

How to Set up a ServiceNow Jira Integration

THE COMPREHENSIVE GUIDE



Table of Contents

04 Why Integrate ServiceNow and Jira

06 How to Choose the Right Technology for Setting up your Azure DevOps Jira Integration

08 How to set up the ServiceNow Jira integration (step-by-step process)

13 Continue with the Basic Mode

15 Continue with the Script Mode

22 Additional Information on Exalate as an Integration Solution

25 Conclusion



As companies grow, the need to integrate data between different platforms becomes more inevitable. For instance, if you're working in ServiceNow and you have a partner or a client who uses Jira, then a ServiceNow Jira integration seems to be the best solution for seamless collaboration.

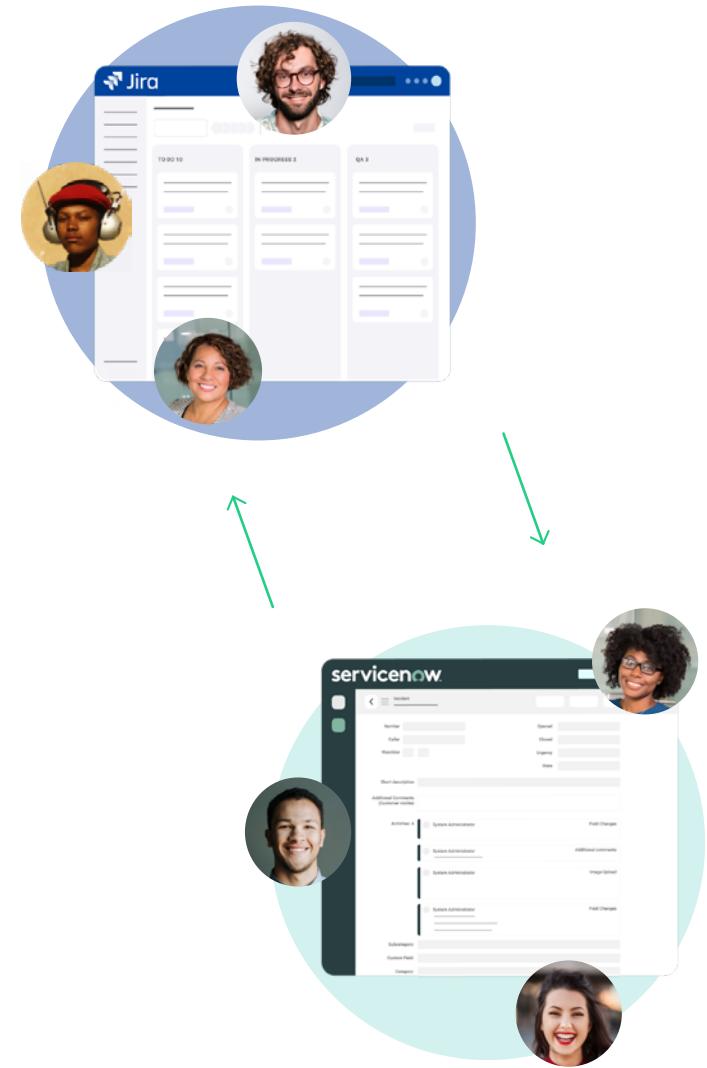
So in this guide, we will discuss the need to integrate ServiceNow Incidents with Jira Issues. (Although this process can also be applied to other entities, like Problems, Cases, Change Requests, etc.)

We will cover why admins set up the ServiceNow Jira integration in the first place, how to choose the right technology to configure the integration, next. And we will cover the step-by-step process on how to set it up.



We also have this and other tutorials in a video format

[CHECK OUR INTEGRATION VIDEOS →](#)



Why Integrate ServiceNow with Jira



Thinking of IT Service Management, ServiceNow has become a mainstream choice for CIOs & the head of IT departments to consider.

Started as an innovative niche cloud platform to manage ITSM processes based on ITIL best practices, ServiceNow has gained enormous traction in recent years. They currently have more than 20.000 customers worldwide & are growing rapidly.

On the other hand, when thinking of Agile Software development, Jira sprints in mind immediately.

As part of the Atlassian product offering, Jira manages issues & projects for Software teams. And it integrates nicely with other Software development tools of Atlassian. In short, for many Software teams, Jira is the natural choice for Issue/Project management. This is especially true when other Atlassian products are used for Software Development too.

With the above positioning of ServiceNow & Jira in the IT Service Management & Software development space in mind, the need for integration becomes obvious.

Here we've summed up two examples of possible scenarios:

- 1.** A company's IT department uses ServiceNow to provide users a One-Stop-Shop portal for all IT services. This includes reporting issues on Software. In-house Software teams use Jira to manage Issues/Projects. They generally prefer the Atlassian suite because it is used for Coding/Collaboration
 - Incidents reported by a user on ServiceNow need to be forwarded to Jira as an issue to be solved by the Software team
 - When the Software team encounters infrastructure issues, they need to be forwarded to ServiceNow as an Incident
- 2.** A company uses ServiceNow for ITSM and one or several Software packages are provided by an external software vendor. The software vendor tracks issues on the Software from all customers with Jira. Then the incident in ServiceNow needs to create an Issue on Jira and get status updates.

How to Choose the Right Technology for Setting up your Integration



When designing an integration between two ticketing tools, three aspects always need to be considered:

1 Decentralized Integration (autonomy)

The ticketing tools at each end of the integration have the means to control what information is sent to the other side and how incoming information is interpreted. Changes in the ticketing tools shouldn't break the integration. Rather they should be easily reflected in the integration.

2 Reliability

A reliable integration is one that always works for the user – even when the other side is not available for whatever reason (such as maintenance). Operational maintenance capability is important to ensure always-on integration.

3 Flexibility in the Configuration

The integration is able to bridge the differences between the two systems. To be able to effectively attribute mapping is the first mandatory step. Having the flexibility to align the process differences between the teams/organizations – a second

Exalate meets all of these requirements and is the tool we'll use for this tutorial

[START NOW](#) (it's free)

In the chapters below we will show you how Exalate addresses the above aspects in more detail. But first, let's go through the step-by-step process of how to set up the integration!

How to Set up a ServiceNow Jira Integration (a Step-by-Step Process)



Step 1

Install the Exalate app on your ServiceNow Instance

To install Exalate on your ServiceNow instance, you'll have to use an "update set".

You can find the step-by-step instructions of the Exalate agent installation for ServiceNow on the following Exalate documentation page: [Installing the Exalate app on your ServiceNow instance](#)

Step 2

Install the Exalate app on your Jira Instance

In order for Exalate to work, it needs to be installed on both sides of the integration. This means you'll also have to install Exalate on Jira. This will be a straightforward process like installing a Jira app.

