BigFix Runbook AI

Integration Guide

Version 6.3

HCL has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on
the HCL website at www.hcltechsw.com.

Table of Contents

Table of Figures

List of Tables

Document Revision History

This guide updates with each release of the product or when necessary.

This table provides the update history of this Integration Guide.

Version Date	Description
June, 2023	BigFix Runbook AI v6.3 Integration Guide

1 Preface

This section provides information about the BigFix Runbook AI Integration Guide and includes the following topics.

- Intended Audience
- About This Guide
- Related Documents
- Conventions

1.1 Intended Audience

This information is intended for administrators authorized for configuring BigFix Runbook AI and enable integrations with various ITSM tools and Runbook Automation / Orchestrator Tools.

1.2 About this Guide

This guide provides instructions to enable integrations with various ITSM and Runbook Automation tools, while configuring BigFix Runbook AI.

1.3 Related Documents

The following documents can be referenced in addition to this guide for further information on the BigFix Runbook AI platform.

- BigFix Runbook AI Configuration Guide
- BigFix Runbook AI Troubleshooting Guide
- BigFix Runbook AI Lab Manual

1.4 Conventions

The following typographic conventions are used in this document:

Table 1 - Conventions

Convention	Element
Boldface	Indicates graphical user interface elements associated with an action, or terms defined in text or the glossary
Underlined Blue face	Indicates cross-reference and links
italic	Indicates document titles, occasional emphasis, or glossary terms
`Courier New` (Font)	Indicates commands within a paragraph, URLs, code in examples, and paths including onscreen text and text input from users
Numbered lists	Indicates steps in a procedure to be followed in a sequence
Bulleted lists	Indicates a list of items that is not necessarily meant to be followed in a sequence

2 BigFix Runbook AI Overview

BigFix Runbook AI is an Intelligent Runbook Automation product which is equipped with Artificial Intelligence, Machine Learning and Natural Language Processing capabilities for simplifying and automating the IT Operations issues resolution lifecycle including incidents, service request tasks, change request tasks and events. It leverages its NLP capabilities for analyzing and understanding the context of a specific issue, recommends the most relevant solution and even triggers the execution, thereby enabling Zero Touch Automated Remediation. It also provides AI-driven Knowledge Recommendation by suggesting relevant knowledge articles from various repositories, both internal and external, as and when required by human agents.

When no runbook is available for automated remediation, it searches & downloads relevant executable codes and scripts for subject matter expert to validate, customize, approve and publish for future use.

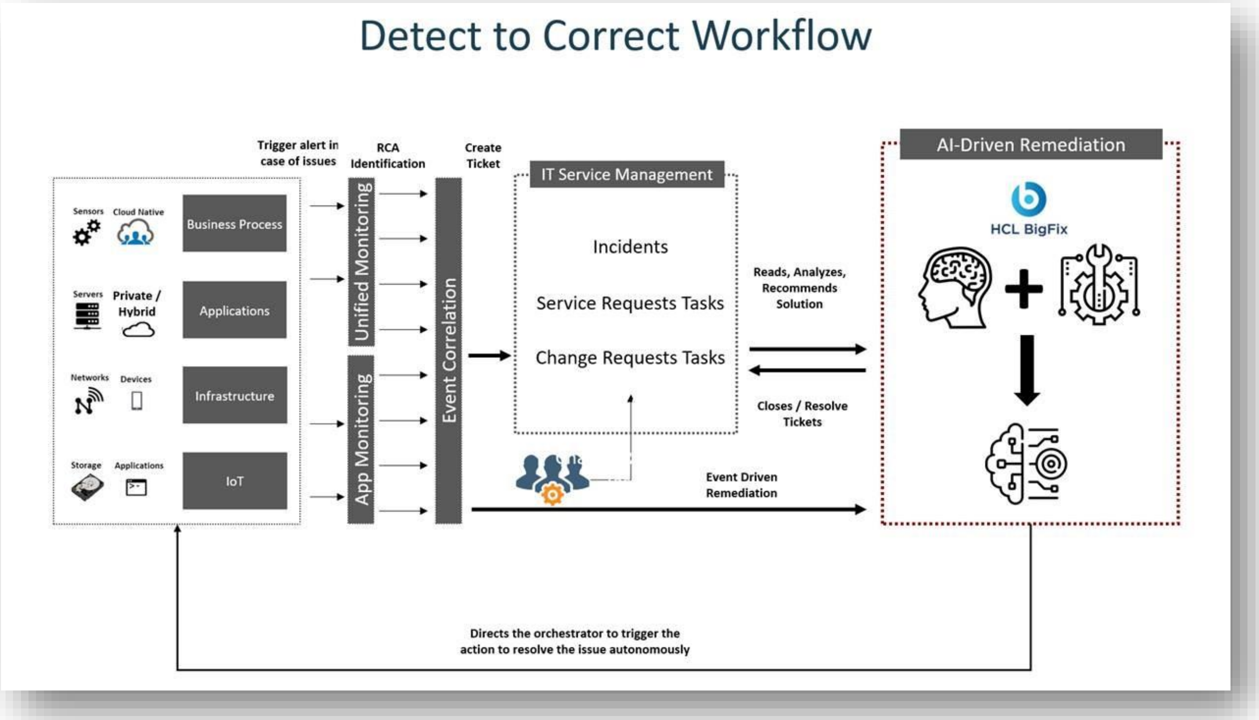


Figure 1 - BigFix Runbook AI Workflow

Intelligent automation powered by BigFix Runbook AI can make a tremendous impact in an enterprise adjusting to the New Normal:

– **Reduce Costs**
  - Achieve up to 30% reduction in service desk related costs
  - Quick and High ROI

– **Mitigate Risks**
  - Avoid operational risks and ensure compliance by avoiding critical outages
  - Reduce escalations and improve SLA compliance by up to 20%
  - Achieve up to 85% reduction in MTTTR

– **Drive Efficiency**
  - Automate redundant tasks and let employees focus on more creative activities
  - Reduce manual effort by 30% to 60%
  - Improve customer satisfaction by up to 50% by providing faster incident and service request resolutions.

– **Rapid Time to Value**
  - Quick implementation in 6 to 8 weeks*

- Leverage 300+ reusable and configurable runbooks out of the box
- Achieve zero-touch automation state in 4 to 5 months*

*Conditions Apply

3 Integration Ecosystem

This section describes the different types of tools with which BigFix Runbook AI can integrate for achieving end to end issue resolution.

Primarily, BigFix Runbook AI integrates with three types of tools –

- **ITSM Tools** – The purpose is to fetch the ticket data from the IT Service Management tool to read / understand the ticket and for making any changes to the ticket like updating the status, work notes, transferring to a different queue or close the ticket.
- ITSM Tools support-
  - ServiceNow
  - BMC Remedy
  - Cherwell ITSM
  - BMC Remedyforce
  - Jira
  - ServiceXchange(SX)
- **Event Management Tools** – The purpose is to fetch the event data from the Event Management tool to understand the issue and recommend / trigger the relevant runbook for remediation. Event Management Tools support -
  - Moogsoft
  - Zenoss
- **RBA / Orchestrator Tools** – The purpose is to direct the RBA / orchestrator tools to trigger the runbook for resolving the incident, after BigFix Runbook AI has identified the appropriate runbook. BigFix Runbook AI also continuously pulls the current status of the execution from the RBA tool and reports it in its Logs section.
- RBA Tools support –
  - HCL BigFix

The subsequent sections will cover the integrations with above tools in detail.

4 Integration with IT Service Management Tools

Any IT Service management tool acts as a data source for BigFix Runbook AI from where it pulls the ticket data and then performs appropriate actions for resolution. Thus, to enable integration with ITSM, it requires for a data source to be created as part of BigFix Runbook AI configuration.

Given that the APIs for **Incident Management**, **Service Request Tasks** and **Change Request Tasks** are different, a separate data source will have to be configured for each of the previously mentioned categories.

Before proceeding with the configuration related to Data Source creation, user has to ensure that an organization has been configured. If not done already, please refer to the Configuration Guide for the same and create the organization before proceeding ahead

4.1 Common pre-requisite

- API to Fetch tickets, Ticket In progress, Ticket Close, Ticket Release
- USER permission to query, modification on Tickets

4.2 Integration with ServiceNow

4.2.1 Incident Management

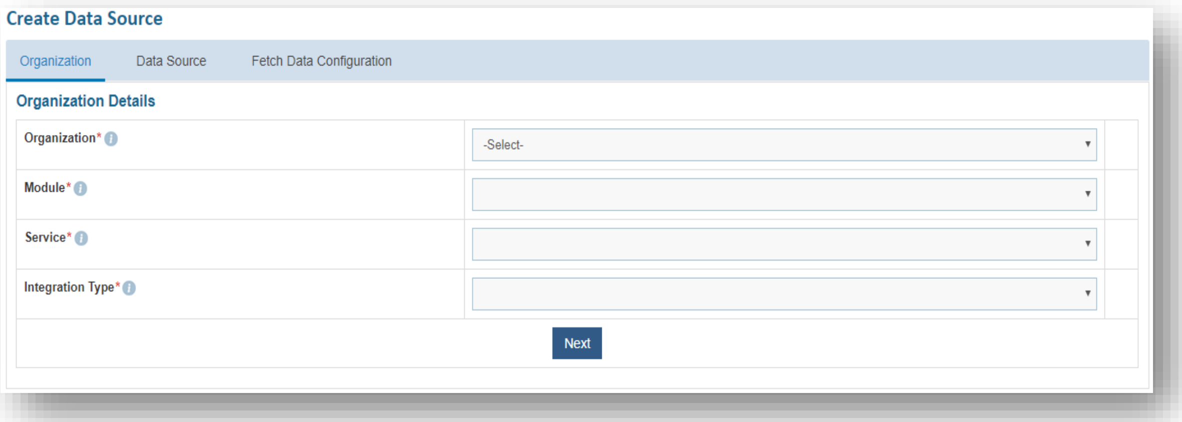
To fetch information about Incidents, usually, creation of a data source for Incident Management should suffice. However, there could be scenarios where some additional fields / values are required from CMDB for processing the tickets – recommending the relevant runbooks and parsing the tickets to extract relevant parameters, for which separate data sources for CMDB CI must be created. Here, we will cover the procedure for creating both kinds of data sources.

