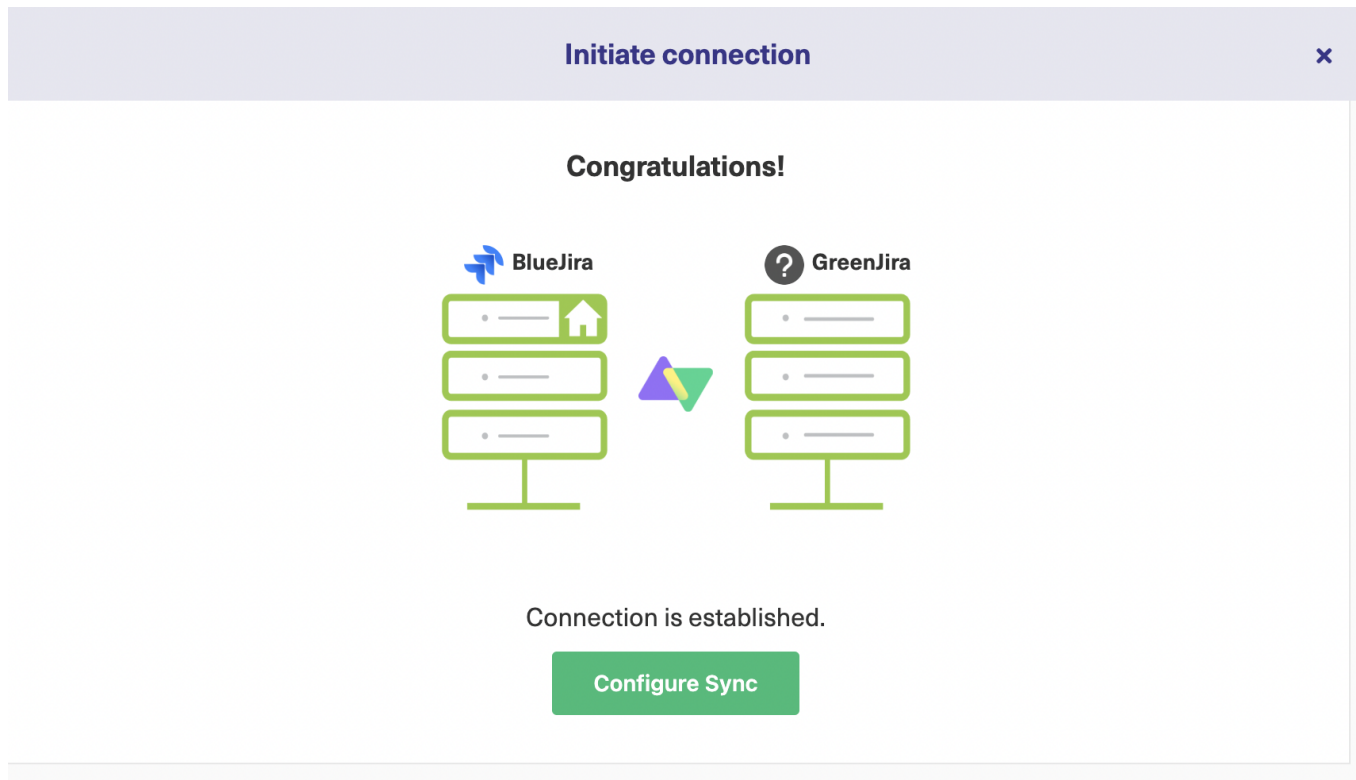
This is where you or your partner – whoever is accepting the invitation – will paste the invitation code generated in the previous step.



Once you click "Next", you'll get the same choice for selecting the project on the Green side.

Hit "Confirm" after selecting it.

This is how your established connection will look like.



That's it - have a chocolate ?!

Step 5: Synchronize a Jira issue

All right, now let's sync a Jira issue.

Now that you have installed Exalate, you get to see it under every issue action. You can find the **Exalate** button on the right sidebar on the issue view.

The screenshot shows a Jira issue view for 'UD-331' in the 'UI Design' project. The left sidebar contains fields for 'Description', 'SF Contact', and 'SF Account', all with 'None' values. Below these is an 'Activity' section with tabs for 'All', 'Comments', 'History', and 'Grid history'. A comment input field is visible with the placeholder 'Add a comment...' and a 'Pro tip: press M to comment' note. The right sidebar features the 'Exalate' section, which displays a status of 'NOT SYNCHRONIZED' and two buttons: 'Exalate' and 'Connect'. A notification banner at the bottom of the sidebar reads: 'Psst, Exalate is currently synchronizing your tasks. Find out how'.