You can find the step-by-step instructions for installing the Exalate agent on your Jira instance here:

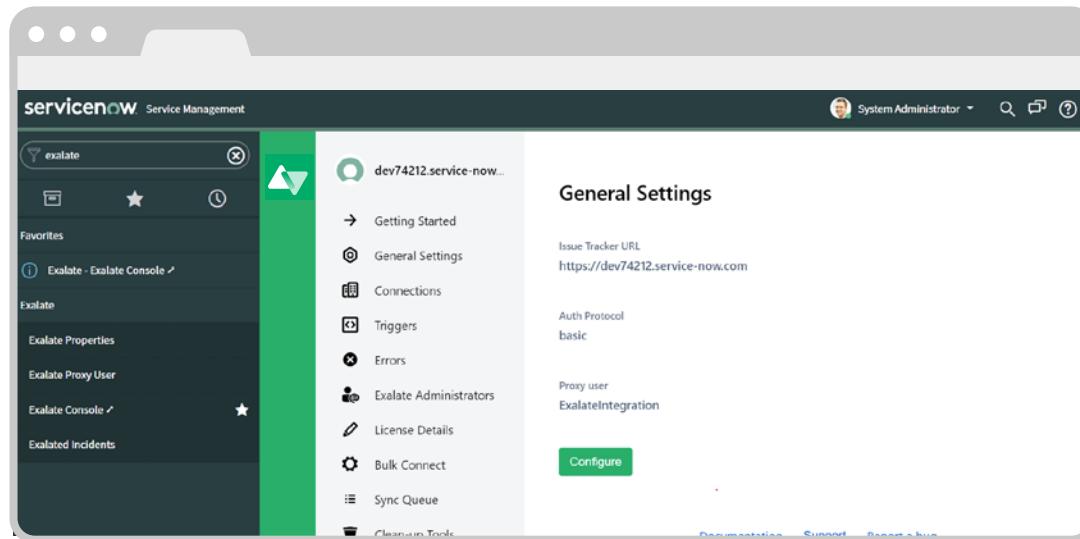
- [For installing on Jira Cloud](#)
- [For installing on Jira on-premise](#)

Step 3

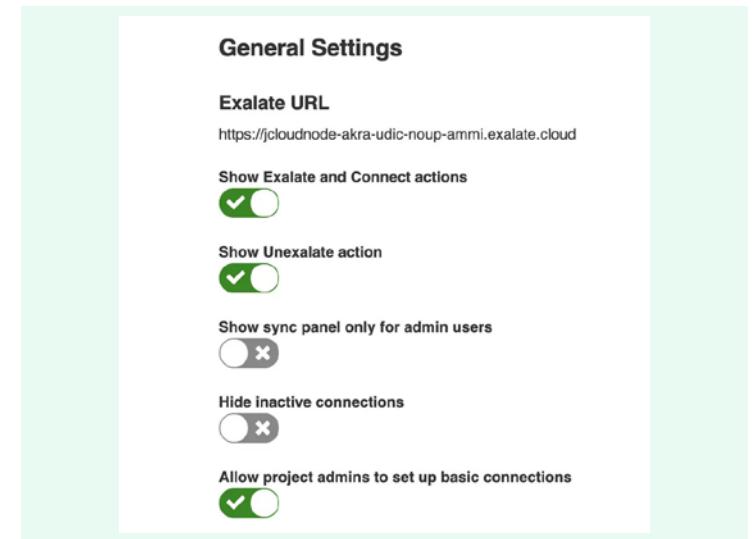
Have a Quick Look at the Exalate Console

The Exalate console provides a user interface for system administrators to configure & maintain the integration.

1 After installing, the Exalate console should be directly accessible as an Application on the ServiceNow instance:



2 At the Jira side, similar configuration options are provided as an Application as well:



With the Exalate console you can, on the one hand, create/maintain your configuration. You can also view Sync Queues and check/resolve errors.

These capabilities will help to maintain the integration efficiently.

But, let's move on to setting up a connection between your Jira and ServiceNow instance.

Step 4

Establish a Connection between Jira and ServiceNow

Once the Exalate agent is installed on both ServiceNow and Jira, you need to set up connections between the two Exalate agents.

Either side can initiate the Connection or else accept the Connection invitation from the other side.

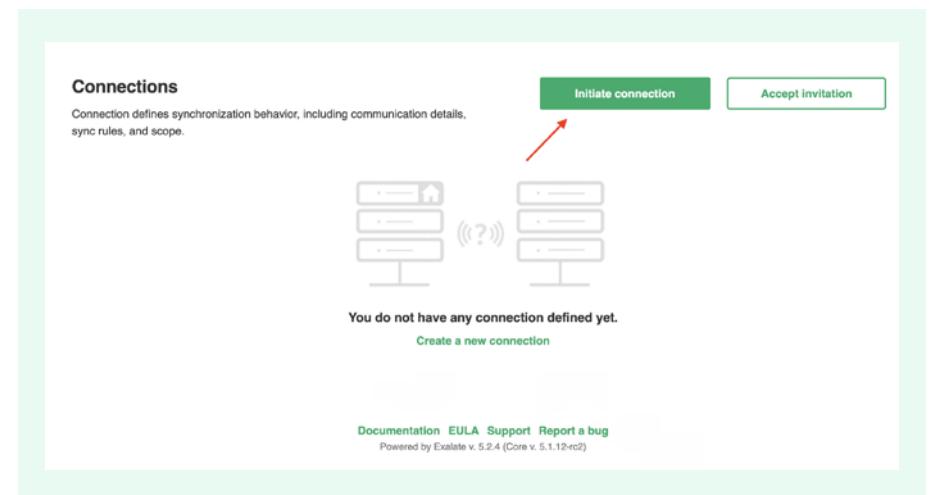
Below you have step-by-step instructions to set up the connection between the two systems:

- [Configure Connection on Exalate ServiceNow agent](#)
- [Configure Connection on Exalate Jira Cloud](#)
- [Configure Connection on Exalate Jira on-premise](#)

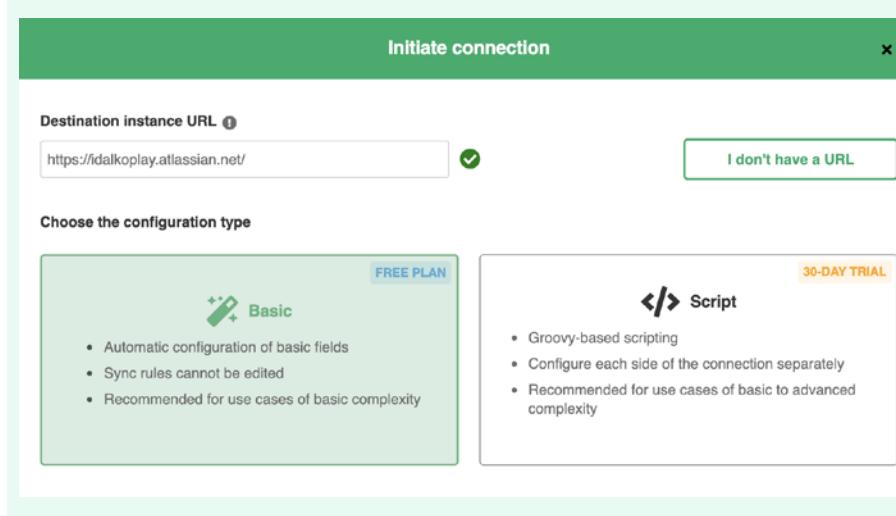
But here's a recap of what it boils down to.