4.2.1.1 Create Data Source for Incident Management

To create a data source for Incident Management, perform the following steps:

- On the main menu bar, click **Actions Tab → Manage Data Sources**.
- The **Create Data Source** page appears with the following tabs:

- Organization
- Data Source
- Fetch Data Configuration
- Release Rules Configuration
- Close Rules Configuration (Optional – applicable only when the ticket closure status update is managed by BigFix Runbook AI directly instead of RBA tool)
- InProgress Rules Configuration (Optional – applicable only when the ticket's in progress status updates is managed by BigFix Runbook AI directly instead of RBA tool)



Create Data Source

Organization | Data Source | Fetch Data Configuration

Organization Details

Organization* : -Select-

Module* :

Service* :

Integration Type* :

Next

Figure 2 – Create Data Source

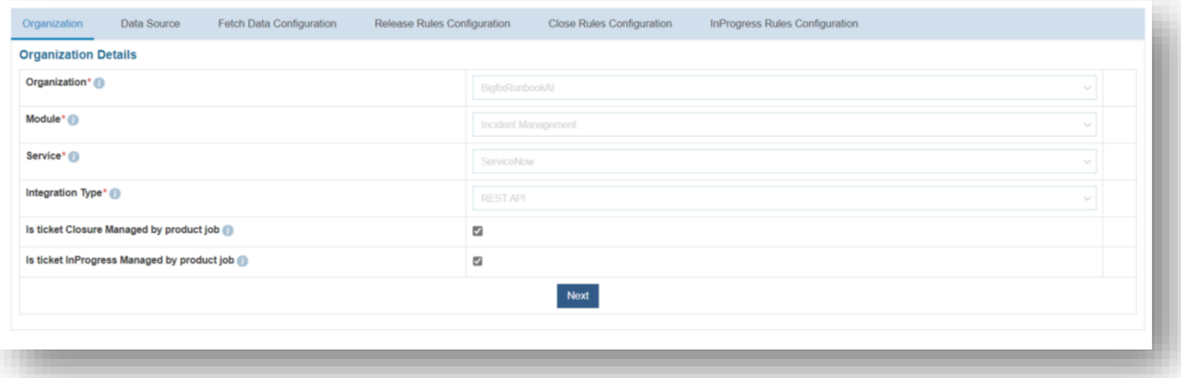
Release Rules Configuration is only applicable for the following **Module** types- **Incident Management, Change Request Task and Service Request Task.** This tab will not be activated for other module types.

– On the **Organization** tab,

- Select the **Organization Name** from the dropdown.
- Select the **Module** as **Incident Management,** since we are configuring this data source for pulling the incident tickets.
- Select the **Service** as **Service Now Tool** as we are configuring the data source for ServiceNow
- Select the **Integration Type** as **REST**, since we will be integrating through REST APIs.
- Check **Is Ticket Closure Managed by BigFix Runbook AI job** if you want BigFix Runbook AI to manage the ticket closure updates instead of the RBA tool. In this scenario, an additional tab **Close Rules Configuration** will be activated for providing further details, steps for which are mentioned later.
- Check "**Is ticket InProgress Managed by BigFix Runbook AI job**" if you want BigFix Runbook AI to manage the ticket's in progress status updates instead of the RBA tool. In this scenario, an

additional tab “**InProgress Rules Configuration**” will be activated for providing further details, steps for which are mentioned later.

- Click **Next**.


Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration
Next

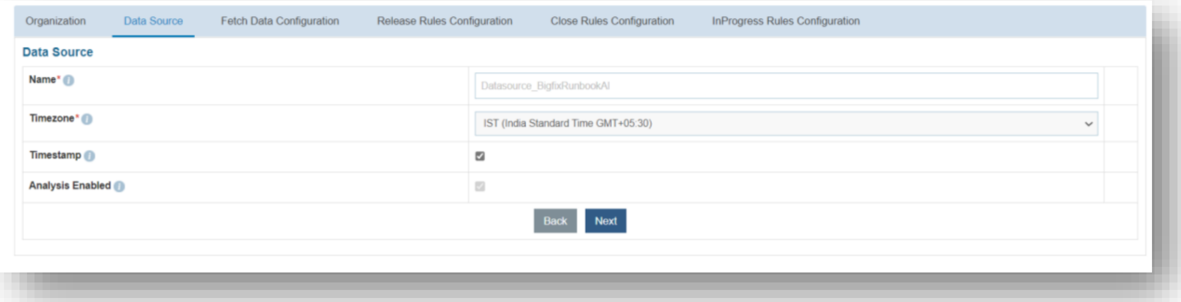
Organization Details

Organization*	BigfixRunbookAI
Module*	Incident Management
Service*	ServiceNow
Integration Type*	REST API
Is ticket Closure Managed by product job	☑
Is ticket InProgress Managed by product job	☑

Figure 3 – Create Data Source (cont.)

– On the **Data Source** tab,

- Type the new data source in the **Name** field.
- Select the **Timezone** to specify the time zone of the selected data source.
- Select **Timestamp** to view the present data with date and time.
- Select **Analysis Enabled** if user wants to analyze the data retrieved from the data source.
- Click **Next**.


Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration
Back
Next

Data Source

Name*	Datasource_BigfixRunbookAI
Timezone*	IST (India Standard Time GMT+05:30)
Timestamp	☑
Analysis Enabled	☑

Figure 4 – Create Data Source (cont.)

– On the **Fetch Data Configuration** tab, type in the details as per the environment.

– In the **Connection Details** section, enter the following details:

- **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.

- **Sample URL** - https://URL.service-now.com/api/now/v1/table/incident?sysparm_fields=#Columns#&sysparm_query=sys_updated_on>=#StartDate#^sys_updated_on<=#EndDate#^ORDERBYsys_updated_on
- **Authentication Type** – Select one of the Authentication Types from Basic / Windows, JWT, OAuth 2.0

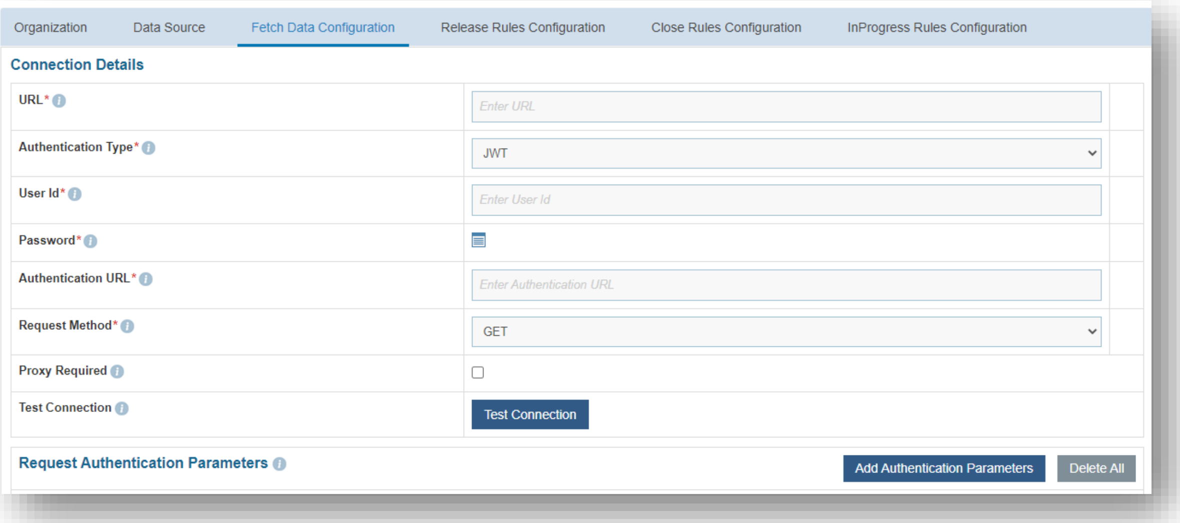
  Selection of **Basic / Windows** requires you to enter -

  - User Id
  - Password.

  Selection of **JWT / OAuth 2.0** requires you to enter -

  - User Id
  - Password
  - Authentication URL

- **Request Method –** Select **GET, POST** or **PUT** as the Request method as per the URL configured.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.


Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

Connection Details

URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	GET
Proxy Required	☐
Test Connection	Test Connection

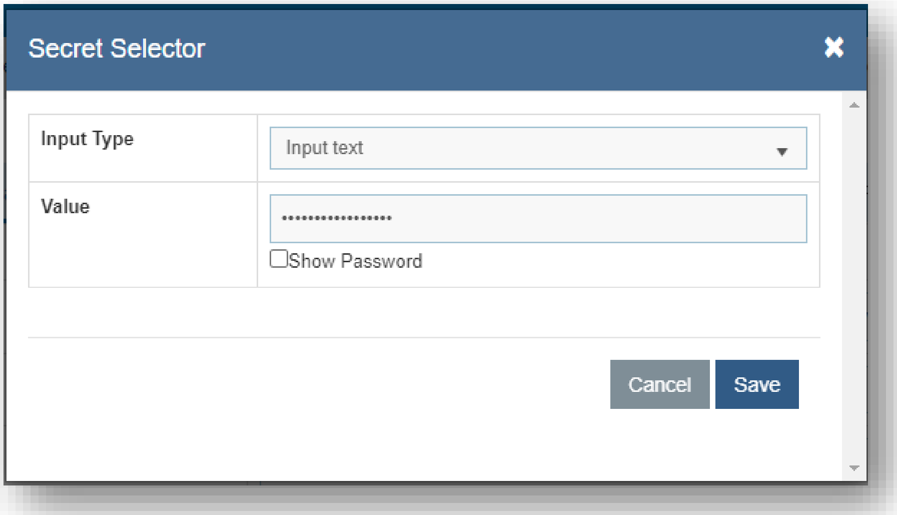
Request Authentication Parameters

Add Authentication Parameters
Delete All

Figure 5 – Create Data Source (Connection Details)

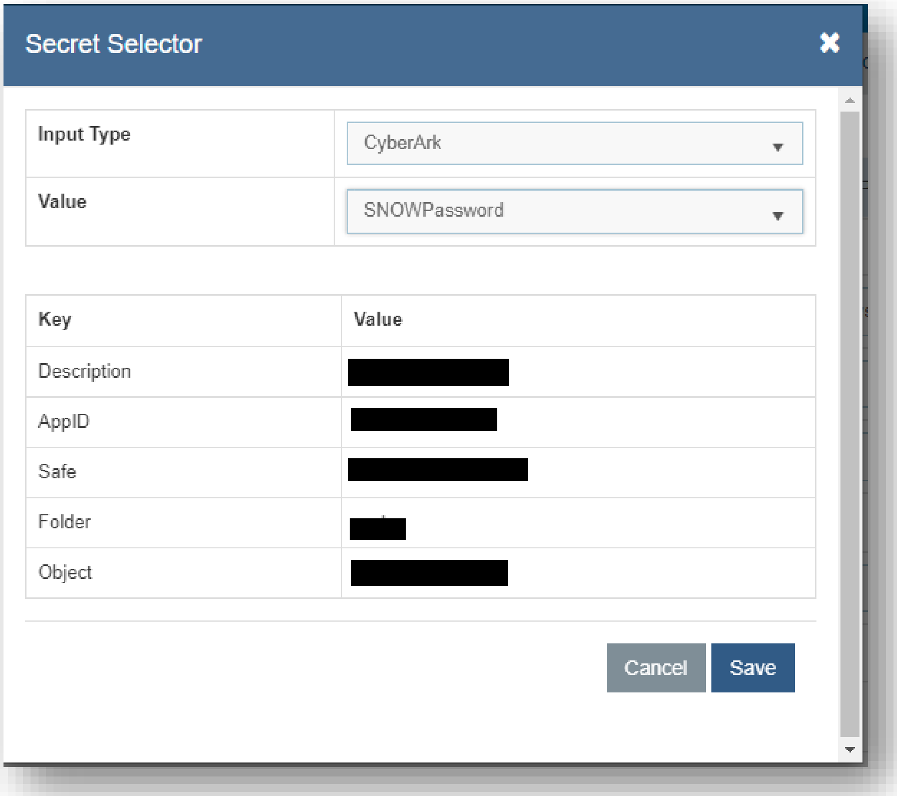
For **password**, click on the icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in

any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Input Type	Input text
Value	••••••••••••••••• □Show Password

Figure 6 – Password in Plaintext



Input Type	CyberArk
Value	SNOWPassword

Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

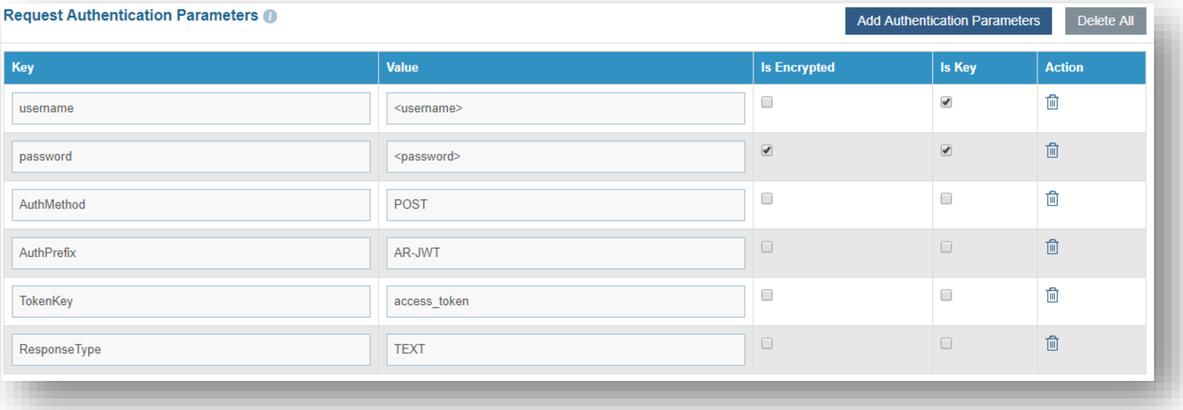
Figure 7 – Password from Key Vault (CyberArk)

– **Request Authentication Parameters** – If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.

– Based on the **Authentication Type**, add the parameters mentioned in the below table.

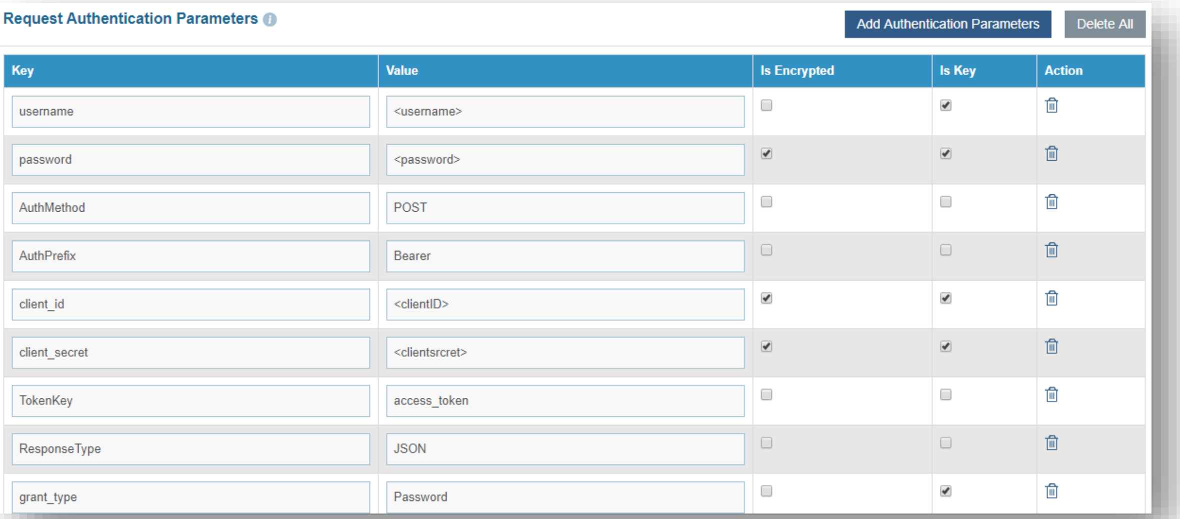
Table 2– Sample Authentication Parameters

Authentication Type	Key	Value	Is Encrypted?	Is Key?
JWT	username	<username>	NO	YES
JWT	password	<password>	YES	YES
JWT	AuthMethod	POST	NO	NO
JWT	AuthPrefix	AR-JWT	NO	NO
JWT	TokenKey	access_token	NO	NO
JWT	ResponseType	TEXT	NO	NO
OAuth2.0	username	<username	NO	YES
OAuth2.0	password	<password>	YES	YES
OAuth2.0	AuthMethod	POST	NO	NO
OAuth2.0	AuthPrefix	Bearer	NO	NO
OAuth2.0	client_id	<clientID>	YES	YES
OAuth2.0	client_secret	<clientsecret>	YES	YES
OAuth2.0	TokenKey	access_token	NO	NO
OAuth2.0	ResponseType	JSON	NO	NO
OAuth2.0	grant_type	Password	NO	YES


Request Authentication Parameters
Add Authentication Parameters
Delete All

Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	AR-JWT	☐	☐	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	TEXT	☐	☐	🗑

Figure 8 – Create Data Source (Request Authentication Parameters for JWT)



Request Authentication Parameters

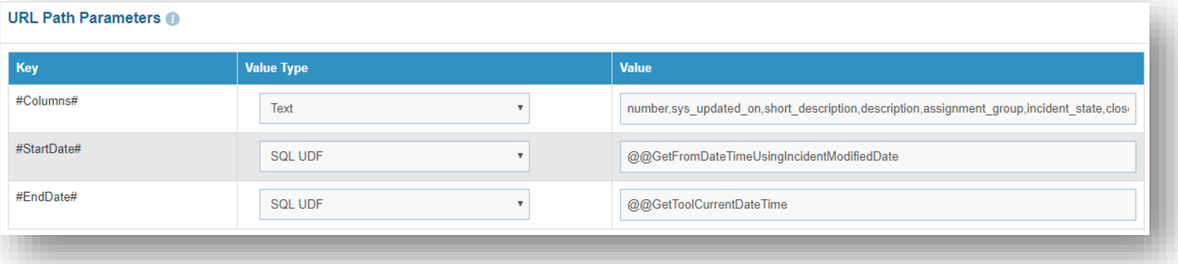
Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	Bearer	☐	☐	🗑
client_id	<clientID>	☑	☑	🗑
client_secret	<clientsrcret>	☑	☑	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	JSON	☐	☐	🗑
grant_type	Password	☐	☑	🗑

Figure 9 – Create Data Source (Request Authentication Parameters for OAuth2.0)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #Columns#

ValueType: Text

Value:

number,sys_updated_on,short_description,description,assignment_gro
up,incident_state,closed_at,category,dv_assigned_to,sys_id


Note - These columns are mandatory. User can add more columns if
more data is required to be fetched from ITSM tool.