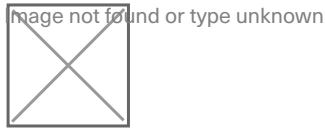
Once you click this, you'll be asked to choose which connection you want to use from a drop-down menu. Be sure to choose the appropriate one and submit your Exalate form.

You'll first see the "waiting for remote" message before it gets synced to the other side. This means your Jira issue is currently being synchronized. It should only take a few seconds for the synchronization to succeed.

You'll know an issue has been synchronized once you see a Sync Status near the bottom right of your issue.

The screenshot shows a Jira issue page for 'UD-331' in the 'UI Design' project. The issue title is 'New issue'. On the left, there are fields for 'Description', 'SF Contact', and 'SF Account', all currently set to 'None'. Below these are tabs for 'Activity', with 'Comments' selected. On the right, the Exalate interface is visible, showing a connection named 'Connection BlueJira_to_GreenJira' with a status of 'SYNCHRONIZED'. A 'Remote link' is provided as '[SC-39] New issue'. There are 'Unexalate' and 'Connect' buttons. A notification at the bottom of the Exalate panel reads: 'Psst, Exalate is currently synchronizing your tasks. Find out how'. The Exalate logo and '© Exalate 2024' are at the bottom of the page.

You can go back to the project and search for the issue which has been synchronized. The comments written on the issue that has been synchronized also appear on the other side.



Step 6: Customize Synchronization Rules Through Scripting

In this section, we are going to discuss customizing synchronization rules through scripting.

For this, we consider the case where first the comments from the Green side are synced and reflected on the Blue side. We then remove the comment sync on the Green side from its outgoing sync rules and see that the comments on the Blue side are no longer synchronized.

The screenshots below show that the comments have been synced.

From the Green side:



To the Blue side:

Projects / UI Design / Add epic / UD-331

New issue

Attach Add a child issue Link issue

Description
Desc

SF Contact
None

SF Account
None

Activity
Show: All Comments History Grid history Newest first

Add a comment...

Pro tip: press M to comment

E Exalate 2 seconds ago
Teja Bhutada commented: commented from green jira
Edit · Delete

To Do

Details

SF1	None
Assignee	Unassigned Assign to me
Labels	None
Resolution code	None
Priority	Medium
Reporter	Teja Bhutada
Automation	Rule executions
Exalate	Open Exalate

Created 3 minutes ago Updated 2 seconds ago [Configure](#)

Now, for changing the sync rules, go to your Connections menu inside the Green instance and select the connection you want to edit. Inside, pick the “Rules” tab.

This is where you have access to the way issue fields are being synchronized. (Check out the [Exalate Documentation](#) for a full overview of these fields.)

In the Outgoing Sync filter, remove the issue comments line.

» BlueJira_to_GreenJira ● Active < Back to Connections Publish

Rules Triggers Statistics Info

▼ Outgoing sync ⓘ

```

1 replica.key           = issue.key
2 replica.type          = issue.type
3 replica.assignee      = issue.assignee
4 replica.reporter      = issue.reporter
5 replica.summary       = issue.summary
6 replica.description   = issue.description
7 replica.labels        = issue.labels
8 // replica.comments   = issue.comments
9 replica.resolution    = issue.resolution
10 replica.status       = issue.status
11 replica.parentId     = issue.parentId
12 replica.priority     = issue.priority
13 replica.attachments  = issue.attachments
14 replica.project      = issue.project
15
16 //Comment these lines out if you are interested in sending the full list of versions and components of the source project.
17 replica.project.versions = []
18 replica.project.components = []
19
20 /*
```

Alternatively, you can keep the code snippet intact and add “//” in front of the line to make it into a comment.

Don't forget to apply these changes by clicking the "Publish" button.

After applying the new sync rules, the comments from the Green side are no longer synchronized. Here's the image on the Blue side. As you can see all the comments have been removed.