You'll first have to "Initiate connection" in the Connections tab in the Exalate console.

You will then be prompted to enter the URL of the destination instance. Assuming we have initiated the connection from the ServiceNow side, the destination URL will be that of Jira.



Next, you have to select the configuration mode for your connection.



Exalate supports 2 configuration modes:



Basic Mode: allows you to establish a connection with pre-existing sync rules. These cannot be configured or changed. It is suitable for simple integration use cases. Exalate offers a [Free Plan](#) so that you can experience it firsthand. The Free plan only supports the Basic mode.



Script Mode: To use the full functionality of Exalate with advanced features and configurations, we recommend you use the Script mode. You can configure it to suit your unique integration case. You can even upgrade from the Free Plan to the Premium Plan in Jira and in ServiceNow anytime you want!



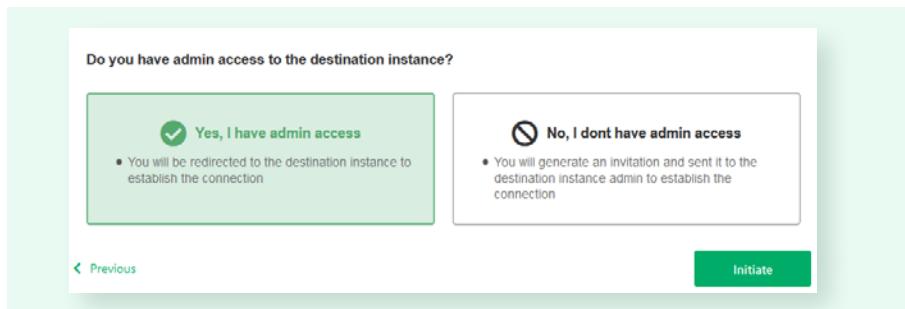
Click on the mode to jump to the relevant sections



Continue with the Basic Mode

- Click "Next" when you select "Basic" on the screen above.

You will be required to verify admin access to the destination side, Jira in our case. Click "Yes, I have admin access" since we have the access already.

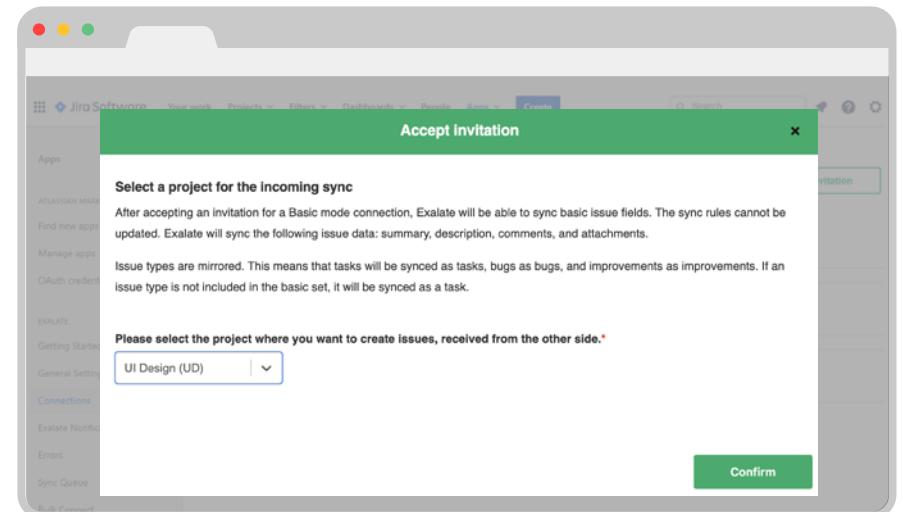


If you don't have access, an invitation code will be generated. You will need to copy and paste the code in the "Accept Invitation" into the Jira instance.

- Exalate then performs a quick check to verify the access. Once the verification is successful, it will automatically redirect you to the destination side (Jira).

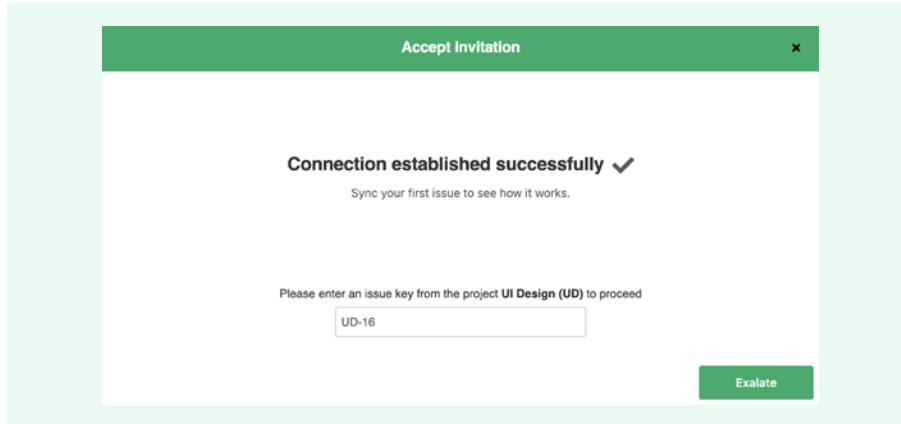
Here select the project on the Jira side that you want the Incidents to synchronize into.

After selecting the appropriate one from the drop-down list, click "Confirm".