Key: #StartDate#

ValueType: SQL UDF

VALUE: @@GetFromDateTimeUsingIncidentModifiedDate


Key: #EndDate#
```

```
ValueType: SQL UDF

VALUE: @@GetToolCurrentDateTime
```



URL Path Parameters

Key	Value Type	Value
#Columns#	Text	number,sys_updated_on,short_description,description,assignment_group,incident_state,clos
#StartDate#	SQL UDF	@@GetFromDateTimeUsingIncidentModifiedDate
#EndDate#	SQL UDF	@@GetToolCurrentDateTime

Figure 10– URL Path Parameters

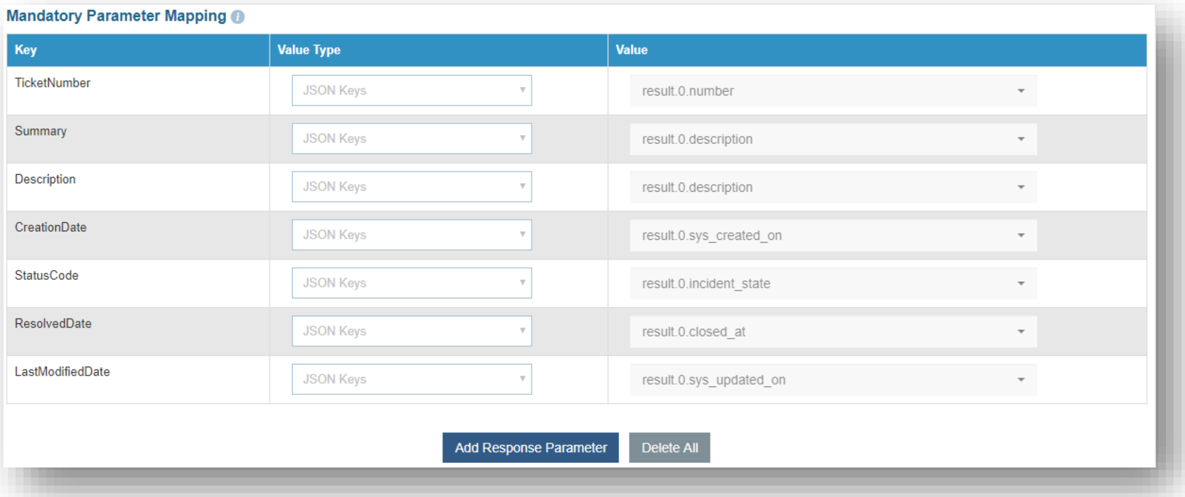
- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body –** Please enter the request body for the configured URL, if applicable.
- **Response Body –** In this section, please enter the output of URL query for one of the incidents in JSON format. A sample response is mentioned below.

```
Response Body –

{  "result": [{   "number": "INC0079154",   "closed_at": "",
"assignment_group": {    "link": "<https://sample.service-
now.com/api/now/v1/table/sys_user_group/All user group>",
"value": "All user group"   },   "incident_state": "6",
"sys_created_on": "2017-12-22 06:59:03",   "description": "Memory
Utilization:10.0.0.11", "short_description": "Memory
Utilization:localhost",  "sys_updated_on": "2018-01-02 06:39:56",
"category": "",   "priority": "4",   "sys_id": "123456"  }] }
```

- After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section**.**
- **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below:

Table 3– Sample Mandatory Parameter Mapping

Key	Value Type	Value
TicketNumber	JSON.Keys	result.0.number
Summary	JSON.Keys	result.0.short_description
Description	JSON.Keys	result.0.description
CreationDate	JSON.Keys	result.0.sys_created_on
StatusCode	JSON.Keys	result.0.incident_state
ResolvedDate	JSON.Keys	result.0.closed_at
LastModifiedDate	JSON.Keys	result.0.sys_updated_on



Mandatory Parameter Mapping

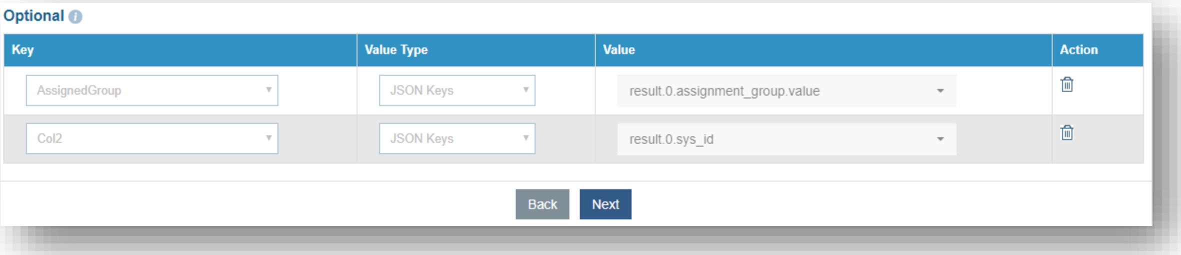
Key	Value Type	Value
TicketNumber	JSON Keys	result.0.number
Summary	JSON Keys	result.0.description
Description	JSON Keys	result.0.description
CreationDate	JSON Keys	result.0.sys_created_on
StatusCode	JSON Keys	result.0.incident_state
ResolvedDate	JSON Keys	result.0.closed_at
LastModifiedDate	JSON Keys	result.0.sys_updated_on

Figure 11 – Mandatory Parameter Mapping

– If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 4– Sample Optional Parameters

Key	Value Type	Value
AssignedGroup	JSON.Keys	result.0.assignment_group.value
Col1	JSON.Keys	result.0.sys_id

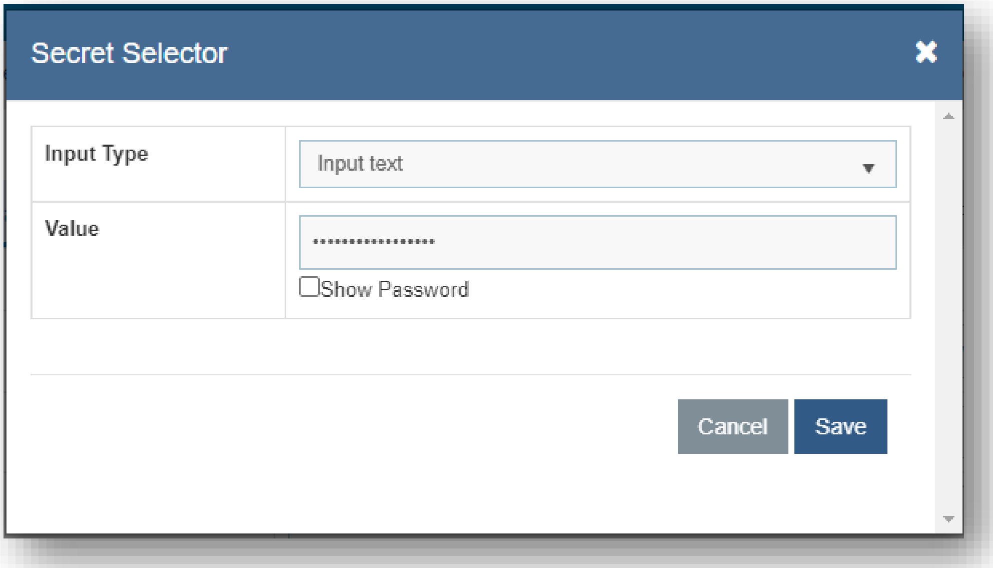


Optional

Key	Value Type	Value	Action
AssignedGroup	JSON Keys	result.0.assignment_group.value	🗑
Col2	JSON Keys	result.0.sys_id	🗑

Figure 12 – Optional Parameter Mapping

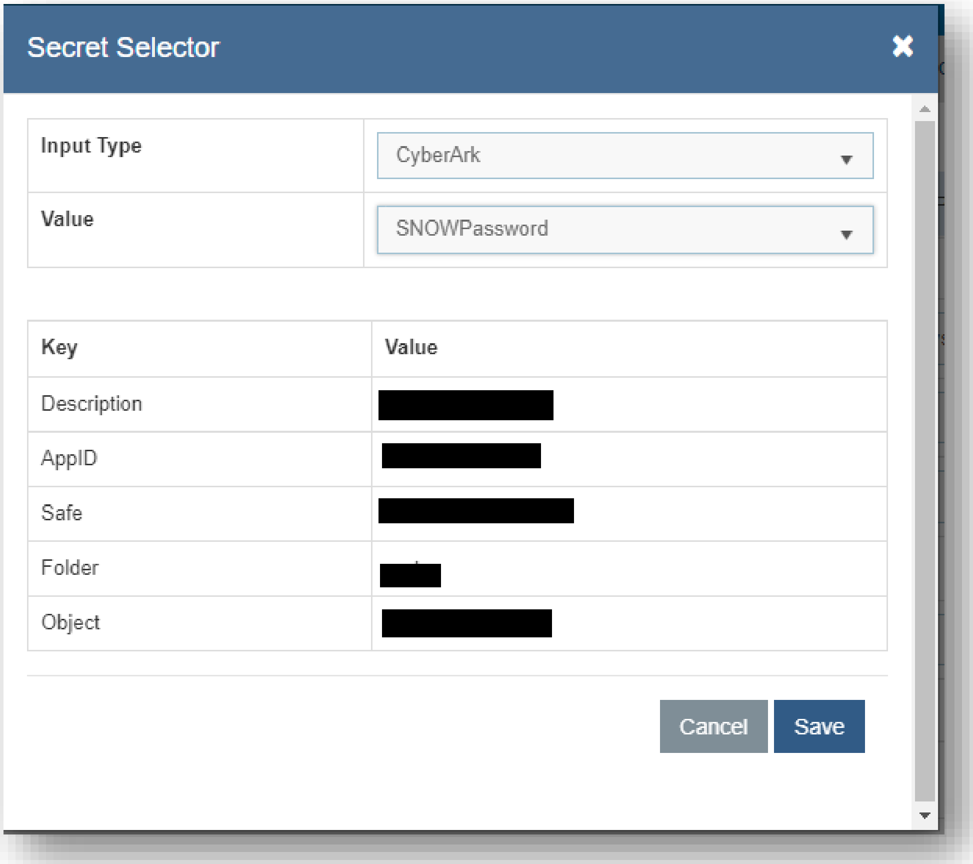
- Click Next to proceed to Release Rules Configuration.
- On **Release Rules Configuration** tab, type in the details as per the requirement.
- In the **Connection Details** section, enter the following details:
  - **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.
  - Sample URL - https://<URL>.service-now.com/api/now/table/incident/#incident#
  - **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously.
  - **User Id**: Enter the user id for the configured ITSM.
  - **Password**: For password, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Secret Selector

Input Type	Input text
Value	•••••••••••••••••• □ Show Password

Figure 13 – Password in Plaintext



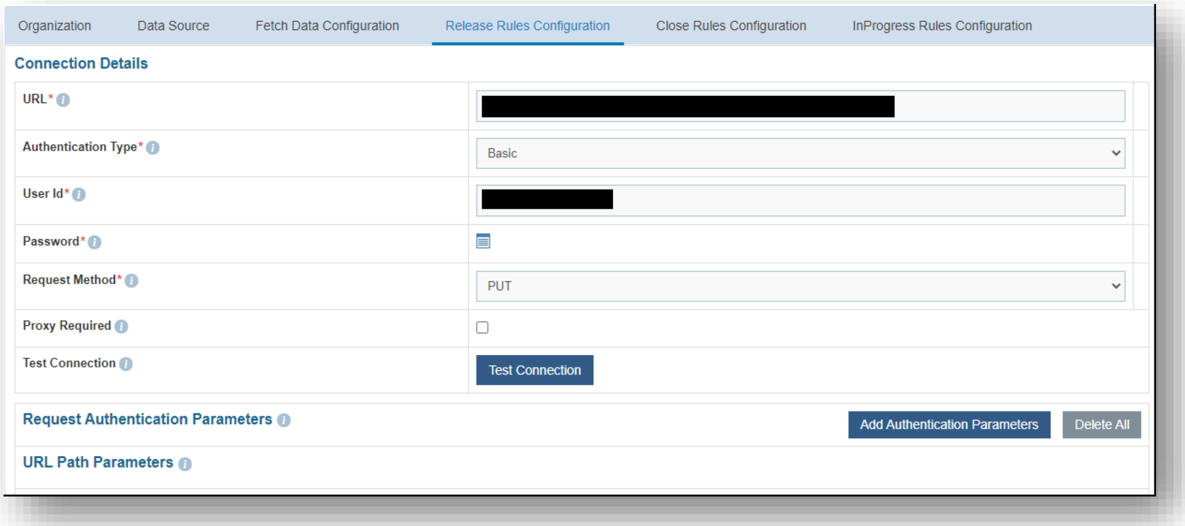
Secret Selector

Input Type	CyberArk
Value	SNOWPassword

Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 14 – Password from Key Vault (CyberArk)

- **Request Method** – Select Request Method as PUT from the drop-down.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.


Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

Connection Details

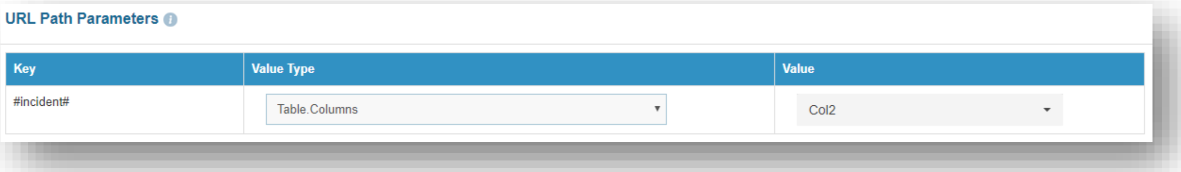
URL*	[redacted]
Authentication Type*	Basic
User Id*	[redacted]
Password*	
Request Method*	PUT
Proxy Required	☐
Test Connection	Test Connection

Request Authentication Parameters
Add Authentication Parameters
Delete All
URL Path Parameters

Figure 15 – Release Rules Configuration (Connection Details)

- **URL Path Parameters** – Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #incident#

ValueType: Table Columns

Value:

Select from dropdown that mapped to sys_id from previous screen
"Col2"
```

