



How to Set up a Salesforce ServiceNow Integration: The Complete Step-by-Step Guide



Table of contents

Why Integrate Salesforce and ServiceNow

- Salesforce
- ServiceNow
- Why Do You Need a Salesforce ServiceNow Integration?

The Right Tool for Setting up a Salesforce ServiceNow Integration

How to set up a Salesforce ServiceNow Integration

- Step 1: Install Exalate on Salesforce
- Step 2: Install Exalate on ServiceNow
- Step 3: Set up a Connection between Salesforce and ServiceNow
- Step 4: Configure your connection to determine what information gets shared
- Step 5: Create Automatic Synchronization Triggers
- Step 6: Start Synchronizing the Information

Common Use Cases

- Customer Support and Sales Teams
- Customer Success Representatives and Sales Teams
- Development and Sales teams

Conclusion



If your teams use powerful platforms like Salesforce and ServiceNow to increase productivity and efficiency, a Salesforce ServiceNow integration can help them gather and collate their ever-increasing information in a better manner. It will help them share important and accurate information with each other, automatically and with the least ado.

You can allow teams to stay on their own platform and at the same time let data be shared bi-directionally with other teams. This guide walks you through why a Salesforce ServiceNow integration can be beneficial and how you can choose the right tool for achieving it. Then it goes on to discuss how such an integration can be configured and implemented end-to-end.

So let's begin!

Here's what we'll cover in this blog post:

- [Why Integrate Salesforce and ServiceNow](#)
- [The Right Tool for Setting up a Salesforce ServiceNow Integration](#)
- [How to Set up a Salesforce ServiceNow Integration](#)
 - [Continue with the Basic Mode](#)
 - [Continue with the Visual Mode](#)
- [Common use Cases](#)



Get the Salesforce ServiceNow Integration Guide

Learn how to integrate Salesforce and ServiceNow, step-by-step.

[GET THE EBOOK](#)

Why Integrate Salesforce and ServiceNow

Salesforce



Salesforce is a leading CRM (Customer Relationship Management) platform. It aims towards bridging the gap between customers and companies. It provides a range of services and apps to different departments in the company like sales, marketing, IT, and the like to help them enrich and enhance their relationships with customers.

ServiceNow

servicenow

ServiceNow is a cloud-based platform that automates workflows. It helps streamline and automate routine work tasks reducing the time spent on administrative work. It also assigns and prioritizes incidents raised, based on agent availability and skill level. It has its roots in ITSM (IT Service Management).

Why Do You Need a Salesforce ServiceNow Integration?

Both these platforms are capable of handling huge amounts of information, beneficial for the teams to do their tasks effectively.

This information, if shared between them, can help them enhance their overall business objectives. Customer service teams using ServiceNow can benefit from customer overviews, feedback, and queries from Salesforce. And the sales team using Salesforce can keep track of incidents reported by important customers.

Sometimes the same information both teams need exists on different platforms, leading to duplication and inefficiency.

So sharing this information will help different teams gather and collate important information within their own platforms. But if done manually, it can be time-consuming, error-prone, and can even lead to incorrect or duplicate information.

If this is all done automatically and in real-time, it can help teams deliver better customer service and an end-to-end enhanced customer experience. The teams won't even have to leave their platforms or switch between applications to get information exchanged. A Salesforce ServiceNow integration can help teams achieve all this with the least possible fuss.

With such integration in place, relevant information can also be filtered and passed between teams. For instance, a sales team might only be interested in seeing the status of incidents raised by key customers and not in other technical details of the incident. Similarly, the support teams might not want to view personal information about the customers, but only the technical details, if any, that can help them solve their incidents faster.

When implementing a ServiceNow Salesforce integration, it is important to choose the right tool for your scenario. This can help you realize the benefits of your integration in a better manner.

The Right Tool for Setting up a Salesforce ServiceNow Integration

It is crucial that your tool understands your integration scenario well and adapts to it seamlessly as every integration case is unique.

Security is also an important aspect of information exchange. So look for tools that have inherent security mechanisms like encrypted file transfer, secure transfer protocols, role-based access controls, and the like.

Tools supporting decentralized integration can help both sides of the integration control what is being shared with the other side and how to interpret information coming from the other side. This ensures that both sides have autonomy and can work independently.

The integration tool must also be reliable to address issues related to downtimes and system failures without manual intervention. Changes must be queued and applied in the same order as their initiation, without the integrating parties noticing the outage.

You might also want to integrate with your other partners, suppliers, customers, or vendors in the future. Having a tool that already integrates with the applications they use will make it easy for you to make the transition.

All these features are supported natively by [Exalate](#) and that is the reason why we choose it for our Salesforce ServiceNow integration.

Exalate has the flexibility to adapt to your unique integration case with the help of its scripting engine. So you don't need to spend a lot of time configuring your tool to fit your case, you simply change the sync rules and get going. We will see how to do this in a while.

Note: *Want to learn all about Exalate's security and architecture? Read [this whitepaper](#).*

Exalate supports a distributed integration, where each side controls what information is sent and received independently without any interference from the other side.

It has a transactional synchronization engine that allows applying changes queued for synchronization in the same order as their initiation without manual intervention, even if there's downtime or your system is being upgraded or your firewall is being configured.

It also comes with integrations for Jira (on-premise and cloud), Github, Zendesk, Azure DevOps, and HP QC/ALM. So if these are the applications that the companies you want to integrate with use, then you can simply extend your Exalate network.

This was just a head start. Let's see how a Salesforce ServiceNow integration is set up.

How to set up a Salesforce ServiceNow Integration

To set up a Salesforce ServiceNow integration, you need to first install Exalate on both sides.

