How to Sync Any Entity in Exalate for ServiceNow

Last Modified on 04/08/2024 8:35 am EDT

Starting from version 5.0.28, Exalate can sync any entity from ServiceNow. This article describes examples of how to sync various fields.

Scripts below use the field and table names from the ServiceNow database. You can find out the field and table names of any entity in ServiceNow. For more information, please read How to find out the name of a field in ServiceNow.

In this section

ServiceNow

Outgoing sync

On the outgoing sync, you define which fields you want to send with each entity. Use the variable `entity.tableName` to know the entity you are executing the script for. With this script, you can sync fields from the **Incident** and **Business Application** entities.

The **Incident** entity is located in the `incident` table in the ServiceNow database. This script shows how to sync the following fields from the `incident` table:

- `key`
- `short_description`
- `description`
- `attachments`
- `comments`
- `state`

The **Business Application** entity is located in the `cmdb_ci_business_app` table in the ServiceNow database. This script shows how to sync the following fields from the `cmdb_ci_business_app` table:

- `key`
- `short_description`
- `description`
- `name`

```
if(entity.tableName == "incident") {
    replica.key            = entity.key
    replica.summary        = entity.short_description
    replica.description    = entity.description
    replica.attachments    = entity.attachments
    replica.comments       = entity.comments
    replica.state          = entity.state
    /*
    Use a field's internal name to send its value
    Example: Resolution Notes -> resolution_notes
    This works for all other entity types as well
    replica.resolution_notes = entity.resolution_notes
    */
}
//any other entity can be synced using the table name and the entity variable
if(entity.tableName == "cmdb_ci_business_app") {
    replica.key            = entity.key
    replica.summary        = entity.short_description
    replica.description    = entity.description
    replica.name           = entity.name
}
```

Script variables

`entity.tableName`

A variable used to know what table to sync. Assign the name of the ServiceNow table name to this variable. This example shows how to sync data from two tables: `incident` and `cmdb_ci_business_app`. You can sync any entity within the `if` condition.

For example, with the `replica.key = entity.key` line, you sync the `key` field from the `incident` table. You can replace `key` with other fields, like `description` or `short_description.`

Incoming sync

This script shows how to set up incoming sync for ServiceNow entities. This example script shows how to sync issues between ServiceNow and Jira.

On the first synchronization, it's important to define what ServiceNow entities you want to create after receiving data from Jira. This is done with the `if(firstSync)` variable. In this example, **Incident** is used as the default entity, where all data is synced. The fields from the `cmdb_ci_business_app` table are synced in the **Business Application** entity.

```
if(firstSync){
 //Decide on the first sync, which entity you want to create based on the remote issue type
 if(replica.typeName == "Business Application"){
    entity.tableName = "cmdb_ci_business_app"
 }else{
    entity.tableName = "incident"
 }
}
if(entity.tableName == "incident") {
    entity.short_description = replica.summary
    entity.description = replica.description
    entity.attachments += replica.addedAttachments
    entity.comments += replica.addedComments
    /*
    Jira Custom Field to ServiceNow Field
    Apply the value from a Jira custom field to the Resolution Notes
    This works for all other entity types as well
    entity.resolution_notes = replica.customFields."Jira CF Name".value
    */
    /*
    Status Synchronization
    Sync status according to the mapping [remote incident status: local incident status]
    If statuses are the same on both sides don't include them in the mapping
    def statusMapping = ["Open":"New", "To Do":"Open"]
    def remoteStatusName = replica.status.name
    entity.state = statusMapping[remoteStatusName] ?: remoteStatusName
   */
}
//any other entity can be synced using the table name and the entity variable
if(entity.tableName == "cmdb_ci_business_app") {
    entity.short_description = replica.summary
    entity.description = replica.description
}
```

Script variables

`if(firstSync)`

A condition where that defines the tables used to store incoming data in ServiceNow. With this script, you can store data in the `incident` and `cmdb_ci_business_app` tables.