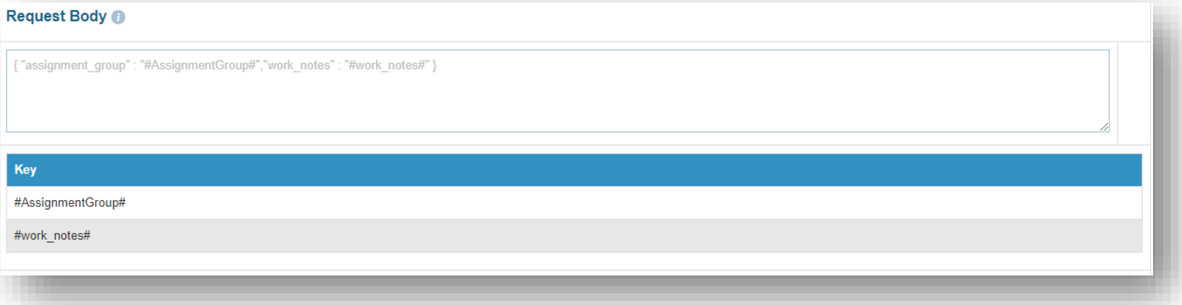

URL Path Parameters

Key	Value Type	Value
#incident#	Table.Columns	Col2

Figure 16 – Release Rules Configuration (URL Path Parameters)

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body –** In this section, please enter the request body in JSON format. A sample request is mentioned below:

```
Request Body –
{ "assignment_group" : "#AssignmentGroup#","work_notes" :
"#work_notes#" }
```



Request Body

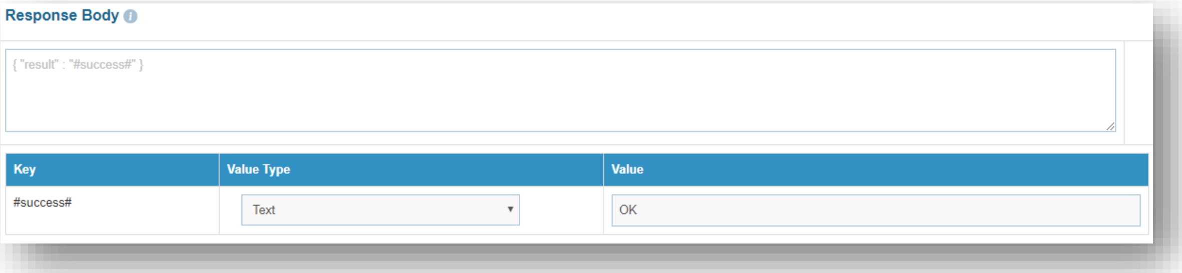
{ "assignment_group" : "#AssignmentGroup#","work_notes" : "#work_notes#" }

Key
#AssignmentGroup#
#work_notes#

Figure 17 – Release Rules Configuration (Request Body)

- **Response Body –** In this section, please enter the response body in JSON format. A sample response is mentioned below.

```
Response Body –
{ "result" : "#success#" }
```



Response Body

{ "result" : "#success#" }

Key	Value Type	Value
#success#	Text	OK

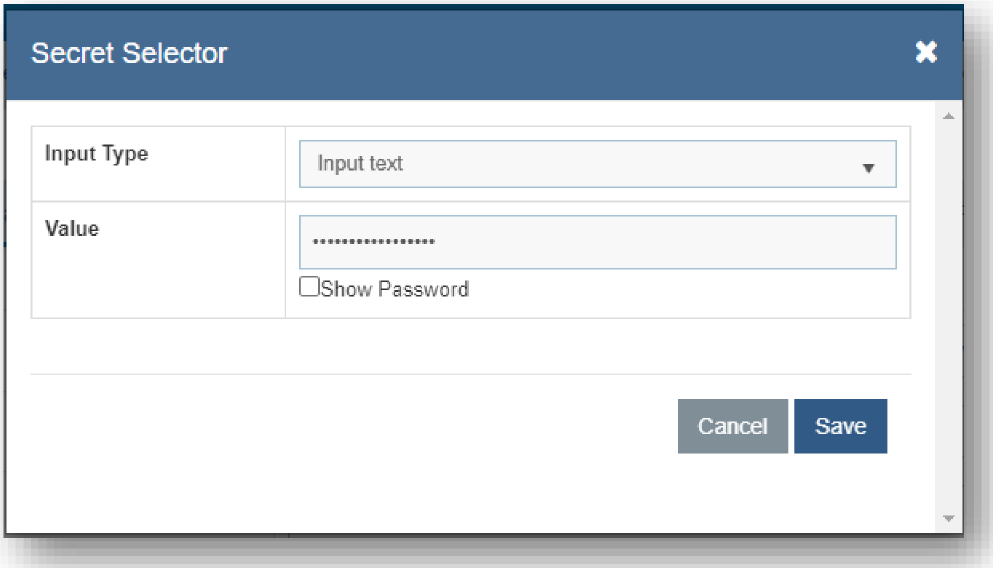
Figure 18 – Release Rules Configuration (Response Body)

- **Response Key Value** mapping can be done as per the below table.