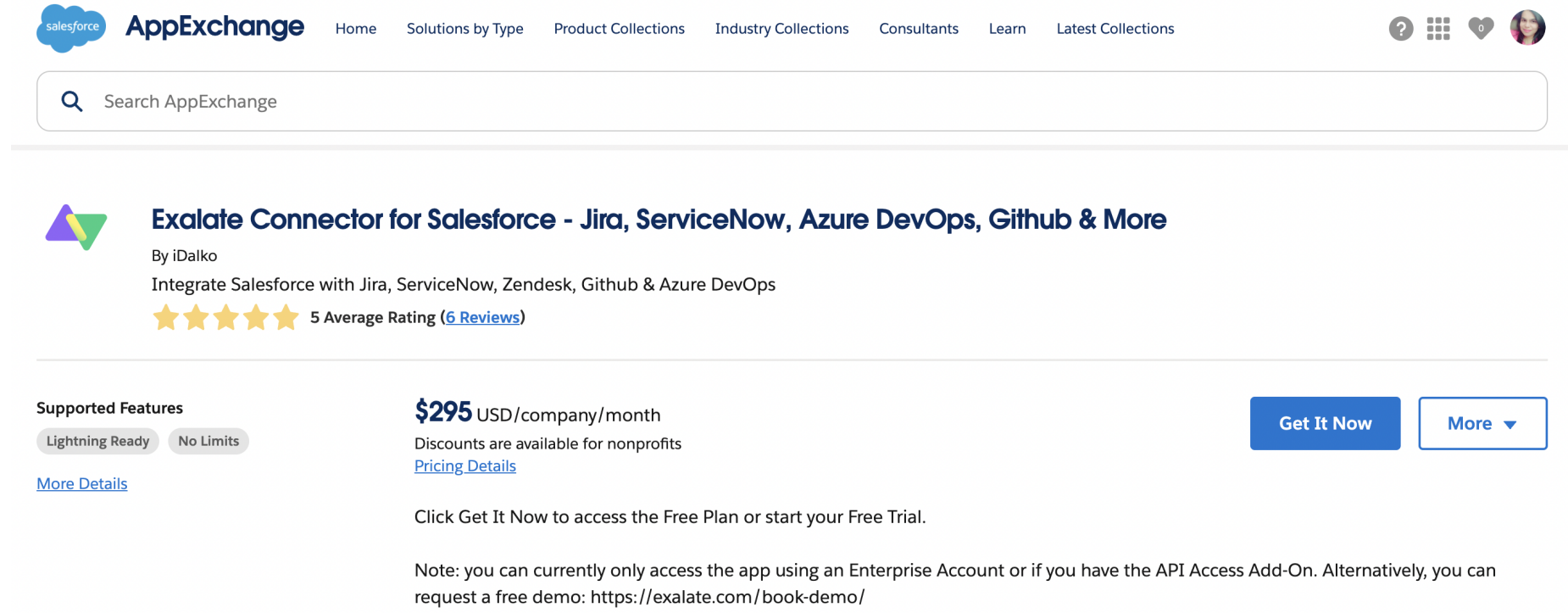
Once done, it can be configured to share the incoming and outgoing information on each side independently of one another. After that, you can start your synchronization.

Step 1: Install Exalate on Salesforce

For installing Exalate on Salesforce search for AppExchange in the global search bar of Salesforce and type AppExchange to open the marketplace. Alternatively, you can visit AppExchange through this [link](#).

Note: Exalate accesses Salesforce through APIs. Salesforce has its own [guidelines](#) for API Access add-ons. For instance, API access is provided by default in Enterprise accounts, while it is not the case with other accounts like Professional. Visit this documentation [page](#) to learn about the different Salesforce editions Exalate supports.

In the "Search AppExchange" search bar, type "Exalate" and select it from the list that appears.



The screenshot shows the Salesforce AppExchange interface. At the top, there is a navigation bar with the Salesforce logo, the text "AppExchange", and several menu items: Home, Solutions by Type, Product Collections, Industry Collections, Consultants, Learn, and Latest Collections. On the right side of the navigation bar, there are icons for help, a grid, a heart, and a user profile. Below the navigation bar is a search bar with the placeholder text "Search AppExchange".

The main content area displays the "Exalate Connector for Salesforce - Jira, ServiceNow, Azure DevOps, Github & More" app listing. The app is by iDalko and is described as "Integrate Salesforce with Jira, ServiceNow, Zendesk, Github & Azure DevOps". It has a 5-star average rating based on 6 reviews.

Under the "Supported Features" section, there are two tags: "Lightning Ready" and "No Limits". The pricing is listed as "\$295 USD/company/month" with a note that discounts are available for nonprofits and a link to "Pricing Details". There are two buttons: "Get It Now" and "More".

Below the pricing, there is a note: "Click Get It Now to access the Free Plan or start your Free Trial." and another note: "Note: you can currently only access the app using an Enterprise Account or if you have the API Access Add-On. Alternatively, you can request a free demo: <https://exalate.com/book-demo/>".

Go through the app details at this point and click the green "Get It Now" button to begin the installation procedure. You will be asked to log in to the AppExchange console if needed. Do that and proceed.

Next, choose the environment in which you need to install the Exalate package. Choose between Production or Sandbox. I am choosing "Install in Production" here.



Some installation details will be displayed. Read them and agree to the terms and conditions at the bottom of the page before clicking "Confirm and Install".

Confirm Installation Details

Free	idalko
Duration	Number of Subscribers
Does Not Expire	Site-wide
Username	
teja.bhutada@idalko.com	

Here are the details we'll share from your profile

[Edit Profile](#)

- * First Name Teja
- * Last Name Bhutada
- * Company Exalate
- * Country India
- Job Title
- State/Province
- * Email teja.bhutada@idalko.com
- Phone

* I have read and agree to the [terms and conditions](#).

Salesforce.com Inc. is not the provider of this application but has conducted a limited security review. [Learn More about the AppExchange Security Review](#)


[Cancel](#) [Confirm and Install](#)



On the next screen, you are supposed to choose the users that will have access to the Exalate app. This is for added security that allows you to control who has access to the app.


I am choosing "Install for Admins Only" since I don't want it to be accessed by others. But you can choose the other 2 options depending on your needs. Don't worry if you can't come to an immediate decision, you can change the permissions anytime! Just click [here](#) to know how to go about that.


For specific profile access, a list of the available users will be displayed and you can choose to set the permissions then. Click "Install" to proceed.




Install ExalateBridgeApp

By Exalate


Install for Admins Only


Install for All Users



Install for Specific Profiles...

App Name	Publisher	Version Name	Version Number
ExalateBridgeApp	Exalate	1.1.0	1.1

Additional Details [View Components](#)


Click "Continue" on the next pop-up asking you for access to third-party websites. So give the necessary permissions to Exalate by clicking "Allow".

A successful installation after this will display the "Installation Complete!" message. Then click the "Done" button to go back to your Salesforce instance.