3

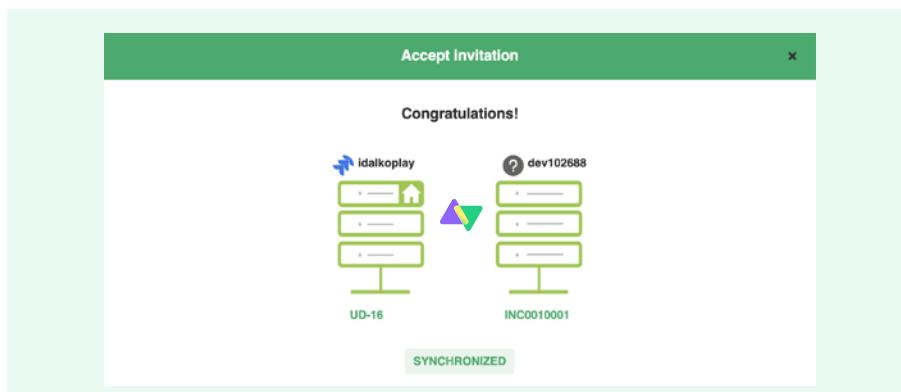
Enter the issue key to start your first synchronization.



4

If you navigate to the ServiceNow side right now, you will be asked to enter the Incident number for synchronization. In either case, press the "Exalate" button.

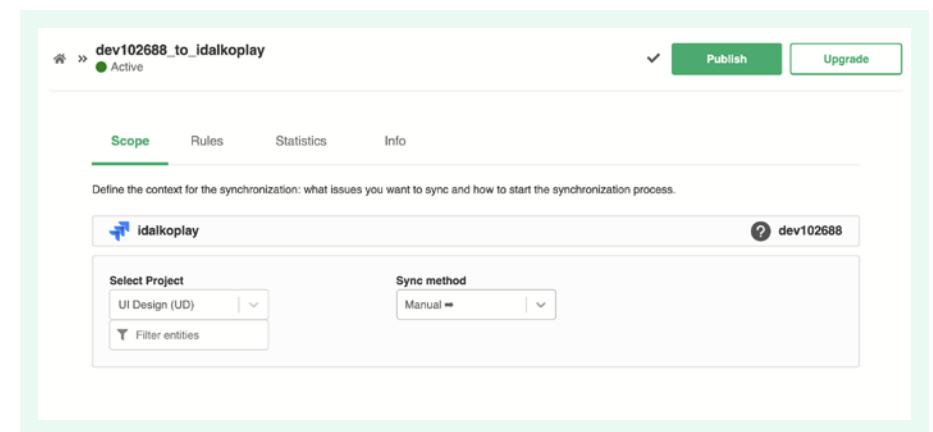
Wait for some time for successful synchronization.



5

You can go to the synchronized Issue or Incident by clicking the respective links shown in the image on step 4.

The Basic mode connection just created can be viewed in the "Connections" tab. If you click the edit connection icon in front of its name, you can see the default mappings. You can choose to upgrade the connection if you want.



Connections created using the Basic mode are simple and you can sync issues or incidents as shown above. You can even create [triggers](#) to start automatic synchronization or choose to sync in bulk using the "[Bulk Connect](#)" option.

Let us now have a look at the Script mode



Continue with the Advanced Scripting Mode

In this mode, one side sends an invitation code and the other side accepts it. Let us see how it works.

1. Send an invitation code

1 Name the connection. For this, provide the local instance name(ServiceNow in our case) and also the remote instance name(Jira). Exalate automatically generates a name for you, but you can change it if you want to. Don't forget to add a description to the connection. This will help you identify the connection when you have a lot of them.

Click "Initiate" when you are ready.

Initiate connection

Connection information

Local instance short name* ServiceNow

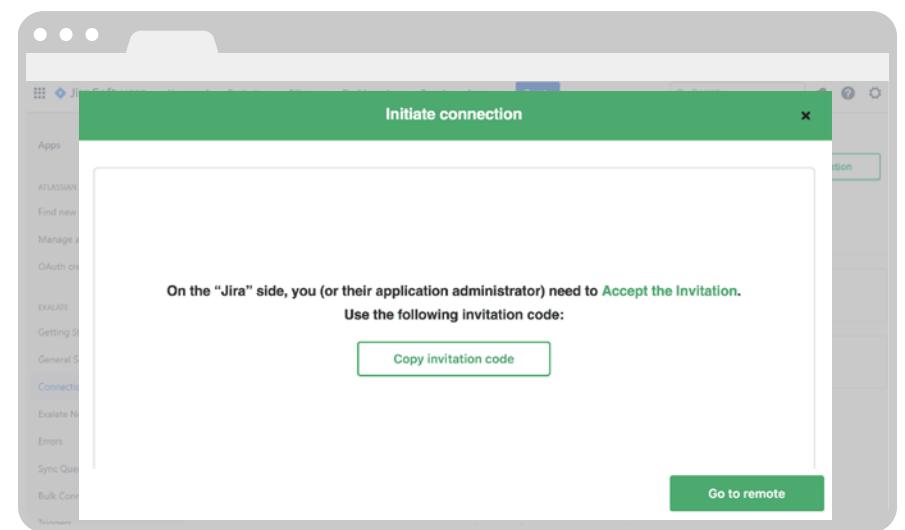
Remote instance short name* Jira

Connection name* ServiceNow_to_Jira

Description
This is a connection between ServiceNow and Jira

Initiate

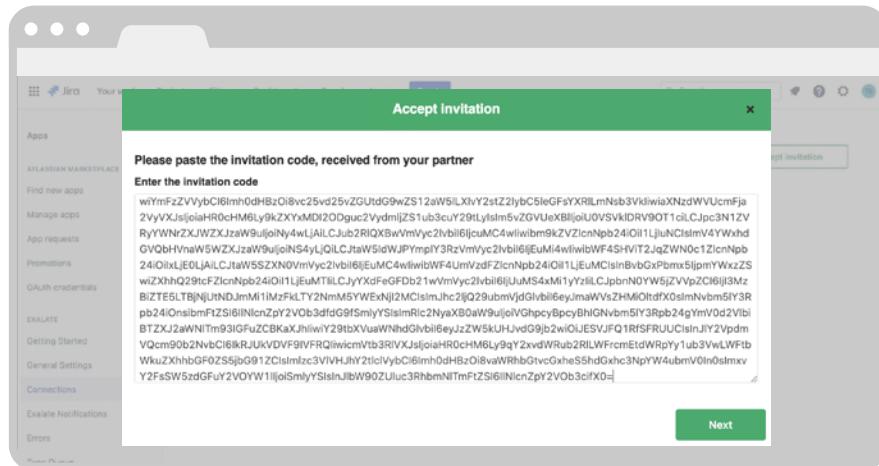
2 This will generate an invitation code. You'll have to copy this so the other side(Jira) can accept the invitation.