Table 5– Sample Response Key Value Mapping

#success#	Text	OK

- On **Close Rules Configuration** tab, type in the details as per the requirement. Check **Same as Release** if similar configurations as mentioned in "Release Rules Configuration" are required, else proceed ahead.
- In the **Connection Details** section, enter the following details:
  - **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.
  - **Sample URL** - https://<url>.service-now.com/api/now/table/incident/#incident#
  - **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously.
  - **User Id**: Enter user id for the configured ITSM tool.
  - **Password**: For password, click on icon next to it. If the password is available in plaintext, then select Input Text as **Input Type** and enter the password in **Value** field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



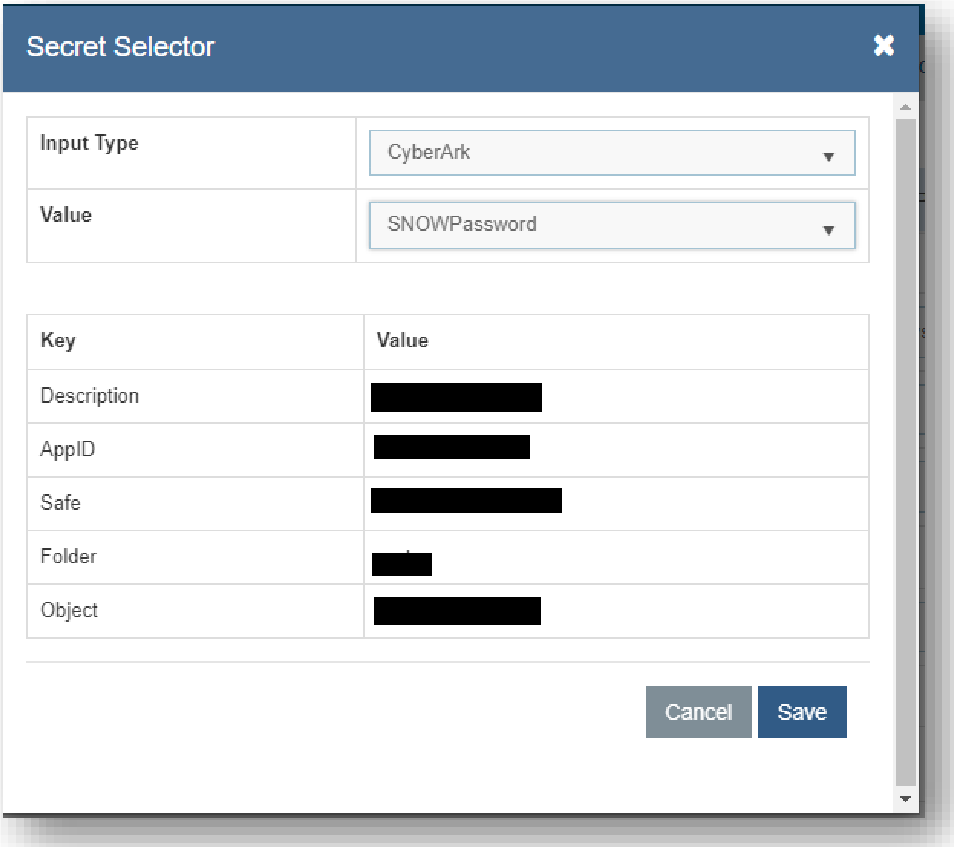
Secret Selector

Input Type: Input text

Value: ••••••••••••••••

□ Show Password

Figure 19 – Password in plaintext



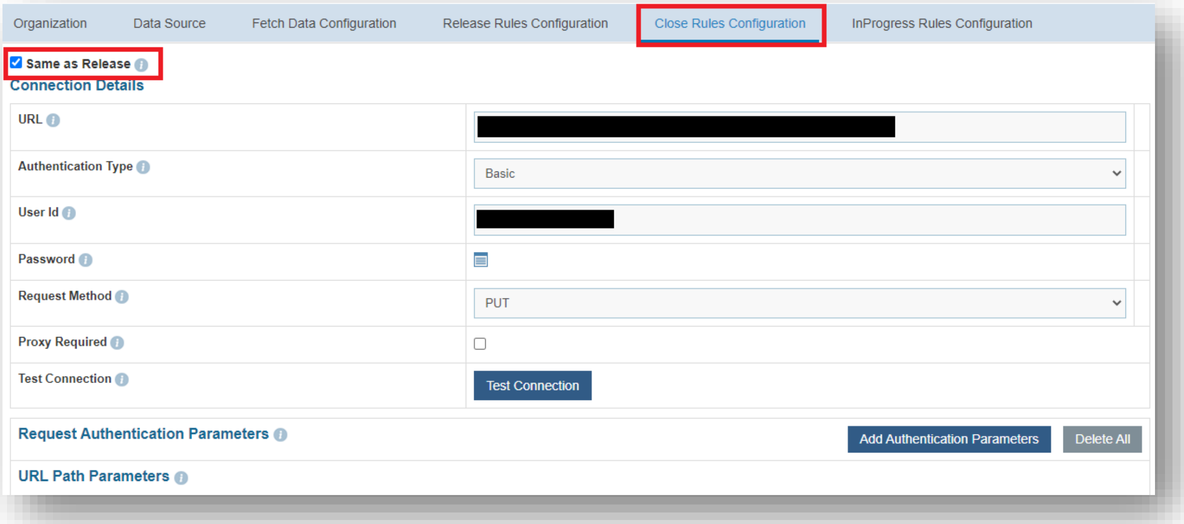
Secret Selector

Input Type	CyberArk
Value	SNOWPassword

Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 20 – Password from Key Vault (CyberArk)

- **Request Method** – Select Request Method as PUT from the drop-down.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.


Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration
Test Connection
Add Authentication Parameters
Delete All

☑ Same as Release

Connection Details

URL	[redacted]
Authentication Type	Basic
User Id	[redacted]
Password	
Request Method	PUT
Proxy Required	□
Test Connection	Test Connection

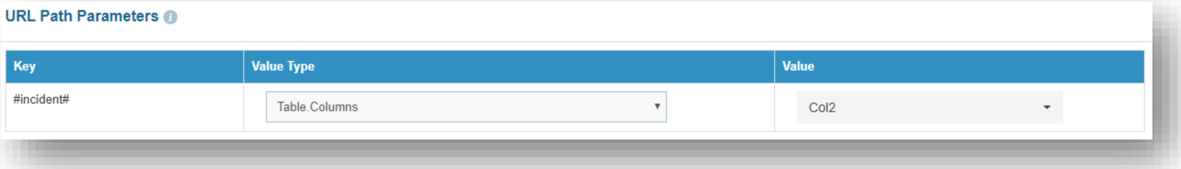
Request Authentication Parameters

URL Path Parameters

Figure 21 – Close Rules Configuration (Connection Details)

– **URL Path Parameters** – Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #incident#

ValueType: Table Columns

Value:

Select from dropdown that mapped to sys_id from previous screen
“Col2”
```



URL Path Parameters

Key	Value Type	Value
#incident#	Table.Columns	Col2

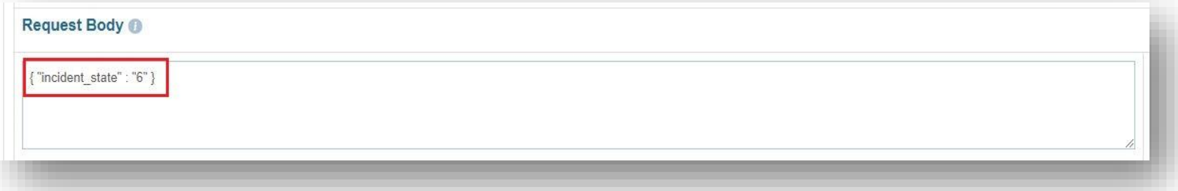
Figure 22 – Close Rules Configuration (URL Path Parameters)

– **Request Header Parameters –** Please enter the request header parameters as required.

– **Request Body –** In this section, please enter the request body in JSON format. A sample request is mentioned below.

```
Request Body –
```

```
{ "incident_state" : "6"} If you also want to add worknotes while
Close ticket, use json {"incident_state":"6", "work_notes":
"#Notes#"}
```



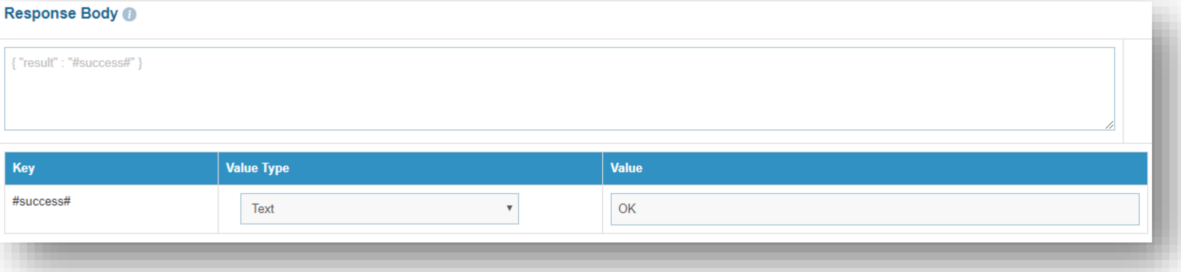
Request Body

{ "incident_state" : "6" }

Figure 23 – Close Rules Configuration (Request Body)

- **Response Body –** In this section, please enter the response body in JSON format. A sample response is mentioned below.