Install ExalateBridgeApp

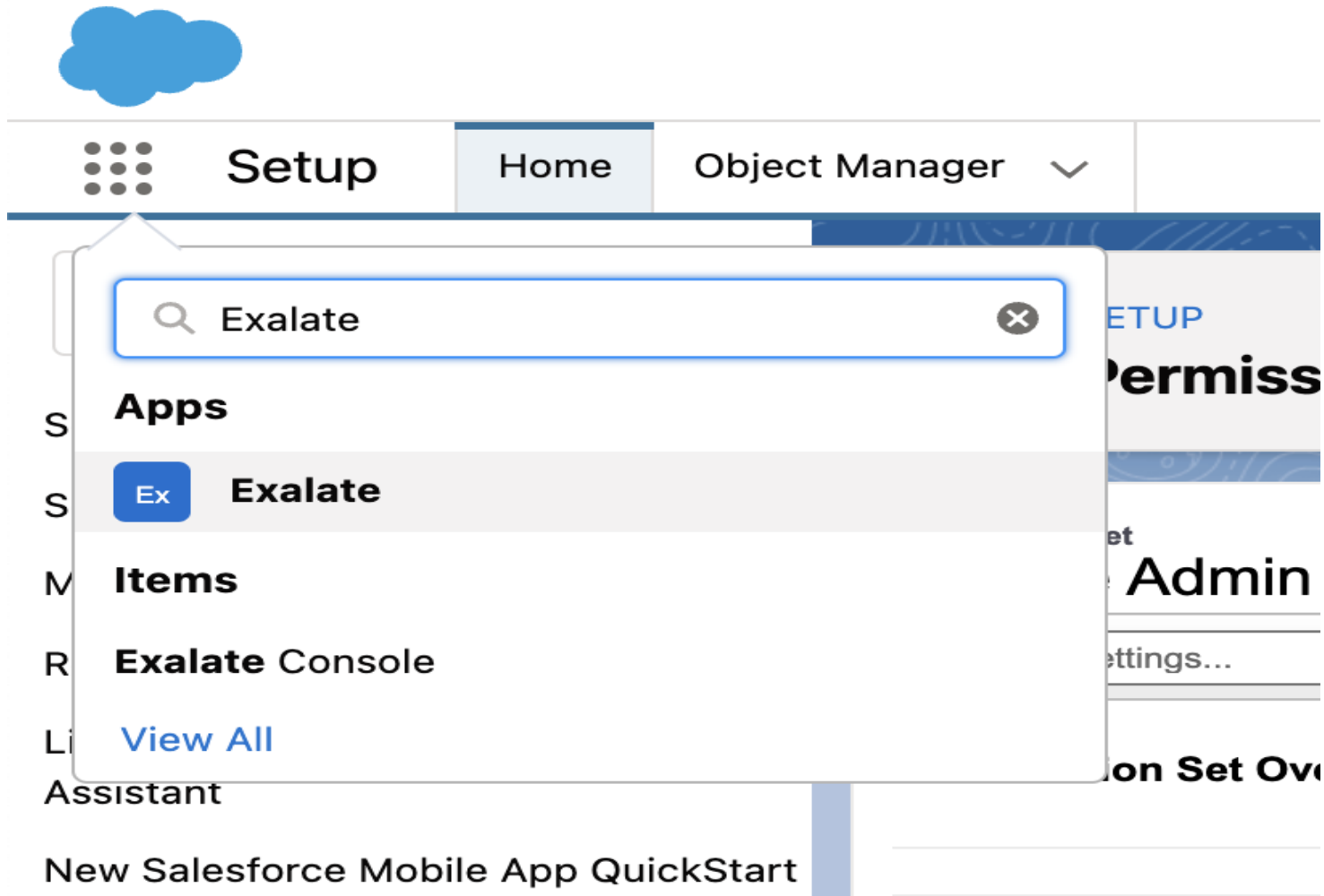
By Exalate

 **Installation Complete!**

[Done](#)

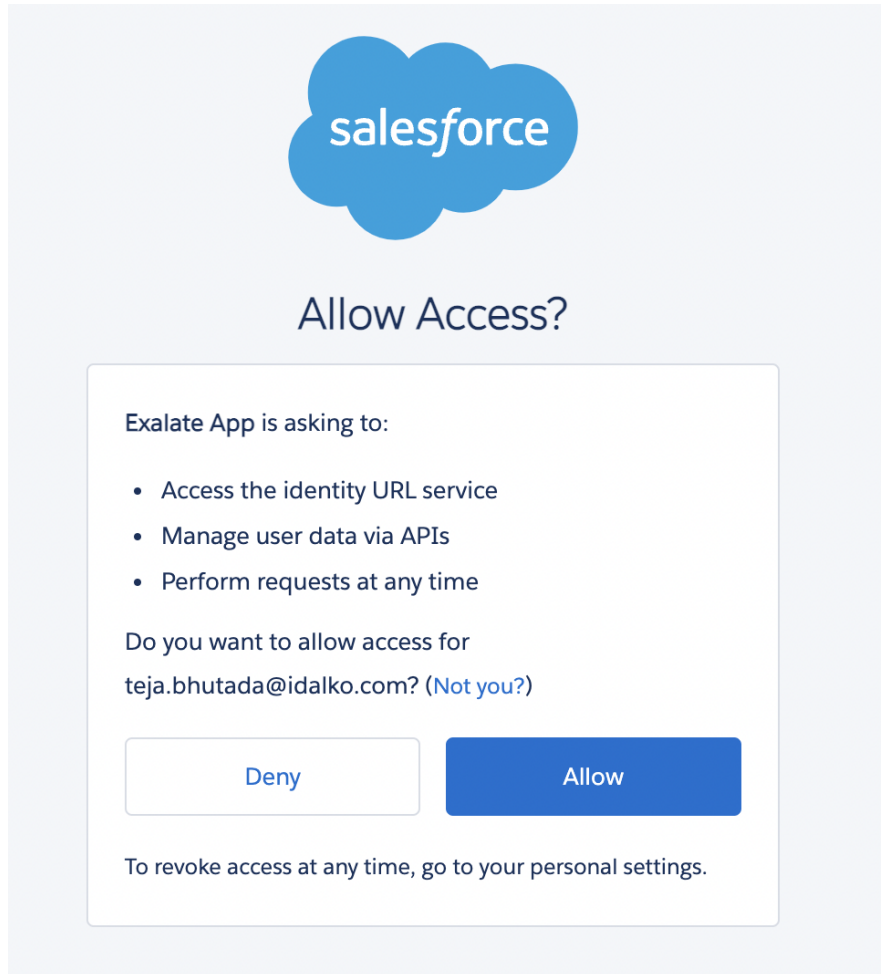
App Name	Publisher	Version Name	Version Number
ExalateBridgeApp	Exalate	1.1.0	1.1

Now is the time you request for the Exalate node. To do that click on the "Apps" section on the left-hand side of the Salesforce environment and search for Exalate.



Then click the "Request node" button to proceed.

Don't forget to "Allow" for necessary permissions.



And then fill in your contact information. This will help Exalate verify your instance and activate your evaluation license.

Fill out the necessary information and click on "Agree and Submit".

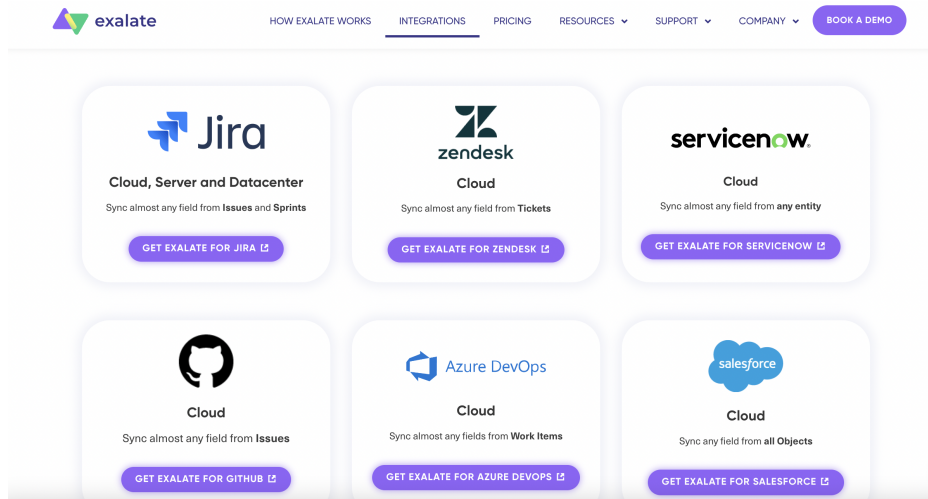
You will now have received an email. Open it and click on "Verify Exalate instance". Then you will be redirected to the Salesforce instance. If required follow these [steps](#) to log in to your Exalate for Salesforce console.