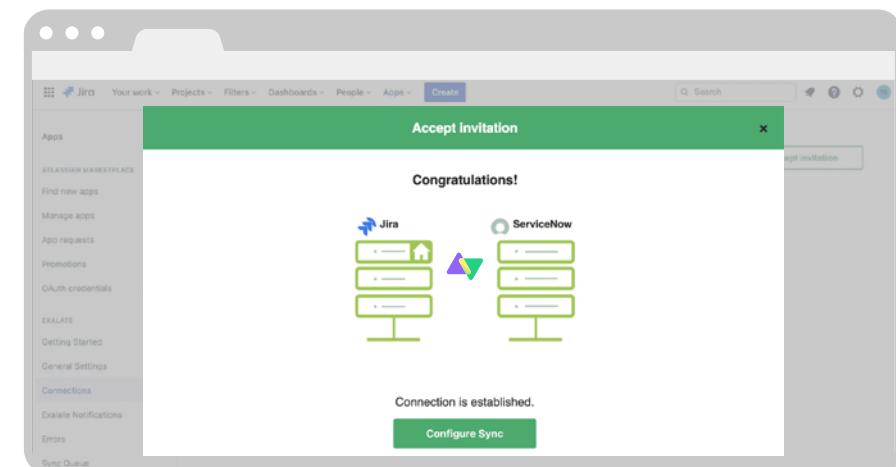
So, move over to your Jira instance either by clicking on "Go to remote" on the screen above or by navigating to "Connections" in the Exalate console in Jira.

2. Accept the invitation code

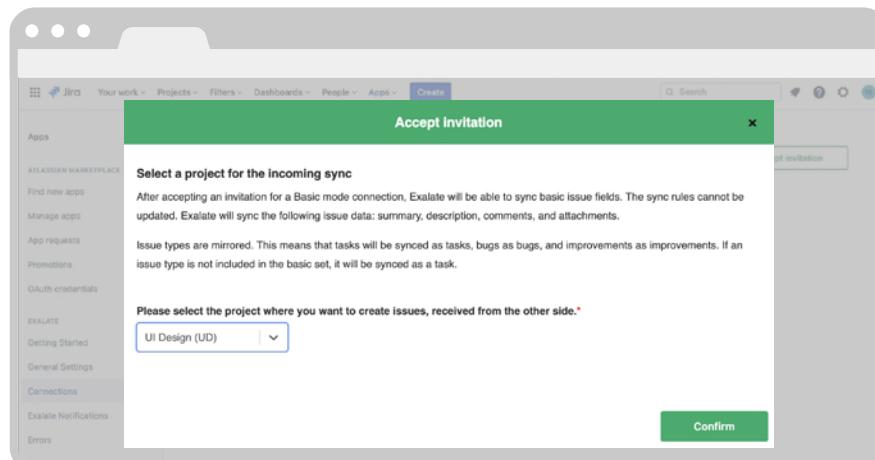
- 1** Click "Accept Invitation". Go ahead and paste the code there:



3 After this, the connection has been established successfully.



- 2** On the next screen, choose the project to work with from the dropdown menu. Then click "Confirm" to create the connection on Jira.



You'll be able to configure the sync rules for each side separately, from the other side. This has been done on purpose, so each side will remain autonomous.

You can choose to configure it immediately by clicking on "Configure Sync" or you can configure this later in Step 6.

After you've accepted the invitation and a connection has been established, we can move on to setting up a rule that will serve as a trigger for the synchronization.

Step 5

Configure your Synchronization Triggers to Determine when to Sync

Once a connection between ServiceNow and Jira is established, the main work of integration can start.

1 Have both sides agree on synchronization rules

At this stage, close cooperation between the Incident/Issue manager is needed to determine when an Incident at the ServiceNow side needs to create an Issue at the Jira side or vice versa.

The agreement can be defined on ServiceNow Exalate & Jira Exalate independently, allowing all possible scenarios. However, it's also possible that you're an admin on both sides.

To create triggers, click the "Triggers" tab in the left-hand Exalate console. If there are any existing triggers, they will be listed here, but the first time you use them, they will be blank. Click the "Create trigger" button to get started.

On the "Add trigger" pop-up that appears, there are several fields you can interact with. The entities you want to apply triggers to can also be selected now. The entity selection will differ from platform to platform.

You can then set the conditions that cause the trigger to activate. In Jira, triggers are written using the JQL query language. You can read more about that [here](#). ServiceNow search syntax is used for the ServiceNow side. Click the "Active" button to activate the trigger and then click "Add".

PRO TIP

Want someone to help you get started with Exalate?

Our partners are located all over the world in more than 50 countries. They will help you with implementation, support, and license management for Exalate. They can give a demo, prepare a PoC, and handle almost any complex use-cases.

[FIND A PARTNER](#)



Set up an automated synchronization trigger

If the process managers have determined that whenever an Incident is assigned to an Assignment group called Jira ServiceDesk, an Issue needs to be created at the Jira side.

The Trigger defined in Exalate ServiceNow looks like the following:

Triggers						Create Trigger
Entity type	When	If	Then sync via Connection	Status	Action	
incident	Events: create/update	active=true^assignment_group=348313a81be000103d9ea9b4bd4bc bc6 If incident is assigned to Jira Service Desk	ServiceNow_to_Jira	<input checked="" type="checkbox"/>	...	

If, at the same time, they also agree that whenever an Issue in Jira has a Label equal to ServiceNow, it will create an Incident in ServiceNow for teams on ServiceNow to solve.