```
Response Body –

{ "result" : "#success#" }
```



Response Body

{ "result" : "#success#" }

Key	Value Type	Value
#success#	Text	OK

Figure 24 – Close Rules Configuration (Response Body)

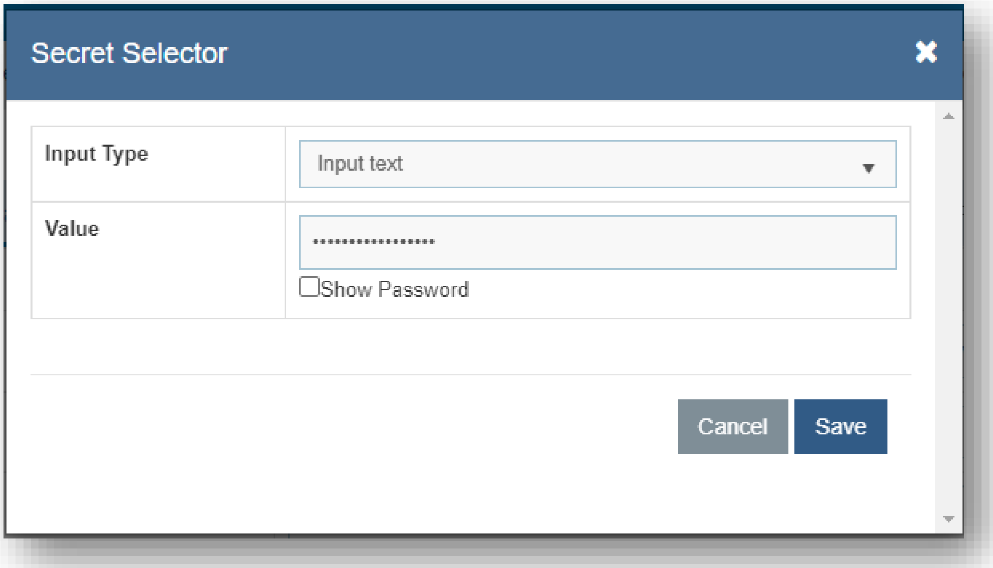
- **Response Key Value** mapping can be done as per the below table.

Table 6– Sample Response Key Value Mapping

#success#	Text	OK

- On **InProgress Rules Configuration** tab, type in the details as per the requirement. Check **Same as Release** if similar configurations as mentioned in “Release Rules Configuration” are required, else proceed ahead.
- In the **Connection Details** section, enter the following details:

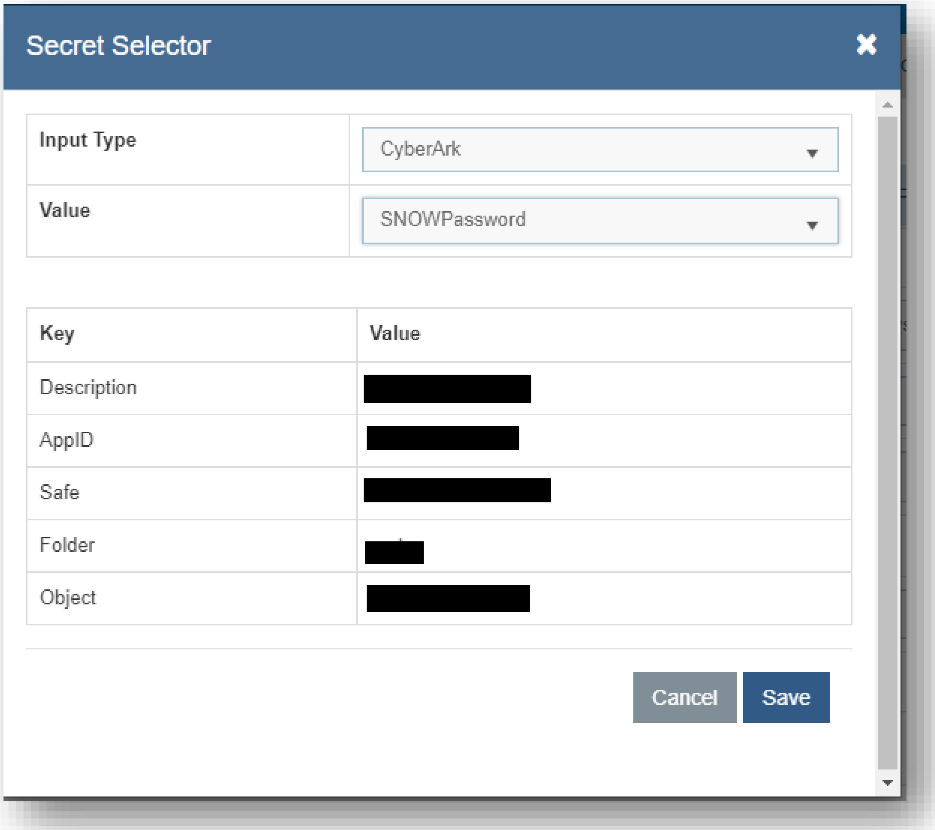
- **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.
- **Sample URL** - https://<url>.service-now.com/api/now/table/incident/#incident#
- **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously.
- **User Id** – Enter the user id for the configured ITSM tool.
- **Password**– For password, click on icon next to it. If the password is available in plaintext then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Secret Selector

Input Type	Input text
Value	••••••••••••••••• □Show Password

Figure 25 – Password in plaintext

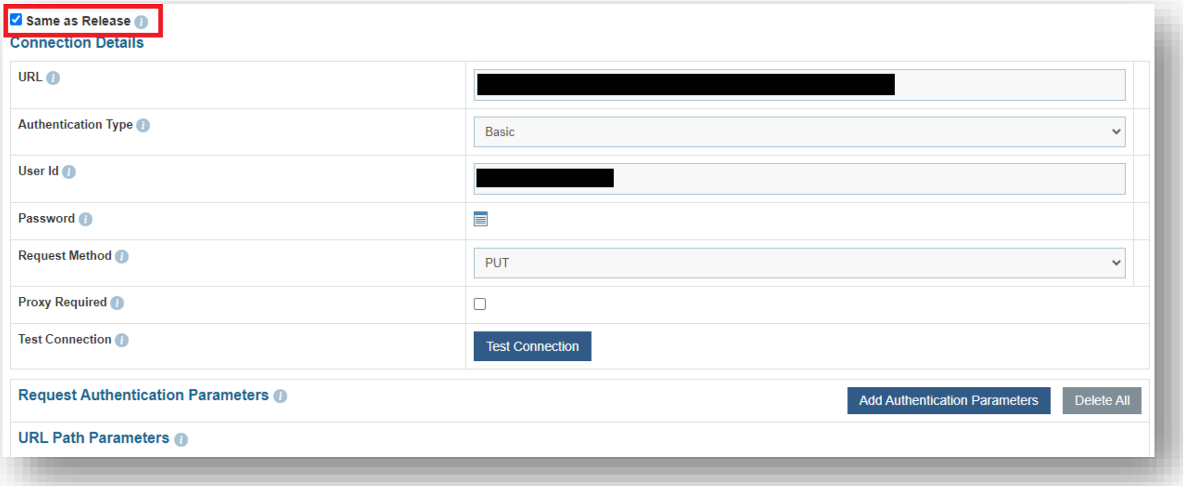


Input Type	CyberArk
Value	SNOWPassword

Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 26 – Password from Key Vault (CyberArk)

- **Request Method** – Select Request Method as PUT from the drop-down.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



☑ Same as Release

Connection Details

URL	[redacted]
Authentication Type	Basic
User Id	[redacted]
Password	
Request Method	PUT
Proxy Required	☐
Test Connection	Test Connection

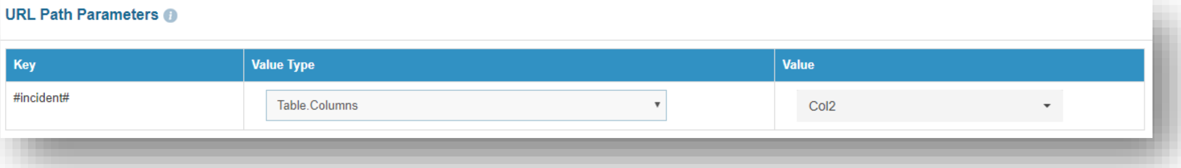
Request Authentication Parameters

URL Path Parameters

Figure 27 – InProgress Rules Configuration (Connection Details)

- **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #incident#

ValueType: Table Columns

Value:

Select from dropdown that mapped to sys_id from previous screen
"Col2"
```