Once done, you can start using Exalate right away! But if you still have to install it on ServiceNow proceed to the next step.

Step 2: Install Exalate on ServiceNow

[Here](#) is a complete guide to installing ServiceNow on Exalate. But we will discuss the process briefly here.

You can request a ServiceNow instance by navigating to Exalate's [integrations page](#).



On this page, click on the ServiceNow logo. A pop-up form will appear. Enter the organization details and a few other personal details and click “Submit”.

You will receive an email with your ServiceNow node URL.

Request a Free Trial of Exalate for ServiceNow

The Exalate for ServiceNow instance will allow you to setup a fully functional integration between your ServiceNow environment and any other supported tracking platform.

Enter Organisation*

Enter First name

Enter Last name*

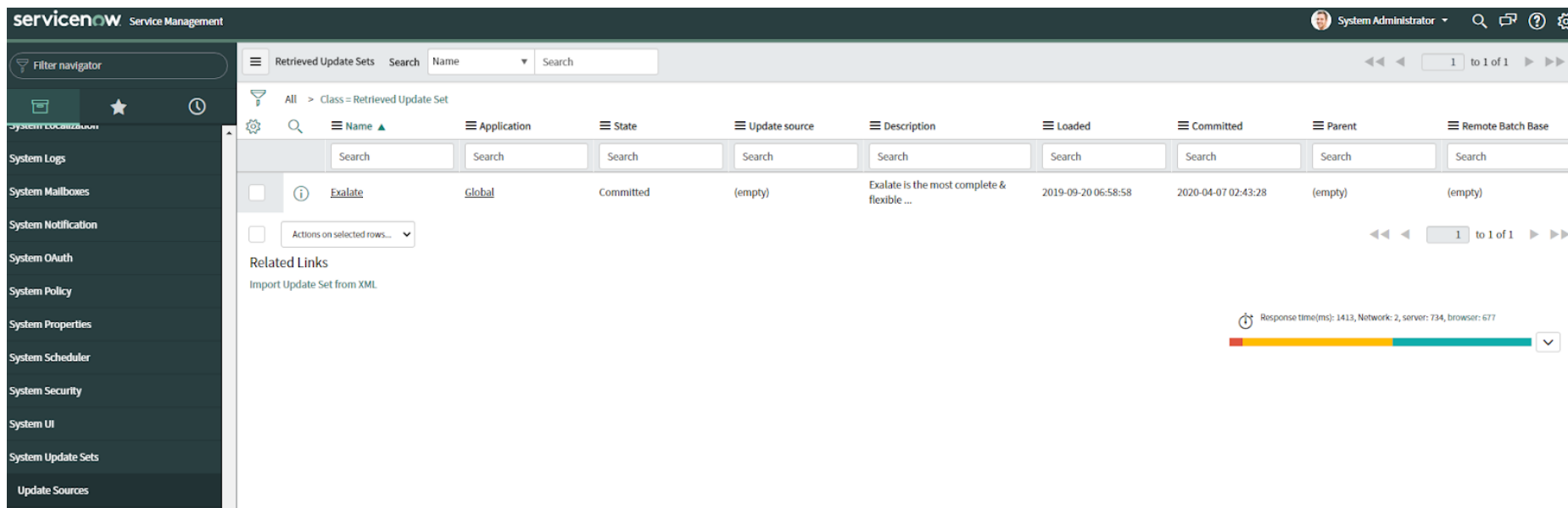
Enter Phone number*

Enter Email address*

SUBMIT

For the next part, you will need to download an XML file. This file consists of information ServiceNow will need to access Exalate. Get it [here](#).

Next, log in to your ServiceNow account. In the left-hand menu search for “System Update Sets”. Click on the “Filter navigator” search box, if the list is too long and you cannot find it. Now expand the “System Update Sets” by clicking on it. In the entry select “Retrieve Update Sets”.



The screenshot shows the ServiceNow interface for 'Retrieved Update Sets'. The left-hand menu is expanded to 'System Update Sets'. The main content area displays a table with the following columns: Name, Application, State, Update source, Description, Loaded, Committed, Parent, and Remote Batch Base. The table contains one row for 'Exalate' with the following values: Name: Exalate, Application: Global, State: Committed, Update source: (empty), Description: Exalate is the most complete & flexible..., Loaded: 2019-09-20 06:58:58, Committed: 2020-04-07 02:43:28, Parent: (empty), Remote Batch Base: (empty). Below the table is a 'Related Links' section with a link to 'Import Update Set from XML'. A response time bar is visible at the bottom right of the interface.

Under the “Related Links” heading, click on “Import Update Set from XML”.



Choose the XML file that you have downloaded and click “Upload”. After a successful update, the XML file will be listed. Now click on the “Preview Update Set”.

Click on “Commit Update Set” to finish the installation of Exalate on ServiceNow.

You can now proceed to the next step.

Step 3: Set up a Connection between Salesforce and ServiceNow

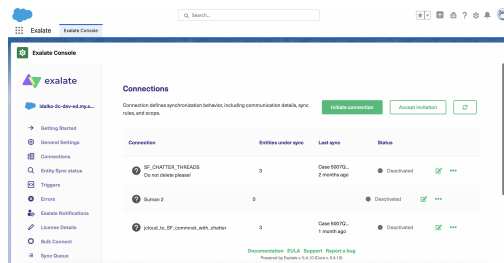
Before you can start synchronization with Exalate, you need to establish a connection between Salesforce and ServiceNow. The connection defines how your synchronization behaves and it specifies your sync rules and scope.

One side initiates the connection and the other side accepts it. This can be done from either the Salesforce or the ServiceNow side. The Exalate UI is uniform across different platforms, so you can perform the remaining steps on either side.

Let us start initiating the connection from the Salesforce instance.

To do this, in the Exalate left-side menu click “Connections”. This screen has a list of all your connections. If it’s your first time, it will be empty.

Start by clicking “Initiate connection”.



A pop-up will appear. Enter the destination URL in the text box. The destination means the other side you want to connect with. In our case, that's ServiceNow.

In case you don't know the URL, you can navigate to the Exalate console on ServiceNow and click on "General Settings" in the left-hand menu. Copy the URL from there.