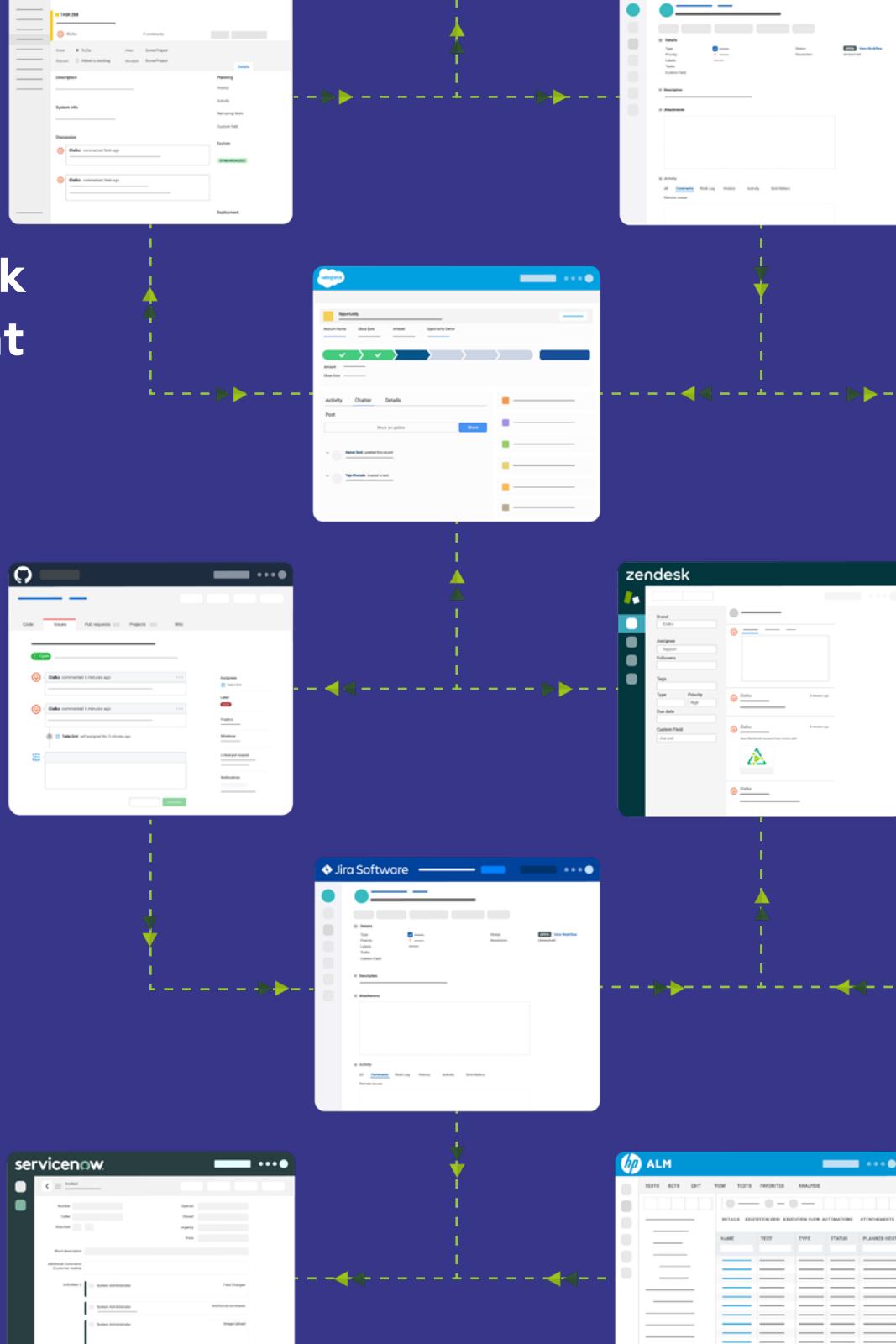
The Trigger defined in Exalate Jira looks like the following:

Triggers						Create Trigger
Entity type	When	If	Then sync via Connection	Status	Action	
issue	Events: create/update	labels = ServiceNow Issues for ServiceNow	ServiceNow_to_Jira	<input checked="" type="checkbox"/>	...	



Bi-directionally connect your work across multiple work management systems in real-time

- A frictionless collaboration across internal teams and outside company borders.
- Complete autonomous control over both outgoing and incoming information
- Maximum security due to complete control over the data being shared with the other systems
- Limitless customizability to fit your unique and complex integration needs
- Set up a synchronization between multiple Jira instances, Salesforce, Azure DevOps, ServiceNow, Zendesk, Github, HP ALM & more



[BOOK A DEMO](#)

[TRY IT FOR FREE](#)

Step 6

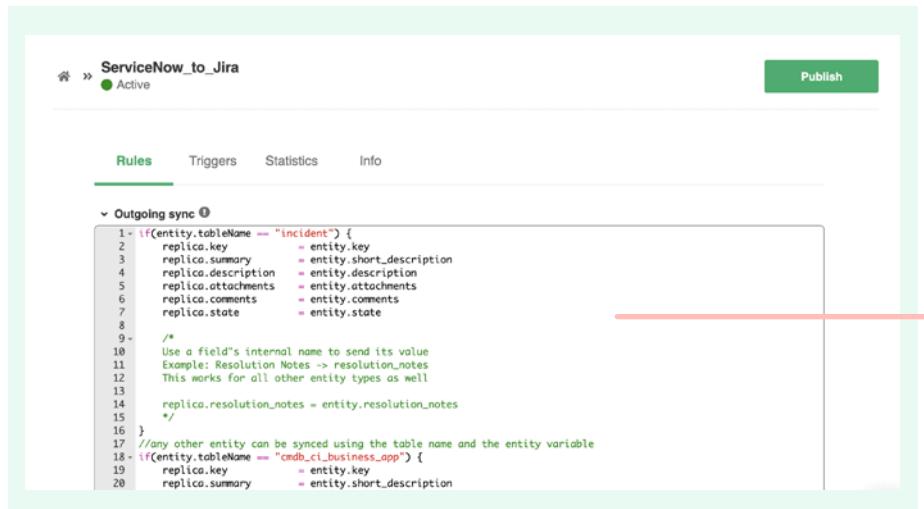
Configure your Connection to Determine the Information to Send

Once an Incident on ServiceNow fulfills the conditions defined by the Trigger, the ServiceNow Exalate will receive access to the Incident through the REST API.

1 Configure the outgoing sync

What information is sent to the Jira Exalate is defined in the Connection Sync Rules -> "Outgoing sync".

Here's a screenshot of what it looks like:



```

ServiceNow_to_Jira
Active
Rules Triggers Statistics Info
Outgoing sync
1- if(entity.tableName == "incident") {
2-     replica.key      = entity.key
3-     replica.summary  = entity.short_description
4-     replica.description = entity.description
5-     replica.attachments = entity.attachments
6-     replica.comments  = entity.comments
7-     replica.state     = entity.state
8-
9- /*
10- Use a field's internal name to send its value
11- Example: Resolution Notes -> resolution_notes
12- This works for all other entity types as well
13-
14- replica.resolution_notes = entity.resolution_notes
15- */
16}
17 //any other entity can be synced using the table name and the entity variable
18- if(entity.tableName == "cmdb_ci_business_app") {
19-     replica.key      = entity.key
20-     replica.summary  = entity.short_description

```

- replica.<attribute> represents the message attributes. In our case, it represents Jira Exalate.
- entity.<attribute> represents the local record attributes. In our case, it represents ServiceNow Incidents.

The above example is an out-of-box, straightforward mapping. However, more complex mapping can be defined using groovy scripts in this section as well. Exalate provides a number of [Script Helpers](#) to reduce the effort to script yourself.



Configure the incoming sync

The incoming sync will determine how to interpret the information received.

The rules on how to interpret the incoming data are configured in the Connection Sync Rules as well.

On the ServiceNow Exalate, there is a distinction for when an Incident is created or updated.