The screenshot shows a Jira issue page for a new issue. The breadcrumb trail is 'Projects / UI Design / Add epic / UD-331'. The issue title is 'New issue'. There are buttons for 'Attach', 'Add a child issue', 'Link issue', and a menu icon. The 'Description' field is empty. The 'SF Contact' and 'SF Account' are both set to 'None'. The 'Activity' section shows 'Show: All Comments History Grid history' and 'Newest first'. There is a comment input field with a 'Pro tip: press M to comment' note. The 'Details' sidebar on the right shows the following information:

SF1	None
Assignee	Unassigned Assign to me
Labels	None
Resolution code	None
Priority	Medium
Reporter	Teja Bhutada
Automation	Rule executions
Exalate	Open Exalate

Created 7 minutes ago
Updated 2 seconds ago

Now if you leave a new comment from the Blue side, it will be synced to the other side.

And that's it!

The synchronization rules have blocked the Green side comments from synchronizing to the Blue instance, but the Green team still has access to all Blue comments.

Now let's go back to some of our use cases.

What if we want to sync attachments to the Green side without giving access to this information on the Blue side?

We can set this up using simple scripting.

What's great is that we only need to do this on one side. So, if we want to block attachments from synchronizing to the Blue side, we can set this up on the 'Outgoing Sync' rules of the Green team.

So for this step, determine which information you want to send out through your outgoing syncs - and which incoming information you want from the other team.

In our case, we'll set it up so that we can get incoming attachments from the Blue team. But the Blue team can't access the Green attachments.

Step 7: Create automatic synchronization triggers

Triggers are used to synchronize issues automatically over the connection. You need to set certain conditions for trigger and if those conditions are satisfied, then the issues are synchronized based on the sync rules you have specified for that connection.

As seen there are already a few triggers created. But if it's your first time, you won't see any. The first trigger synchronizes issues that are assigned to "Kevin" and the second one synchronizes issues of the type "Story".

You can edit or delete the trigger under the "Action" heading. If you click the 3 dots next to the name of the trigger, you can sync existing issues meeting the trigger condition by clicking the "[Bulk Exalate](#)" option.

Click the "Create trigger" button on the top right to create a new trigger.

Add trigger ×

Specify the search query to synchronize entities automatically. All entities that fit the query will be triggered for synchronization.
[Find more details.](#)

Trigger will apply to selected entity type* ⓘ

issue | ▾

If* ⓘ

issuetype=Task

Notes

sync issues where the issue type is task

Active?

Add

First, select the entity to which the trigger is applied from a drop-down list. In our case, we have selected "issue". It means we are applying triggers to issues in Jira.

In the "If" section, specify the condition that sets the trigger off. Here, we use Jira Query Language for the conditions. It is extremely simple to understand.

As shown in the figure, we have created a trigger for issue types = "Task". So all your tasks in Jira will be synchronized over the connection. Leave "Notes" to understand the purpose of your trigger.

There is an "Active" toggle switch to activate or deactivate the trigger. To make the trigger work set it to active. Then click "Add". You can see the newly created trigger in your triggers list now.

Step 8: Handle Sync Errors

So what happens if an error occurs during a Jira issue synchronization?

Because Exalate works with a transaction-based system, any connection errors won't break the synchronization and cause conflicts between the issue and its clone.

This basically means you don't need to do any housekeeping in Exalate to fix conflicts if connection errors arise. Because there won't be conflicts at all.

If an error occurs, you can seek assistance from the Exalate team. Once the issue is resolved, your synchronization will resume in the order the changes were made.

Wrapping it Up

There you have it, 8 steps to achieving a simple Jira to Jira sync. Here's a quick recap of the basics:

- Establish a connection on one end
- Accept the invitation on the other end
- “Exalate” your new or existing issues to establish issue synchronization
- Tweak any sync rules if needed

By synchronizing Jira issues in this way using Exalate, you'll avoid losing time and information. And you'll make your workflow much simpler. Additionally, if you need separate connections with varying sync rules, you can create multiple connections between two projects with multiple sync rules.

If you have an interest in setting up sync using Exalate, you can [try it free](#). And if you need some help setting it up, we can [schedule a demo](#) on how to use it for your particular use case.

Recommended Reads:

- [Jira Integrations: Integrate Jira and Other Systems Bidirectionally](#)
- [How to set up a Jira Salesforce Integration: The Comprehensive Guide](#)
- [Jira Azure DevOps Integration: The Complete Step-by-Step Guide](#)
- [Jira ServiceNow Integration: How to Set up an Integration in 6 Steps](#)
- [The Step-by-Step Guide to Setting up a Jira Zendesk Integration](#)
- [How to Set up a Jira HP ALM Integration](#)