After entering the URL, Exalate performs a quick check to see if it's been installed on the other side. You will get appropriate status messages.

Initiate connection ×


Destination instance URL ⓘ

✓ I don't have a URL

Choose the configuration type

 **Basic**

- Automatic configuration of basic fields
- Sync rules cannot be edited
- Only Cases can be synced
- Recommended for use cases of basic complexity

 **Script**

- Groovy-based scripting
- Configure each side of the connection separately
- Recommended for use cases of basic to advanced complexity

Next

You are then prompted to choose the mode of connection.

Exalate supports 2 modes: **Basic** and **Script**. The UI for both these modes is slightly different, so we will take a look at both of them in detail.

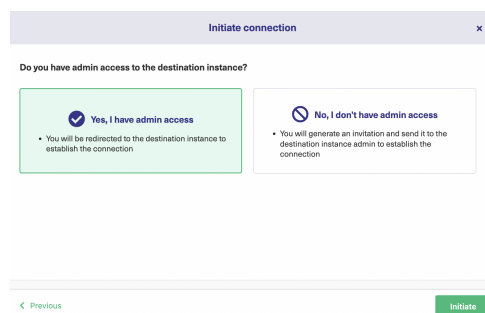
Continue with the Basic Mode

The Basic mode is a low-code configuration UI. It comes with pre-defined sync rules and mappings that you cannot change. It's suitable for your basic synchronization needs. But if you need some advanced customizations, you can [upgrade](#) to the script mode anytime!

Note: Exalate also comes with a Forever [Free plan](#). With this plan, you can have Basic mode connections with up to 1000 free syncs per month. Get started with the Free plan [here](#).

To continue with this mode, click “Yes, I have admin access” on the next screen if you have access to the ServiceNow instance. Click no if you don't and Exalate will generate an invitation code for you, that you can copy and paste on the ServiceNow side. You can refer to the details on how to go about this in the Script mode section.

Either way, click “Initiate”.



Initiate connection x

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

After a quick verification, you will be redirected to the ServiceNow side.

Here, you can start your first synchronization by entering the incident number and clicking “Exalate”.

Exalate ×

Connection established successfully ✓

Sync your first incident to see how it works.

Please enter an incident number to proceed

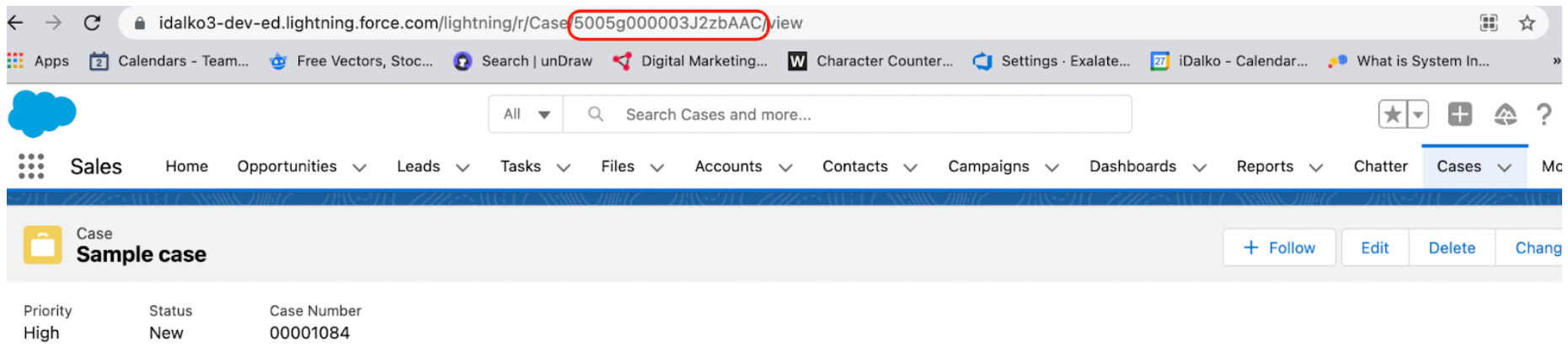
Exalate



You can also choose to synchronize later by closing this window and creating [triggers](#) or using the “[Bulk Connect](#)” option.

Note: At this point, you can even go to the Exalate console on the Salesforce side. You will see a similar screen being displayed there, asking you to enter the Case key to sync. If you don't see this screen, you can click on the 3 dots next to your connection name in the “Connections” tab and click “Exalate”.

For fetching the URL of Salesforce entities you must click on that entity and copy the number from the URL, as shown in the image below.



The screenshot shows a Salesforce interface. The browser address bar contains the URL: `idalko3-dev-ed.lightning.force.com/lightning/r/Case/5005g000003J2zbAAC/view`. The case ID `5005g000003J2zbAAC` is highlighted with a red box. The page header includes the Salesforce logo and navigation tabs for Sales, Home, Opportunities, Leads, Tasks, Files, Accounts, Contacts, Campaigns, Dashboards, Reports, Chatter, Cases, and My Cases. The main content area shows a case titled "Sample case" with a status of "New" and a priority of "High". The case number is "00001084". Action buttons for "+ Follow", "Edit", "Delete", and "Change" are visible.

Wait for some time.

A successful synchronization looks like this. You can click on the remote link to see the synced entity on the Salesforce side, or you can click on the incident number link to go to the incident on the ServiceNow side.



To view the default mappings of the Basic mode connection, click on the edit connection icon next to the connection name under “Connections” in the Exalate console.

	Service-Now_to_JiraCloud	11	incident INC00... 4 days ago	● Active	
	idalko-2c-dev-ed_to_ven03945	1	incident INC00... 12 seconds ago	● Active	

Edit connection

As seen, you cannot change these default mappings, but choose to upgrade for advanced integration use cases.

idalko-2c-dev-ed_to_ven03945 Active [Back to Connections](#) [Publish](#) [Upgrade](#)

Scope **Rules** Statistics Info

Configure the synchronization behavior with the help of field mappings and script rules.

[Expand all](#) [Collapse all](#) [+ Add mapping](#)

Order	ven03945	Sync direction	idalko-2c-dev-ed	
1	Type	↔	Type	
2	Short description	↔	Short description	
3	Description	↔	Description	
4	Additional comments	↔	Additional comments	
5	Attachments	↔	Attachments	

Continue with the Script Mode

If you have complex and advanced use cases, we recommend you use the Script mode. It gives you the ability to modify the sync rules and control what and how information is shared with the other side.

To continue using the Script mode, you must generate an invitation code from one side and accept the code on the other side.

We have already initiated the connection from the Salesforce side.


Select “Script” on the screen below and click “Next”.

Initiate connection ×


Destination instance URL ⓘ

✓ I don't have a URL

Choose the configuration type

 **Basic**

- Automatic configuration of basic fields
- Sync rules cannot be edited
- Only Cases can be synced
- Recommended for use cases of basic complexity

 **Script**

- Groovy-based scripting
- Configure each side of the connection separately
- Recommended for use cases of basic to advanced complexity

Next

Name the connection now.