In the example shown below, if it's the first sync then an incident will be created. Otherwise, details like summary, description, attachments, and comments of an Incident already present will be updated.

```
✓ Incoming sync ⓘ
1- if(firstSync){
2-   //Decide on the first sync, which entity you want to create based on the remote issue type
3-   entity.tableName = "incident"
4- }
5-
6- if(entity.tableName == "incident") {
7-   entity.short_description = replica.summary
8-   entity.description = replica.description
9-   entity.attachments += replica.addedAttachments
10-  entity.comments += replica.addedComments
11-
12-  /*
13-   Jira Custom Field to ServiceNow Field
14-   Apply the value from a Jira custom field to the Resolution Notes
15-   This works for all other entity types as well
16-
17-   entity.resolution_notes = replica.customFields."Jira CF Name".value
18-  */
19-
20-  /*
21-   Status Synchronization
22-
23-   Sync status according to the mapping [remote incident status: local incident status]
24-   If statuses are the same on both sides don't include them in the mapping
25-   def statusMapping = ["Open":"New", "To Do":"Open"]
26-   def remoteStatusName = replica.status.name
27-   entity.state = statusMapping[remoteStatusName] ?: remoteStatusName
28-   */
29- }
30-
31- //any other entity can be synced using the table name and the entity variable
32- if(entity.tableName == "cmdb_ci_business_app") {
33-   entity.short_description = replica.summary
34-   entity.description = replica.description
35- }
```

Just like with outgoing sync rules, more complex mappings can be scripted.

Below is an example of mapping the ServiceNow Incident States with the Jira Issue Status. (Again Exalate Script Helpers can help reduce the scripting effort.)

```
1  def statusMapping = [
2-
3-   // Jira issue status <-> SNOW incident state
4-   "To Do" : "New",
5-   "In Progress" : "In Progres",
6-   "Done" : "Canceled"
7- ]
8  def defaultStatus = "New"
9  def statusName    = statusMapping[replica.status?.name] ?: defaultStatus // set default status
10 issue.status.name = statusName
```

Additional Information on Exalate as an Integration Solution

Since we used Exalate to set up the Jira ServiceNow integration, you might have some questions about this solution.

Here I'll explain a bit more about Exalate's architecture and security.



On Autonomy in Architecture

The basic architectural setup of Exalate as an integration enabler between two systems is depicted below:



In our scenario Tracker (Blue) would be your ServiceNow instance, Tracker (Red) would be the Jira Server or Cloud instance.

Tracker(Blue) and Tracker(Red) have a separate Exalate Agent dedicated to your ServiceNow/Jira.

- Letters A - F depicts the information flow between the systems. ServiceNow and Jira communicate with each other through the dedicated Exalate agent, keeping the autonomy of your system
- Your (ServiceNow connected) Exalate Agent controls what information is sent and how incoming information is mapped.
- Exalate agent is available for ServiceNow instance, Jira Cloud, Jira On-Premise, Salesforce, HP ALM/QC, Github, Zendesk, and more to come



*“*The best feature is definitely its flexibility. We can do a lot of scripting which we use a lot at this moment. The scripting makes Exalate as flexible as a solution can be.

CHRISTOF CUYPERS | GANTNER

[READ FULL STORY →](#)



*“*Take your time to dive into what's possible. There's so much more under the hood that you could do - if you keep it simple and use the defaults, it's only 5% of what you can do, you could build crazy and cool stuff.

MARTEEN CAUTREELS | DPG MEDIA

[READ FULL STORY →](#)

On Security

Exalate is an intelligent middleware that transports data between ServiceNow & Jira.

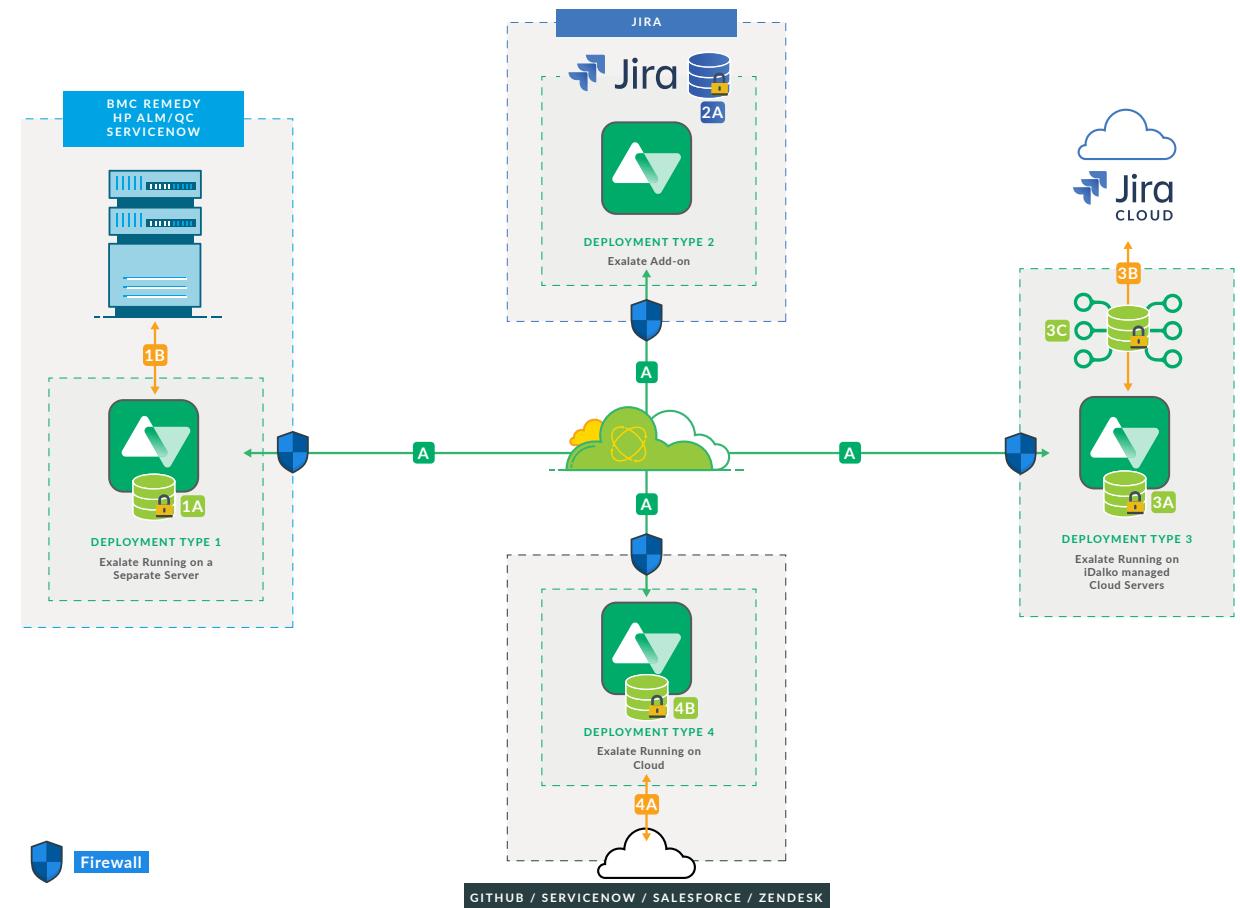
The Security consideration is described in great detail in the following whitepaper, free for download:
["Exalate Security and Architecture Whitepaper"](#).