`entity.tableName`

A variable is used to know what table you can sync. Assign the name of the ServiceNow table name to this variable. In this example, you can sync data into two tables: `incident` and `cmdb_ci_business_app`. You can sync any entity within the `if` condition.

For example, with the `entity.short_description = replica.summary` line, you sync the `summary` field from Jira issues into the `short_description` field from `incident` table. You can replace `short_description` with other fields, like `description`.

`def statusMapping`

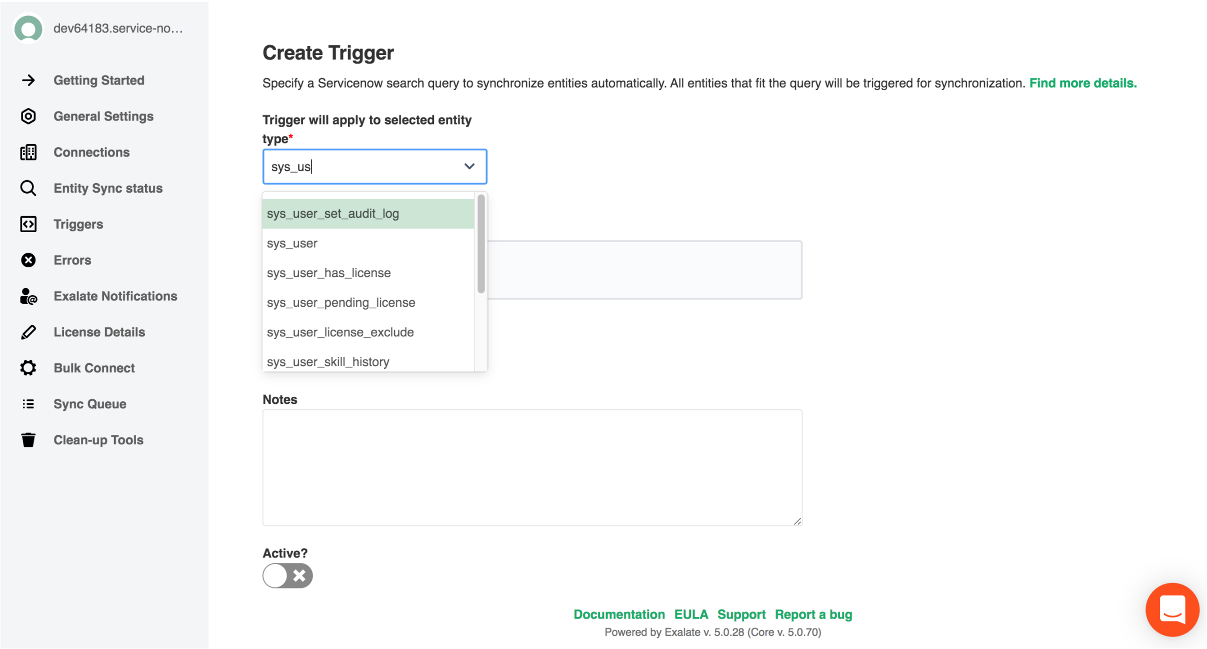
Sets mapping for statuses according to the following template `[remote incident status: local incident status]`. If you set the mapping from the example in ServiceNow, the `New` status in ServiceNow is displayed as `Open` in Jira.

Creating a trigger

If you want an entity to be synced automatically when it matches a ServiceNow query, you need to create a trigger for it in the Triggers tab.

To create a trigger for any entity (table) on your ServiceNow ticket, just start typing the table name on the entity select field:

> **Note**: If there is only one entity type available, the select dropdown menu is disabled.



Create 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*: sys_us

- sys_user_set_audit_log
- sys_user
- sys_user_has_license
- sys_user_pending_license
- sys_user_license_exclude
- sys_user_skill_history

Notes

Active?

Documentation EULA Support Report a bug
Powered by Exalate v. 5.0.28 (Core v. 5.0.70)

> Have more questions? Ask the community

ON THIS PAGE
In this section
ServiceNow
Creating a trigger