For this, you must give a name to your local instance (Salesforce in our case) and also to the remote instance (ServiceNow). Exalate automatically generates a connection name for you by combining the local and remote instance names. But you can change it if you want.

Also, give it a description, because it is extremely useful when you have a lot of connections and it becomes difficult to know why you created them in the first place.

Whenever you are ready, click “Initiate”.

Initiate connection ×

Connection information

Local instance short name*	Remote instance short name*
<input type="text" value="Salesforce"/>	<input type="text" value="ServiceNow"/>

Connection name*

Description

[< Previous](#) [Initiate](#)

An invitation code is generated. This is a unique string that Exalate generates for every connection you set up. It is a part of its security mechanism.

Copy the code somewhere safe and click “Done”.

Initiate connection ×

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

Use the following invitation code:

[Copy invitation code](#)










Done

If you do this, the connection on the Salesforce side will be shown as “Pending” and only when you finish setting it up on the ServiceNow instance, will the status become “Active”.

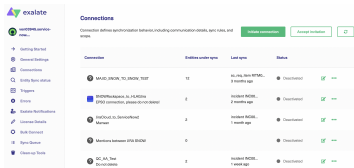
Connections

Connection defines synchronization behavior, including communication details, sync rules, and scope.

Initiate connection
Accept invitation
↻

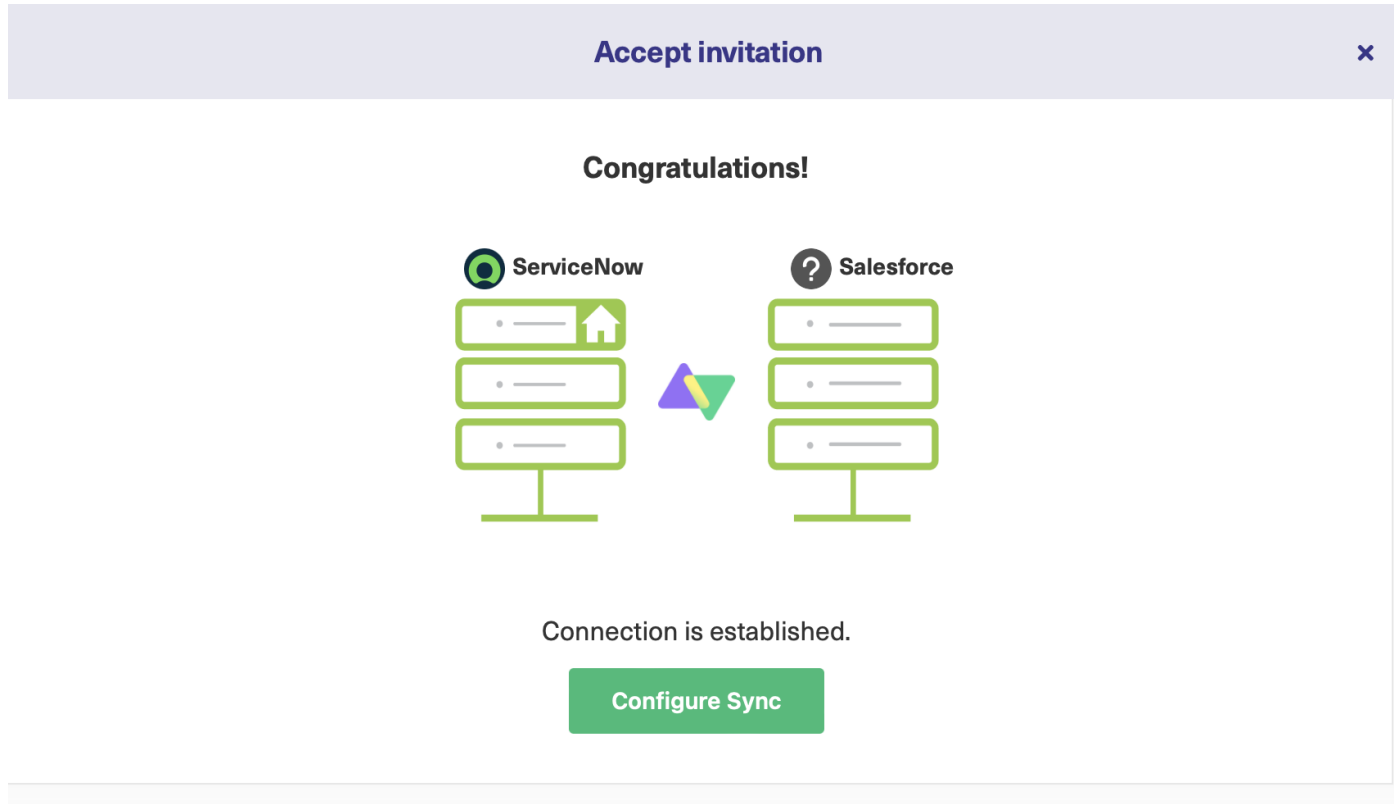
Connection	Entities under sync	Last sync	Status
 SF_CHATTER_THREADS Do not delete please!	3	Case 5007Q... 2 months ago	 Deactivated  
 Salesforce_to_ServiceNow This is a connection between Salesforce and Ser...	0		 Pending   

On the ServiceNow side, under “Connections”, click the “Accept Invitation” button.











After this, your connection has been successfully established and you can see the status as follows. You can now configure it to control what information is sent to the other side.

For this click on the “Configure Sync” button and proceed to the next step.



Step 4: Configure your connection to determine what information gets shared

If you don't wish to configure the connection right away, you can do it later by clicking on the edit connection icon next to your connection name under the "Connections" tab.

Connection	Entities under sync	Last sync	Status		
 SF_CHATTER_THREADS Do not delete please!	3	Case 5007Q... 2 months ago	 Deactivated		
 Salesforce_to_ServiceNow This is a connection between Salesforce and Ser...	0		 Active		

From here, you can choose to activate, deactivate, or delete the connection by clicking on the 3 dots. You can even go to the other side of the connection by clicking the remote antenna button. The status of your connection is also displayed here apart from some other general information.

Let's proceed to the steps after you press the "Configure Sync" button. Editing the connection also has similar screens, so either way, there are 4 tabs displayed: "Rules", "Triggers", "Statistics" and "Info".

The "Statistics" tab gives an overview of synchronization-related information like the number of issues, comments, and attachments synced. It also states the number of issues last synced, in addition to the date and time of the sync.

The “Info” tab gives general information related to the connection like its name, type, description, and destination URL. You can edit the description here if you want.

» **Salesforce_to_ServiceNow**
● Active

< [Back to Connections](#)

[Publish](#)

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
21   replica.description   = entity.description
22   replica.name          = entity.name
23 }
24
```

[Copy outgoing sync processor to clipboard](#)

▼ **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") {
```

We will discuss the “Rules” tab now and “Triggers” in step 6.