The image on the right explains the different deployment models Exalate is supporting.

Exalate for ServiceNow can be deployed either in the cloud or on-premise.



To learn more about how Exalate keeps data safe, read this free
[Exalate Security and Architecture Whitepaper](#)



This image shows the relationship between Exalate and the services it supports.

Comparison between Exalate and the IntegrationHub with the Jira Spoke

ServiceNow itself has a capability called IntegrationHub that allows reusable integrations with third-party systems and calls them from anywhere in the ServiceNow platform.

Using the IntegrationHub requires you to get a separate subscription.

The Standard package of IntegrationHub includes a Jira Spoke with a (limited) number of Actions that allow ServiceNow to manage issues, users, stories, and groups in Jira. And it retrieves Jira data to use in a flow.

Jira spoke V3.0.5 uses bi-directional webhooks and subscribes to Jira with a ServiceNow callback URL.

So, why use a third-party tool, like Exalate, to set up the integration?

Below I've added some considerations:

1. Use Exalate when

- The integration is bi-directional. Then both sides will be able to trigger the integration and decide on the information to be exchanged
- The integration is point-to-point between the two systems.

2. Use IntegrationHub when

- ServiceNow has the orchestration role to create an automated flow, involving several systems.
- ServiceNow controls the trigger of integration, remote system (in our case Jira) is the receiver of the commands

Conclusion



Many ITSM-teams adopt ServiceNow as a best-in-class solution.

While development and Project Management teams still widely consider Jira as the ideal solution.

As efficient collaboration is crucial for any effective organization, integrations between these systems are increasingly common.

When considering the right technology to set up the integration, it's important to keep into account the following three aspects:

1. Does each side have **autonomy** over outgoing and incoming information and how it is interpreted?
2. Is the solution **reliable** enough to continue working, independently from changes to the other side of the integration?
3. Is the solution **flexible** enough to cover a wide variety of use cases and evolve with your way of working?

In this guide, we've set up the integration using Exalate. And we've described the step-by-step process on how to complete this integration.



“The best feature is definitely its flexibility. We can do a lot of scripting which we use a lot at this moment. The scripting makes Exalate as flexible as a solution can be.

CHRISTOF CUYPERS | GANTNER

[READ FULL STORY →](#)



“Take your time to dive into what's possible. There's so much more under the hood that you could do - if you keep it simple and use the defaults, it's only 5% of what you can do, you could build crazy and cool stuff.

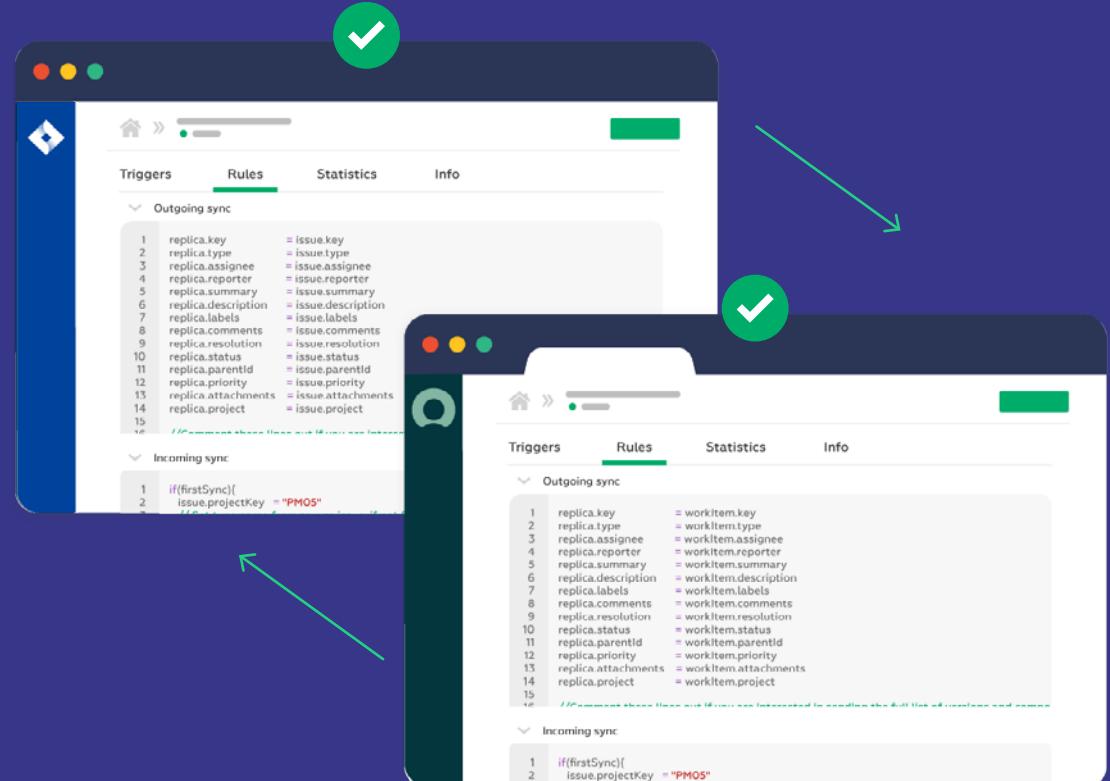
MARTEEN CAUTREELS | DPG MEDIA

[READ FULL STORY →](#)



The Leading Cross-Company Integration Solution

Exalate is the leading and the only solution specifically built for cross-company integration scenarios with the most complex synchronization use cases.



“The first thing that hits you when using this app is how easy it is to get a synchronization up and running. By default, everything works so well.

“With the amazing support team and the flexible capability of Exalate, we were able to see it work nicely and become an important part of our work life cycle.

“Exalate is marketed as the most flexible synchronization tool for issue trackers and that short definitions is pretty accurate from what we have observed.

[BOOK A DEMO](#)

[TRY IT FOR FREE](#)

LEARN MORE



[DOWNLOAD THE EBOOK](#)



[DOWNLOAD THE EBOOK](#)



[DOWNLOAD THE EBOOK](#)