Exalate has an “Outgoing sync” and “Incoming sync” written in Groovy scripting language. Both are rules that control what information is sent and received from the other side.

The outgoing sync rules as their name suggests control what information must be sent to the destination side, while the incoming rules control how to interpret the information coming from the other side.

Both these rules are present on either side of the connection i.e Salesforce and ServiceNow. So you have complete autonomy in deciding what must be shared and what shouldn't. You don't need to configure or even inform the other side about the way you modify your script rules. So if you are in Salesforce, "Outgoing sync" specifies what information you send to ServiceNow and the "Incoming sync" specifies how you receive information from ServiceNow. This holds true for ServiceNow as well.

The rules on the Salesforce side look like this:

The screenshot shows the configuration page for a connection named "Salesforce_to_ServiceNow". At the top, there is a status indicator "Active" and two buttons: "Back to Connections" and "Publish". Below this, there are four tabs: "Rules", "Triggers", "Statistics", and "Info". The "Rules" tab is selected and shows two sections:

- Outgoing sync**: A Groovy script defining a processor for outgoing sync. The script maps fields from the source entity to the destination replica.

```
1 if(entity.entityType == "Case"){
2   replica.key = entity.Id
3   replica.summary = entity.Subject
4   replica.description = entity.Description
5   replica.comments = entity.comments
6   replica.attachments = entity.attachments
7   replica.Status = entity.Status
8 }
```
- Incoming sync**: A Groovy script defining a processor for incoming sync. The script checks if it's the first sync and then filters for "Case" entities.

```
1 if(firstSync){
2   entity.entityType = "Case"
3 }
4 if(entity.entityType == "Case"){
```

Below the "Outgoing sync" script, there is a link: "Copy outgoing sync processor to clipboard".

You can edit the rules to control the information flow.

For this, you can either delete a particular line to stop sharing information or add additional lines for syncing something extra like custom fields.

For deletion, remove the particular line from the incoming or outgoing sync or simply add a comment.

Add comments by adding “//” at the start of the line. If you need to comment a block of lines add “/*” at the starting of the block and “*/” where the comment ends.

For example, if we add a comment to the line `replica.description = entity.Description` in the Outgoing sync section, then the description of the Case will not be shared with the ServiceNow instance.

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

Rules Triggers Statistics Info

▼ **Outgoing sync** ⓘ

```
1 - if(entity.entityType == "Case") {
2   replica.key = entity.Id
3   replica.summary = entity.Subject
4   //replica.description = entity.Description
5   replica.comments = entity.comments
6   replica.attachments = entity.attachments
7   replica.Status = entity.Status
8 }
```

[Copy outgoing sync processor to clipboard](#)

▼ **Incoming sync** ⓘ

```
1 - if(firstSync){
2   |   entity.entityType = "Case"
3   }
4 - if(entity.entityType == "Case"){
```


Any Salesforce entity can be synced by adding additional scripts relevant to that entity. Refer to this [document](#) on how to do it. For the ServiceNow side, check out this [one](#) instead.

For example, if you want to sync Tasks in addition to Cases in Salesforce, then you add the following script in the Outgoing sync.

```
if(entity.entityType == "Task") {  
  
    replica.key      = entity.Id  
  
    replica.summary  = entity.Subject  
  
    replica.description = entity.Description  
  
}
```

Accordingly, the subject and description of the Task will be shared on the ServiceNow side.

Once done, click “Publish”. The changes you have made will be saved and applied from the next sync onwards.

To add “Triggers”, don’t click “Publish” just yet and proceed to the next step.

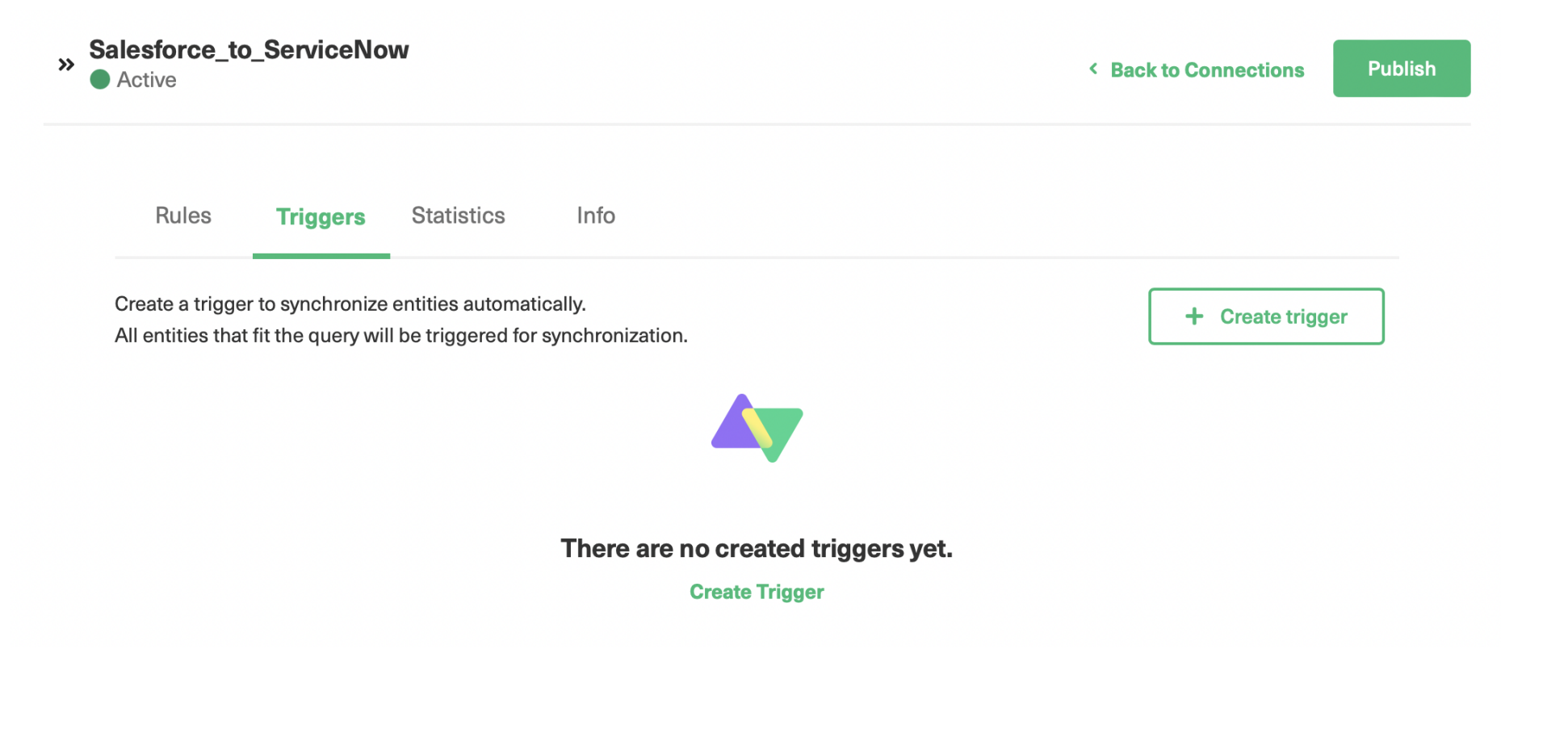
Step 5: Create Automatic Synchronization Triggers

Triggers are used to synchronize Salesforce and ServiceNow entities automatically.

For creating triggers you need to add a query. When Exalate comes across entities that match the search query, they are automatically synchronized with the other side according to the sync rules you have set.

To begin with, click on the “Triggers” tab on the above screen. You can also reach triggers by clicking on the left-hand “Triggers” menu of the Exalate console. If you do it this way, there is an extra step of selecting the connection you want to apply the trigger for. All the other steps are the same.

On this screen, you will see all your triggers. To create a new one, click “Create trigger”.



An “Add trigger” pop-up will appear.

You start by selecting the entity from a dropdown list to which the trigger is applied.

There are many ServiceNow entities like Change Requests, Cases, Problems, etc. supported by Exalate.

We have selected “Incident” for this example.

Add trigger ✕

Specify a ServiceNow 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* ⓘ

incident | ▾

If* ⓘ

urgency=1

Notes

This is the trigger for syncing incidents that have the urgency = 1

Active?

Add

The “If” section allows you to enter search queries specific to the platform. Since we are on the ServiceNow side, we have given the query as “urgency=1”. So a trigger to sync Incidents with the highest priority is created. When such an incident is generated, it will be automatically synced.

Note: Refer to this [page](#) if you want to know more about creating triggers.

The same on the Salesforce side will allow you to select Salesforce entities from a drop-down list. There are many Salesforce entities supported by Exalate, but the most common ones are **Account, Opportunity, Case, Task,** and **Product**. Here, we have selected an “Opportunity”.

And then either select conditions to filter the Opportunity by entering the details in the text boxes or toggle the switch to use the search query instead. For the search query on the Salesforce side, the [Salesforce Object Query Language](#) (SOQL) is used.

Add trigger ×

Specify a search query using Salesforce advanced search syntax 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* !

Opportunity ▾

Use search query ×

Select conditions to filter Opportunity for synchronization:

Name	Description	Quantity
Name +	Description +	Quantity +

Next Step

Next Step + More ▾

Learn more about how to add search queries in Salesforce [here](#).

There is an option to add “Notes” to the trigger. Add them to describe what the trigger does.

You can also use the “Active” toggle switch to activate or deactivate the trigger. This is useful when you don’t want to use the trigger right now, but also don’t want to go through the effort of creating it from scratch again. Remember to toggle it to the "Active" stage to make the trigger work.

Click “Add” when you are done.

» **Salesforce_to_ServiceNow** ● Active < Back to Connections Publish

Rules **Triggers** Statistics Info

Create a trigger to synchronize entities automatically.
All entities that fit the query will be triggered for synchronization. + Create trigger

When	If	Status	Action
Incident Events: create/update	urgency=1	<input checked="" type="checkbox"/>	

< 1 >

You can see the trigger listed on the previous screen as shown above. There is a button to edit or delete the trigger. You also synchronize the existing entities that fulfill the trigger condition using the “Bulk exalate” option by clicking the 3 dots next to the trigger.

After you are done making the changes, click “Publish”.

Step 6: Start Synchronizing the Information

For the **Script mode**:

Adding or deleting scripts in [sync rules](#) means you control what is sent and received, but to start the synchronization, you either need to create [triggers](#) or sync existing Salesforce and ServiceNow entities in bulk using the [“Bulk Connect”](#) option.

For the **Basic mode**:

You can simply sync entities by entering the Incident number or the Case number after creating a connection as we saw in step number 3.

Or by clicking on the 3 dots next to the connection name in the “Connections” tab either on the Salesforce or the ServiceNow side and then selecting “Exalate”.

You can create a trigger or sync entities in bulk too. But you cannot add new ones or change the existing default mappings for advanced configurations like in the Script mode.

Common Use Cases

Customer Support and Sales Teams

Perhaps the most common scenario that comes to mind for a Salesforce ServiceNow integration is for your sales team to have an automatic update on the “Incident” status of key customers.

So when an Incident of urgency =1 from a particular customer comes up in ServiceNow, it will automatically create a Case or a Task (depending on the trigger you set) on the Salesforce side, so that the sales team can keep track of it automatically thereon.

Any updates or additional information requested by the customer for an Incident can be viewed in Salesforce and provided to the support team by the sales team if they have it with them. This can resolve Incidents faster and improve SLAs, leaving you with happier customers.

The customer support agents can also review Cases, Accounts, or Opportunities in Salesforce to gather information relevant to their Incidents, helping them to resolve them faster.

Customer Success Representatives and Sales Teams

You can also envision the customer success representatives having a complete overview of the customers in their ServiceNow by gathering information from Accounts, Opportunities, and Cases in Salesforce.

They can then work on maintaining a low churn and help generate better revenues for the business. Sales teams can also proactively provide feedback, queries, or other relevant details to the customer success teams to help them with their responsibilities better.

Development and Sales teams

It's possible that your development team practices agile development methodology and uses ServiceNow. Your sales team, on the other hand, uses Salesforce. Since the sales team is constantly in contact with customers and is well aware of their feedback, complaints, queries, or feature requests, it would be helpful if they get to exchange this information with the development team.

This can in turn help the developers get an idea about what customers or end-users are looking for in a product or how they want to improve it. An integration between ServiceNow and Salesforce can help these teams share information and filter out unnecessary details.

Conclusion

A Salesforce ServiceNow integration can definitely help teams close their information gaps and bring them to work towards a common goal of improving customer experience.

For this, we saw how such an integration will be beneficial for teams working on different platforms like ServiceNow and Salesforce, without them having to leave their own familiar environment.

We then got a glimpse of what the right integration tool must inherently support and chose Exalate as the tool for our integration because it supports the features we were looking for.

And then we finally saw how such an integration can be implemented through a step-by-step approach. We also saw common use cases that can benefit from such an integration.

Recommended Reads:

- [How to set up a Jira Salesforce Integration](#)
- [ServiceNow to ServiceNow Integration: The Step-by-Step Guide to Setting up a Two-Way Sync](#)
- [Salesforce to Salesforce Integration: Sync Multiple Salesforce Instances Bidirectionally](#)

- [Jira ServiceNow Integration: How to Set up an Integration in 6 Steps](#)
- [Salesforce Zendesk Integration](#)
- [Zendesk ServiceNow Integration](#)
- [How to Set up a ServiceNow GitHub Integration](#)
- [How to Set Up an Azure DevOps ServiceNow Integration](#)
- [ServiceNow Integrations: Integrate ServiceNow and Other Systems Bidirectionally](#)