URL Path Parameters

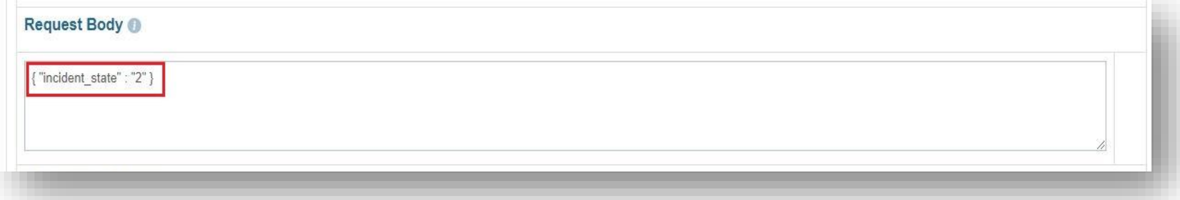
Key	Value Type	Value
#incident#	Table.Columns	Col2

Figure 28 – InProgress Rules Configuration (URL Path Parameters)

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body –** In this section, please enter the request body in JSON format. A sample request is mentioned below:

```
Request Body –
```

```
{"incident_state" : "2"} If you also want to add worknotes while inprogress ticket, use json {"incident_state":"2", "work_notes": "#Notes#"}
```

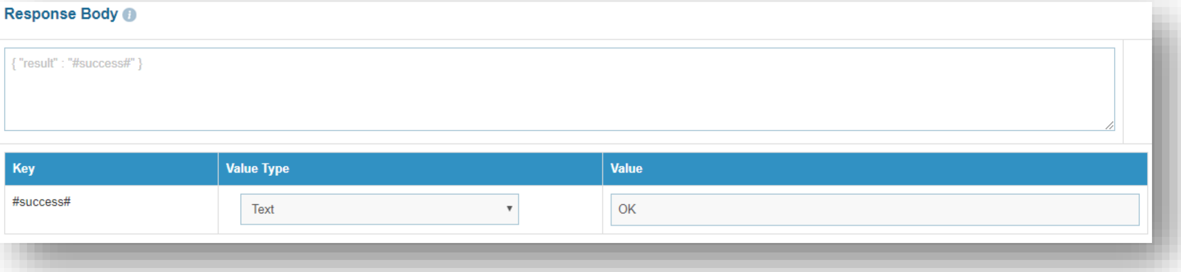

Request Body

{"incident_state" : "2"}

Figure 29 – InProgress Rules Configuration (Request Body)

– **Response Body –** In this section, please enter the response body in JSON format. A sample response is mentioned below.

```
Response Body –

{ "result" : "#success#" }
```


Response Body

{ "result" : "#success#" }

Key	Value Type	Value
#success#	Text	OK

Figure 30 – InProgress Rules Configuration (Response Body)

– **Response Key Value** mapping can be done as per the below table:

Table 7– Sample Response Key Value Mapping

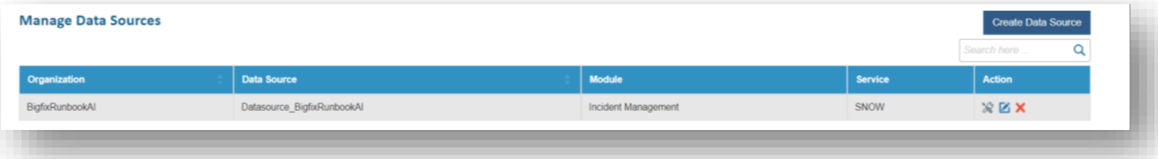
#success#	Text	OK

– Click **Submit** to add the data source.

– In order to bring the tickets within BigFix Runbook AI scope, a specific queue needs to be configured in the ITSM tool and the same has to be configured in BigFix Runbook AI. This is achieved through **Manage the Entry Criteria**. Please perform the below steps –

– Go to Action Tab and click Manage Data Sources.

- On the **Data Sources** tab, click ⚒ next to the data source that user wants to manage. **Manage Entry Criteria** screen appears.

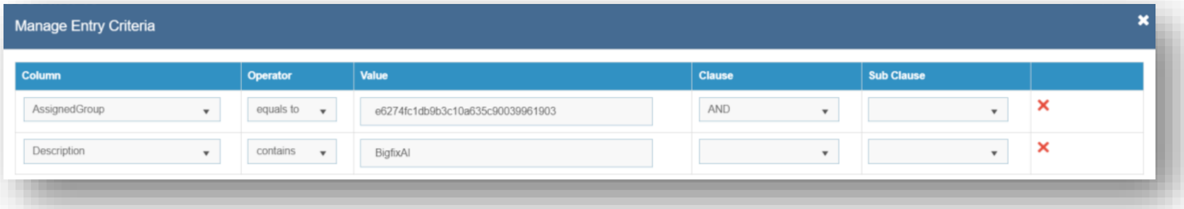


Manage Data Sources

Organization	Data Source	Module	Service	Action
BigfixRunbookAI	Datasource_BigfixRunbookAI	Incident Management	SNOW	⚒ ✎ ✕

Figure 31 – Manage Entry Criteria

- Select ‘AssignedGroup’ for the **Column** field and ‘equals to’ for the **Operator** field.
- Enter the sys_id of the assignment group in ServiceNow in the **Value** field.
- **Clause** and **Sub-Clause** fields can also be added based on requirement.



Manage Entry Criteria

Column	Operator	Value	Clause	Sub Clause	
AssignedGroup	equals to	e6274fc1db9b3c10a635c90039961903	AND		✕
Description	contains	BigfixAI			✕

Figure 32 – Manage Entry Criteria (cont.)

- Click **Save**.

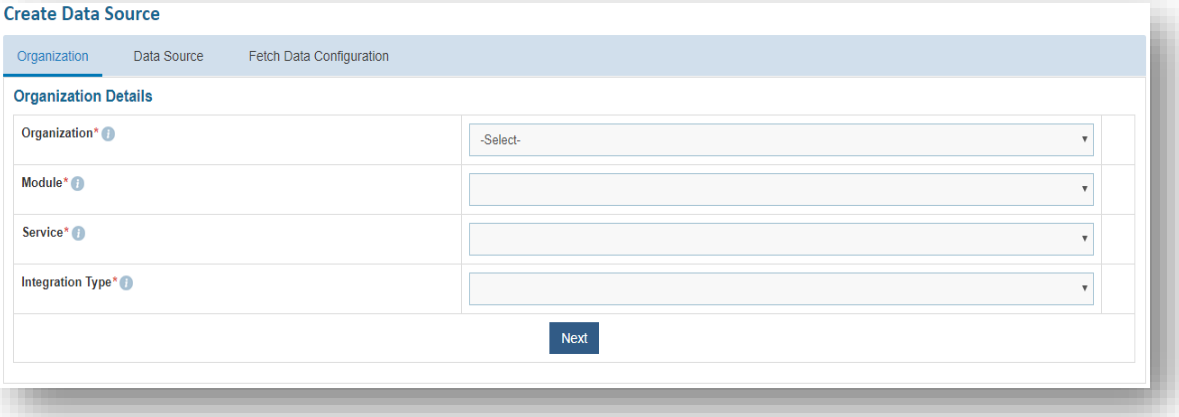
4.2.1.2 Create Data Source for CMDB CI

To use the field values of CMDB CI for the purpose of Recommendation and Parsing by BigFix Runbook AI services, two data sources need to be created.

To create a data source for CMDB CI, please refer to Create Data Source for Incident Management.

To create a data source for CMDB CI, perform the following steps:

- On the main menu bar, click **Actions Tab → Manage Data Sources**.
- The **Create Data Source** page appears with the following tabs:
  - Organization
  - Data Source
  - Fetch Data Configuration



Create Data Source

Organization Details

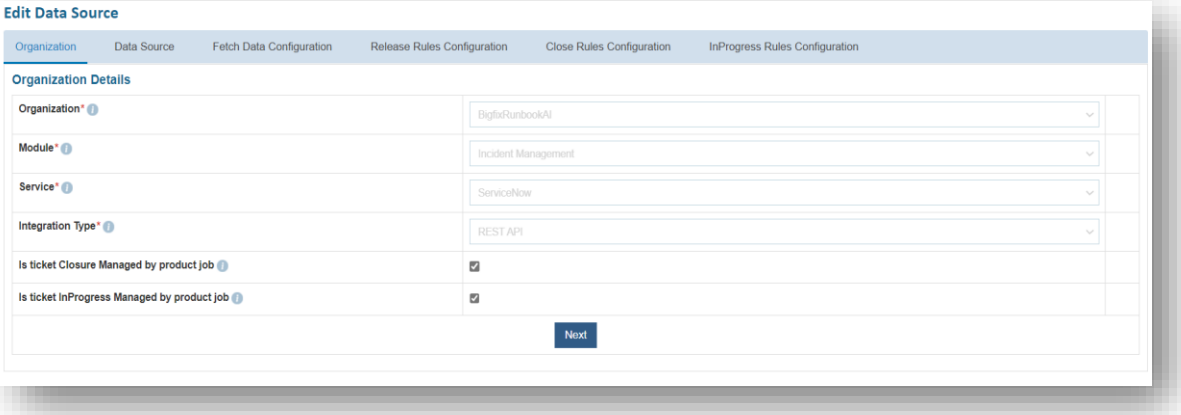
Organization*	-Select-
Module*	
Service*	
Integration Type*	

Figure 33 – Create Data Source – CMDB CI

Release Rules Configuration is only applicable for the following **Module** types:
- Incident Management,
- Change Request Task and
- Service Request Task. (This tab will not be activated for other module types.)

– On the **Organization** tab:

- Select the **Organization Name** from the dropdown.
- Select the **Module** as **CMDB CI,** since we are configuring this data source for using its field value for the incidents.
- Select the **Service** as **Service Now Tool** as we are configuring the data source for ServiceNow
- Select the **Integration Type** as **REST**, since we will be integrating through REST APIs.
- Click **Next**.



Edit Data Source

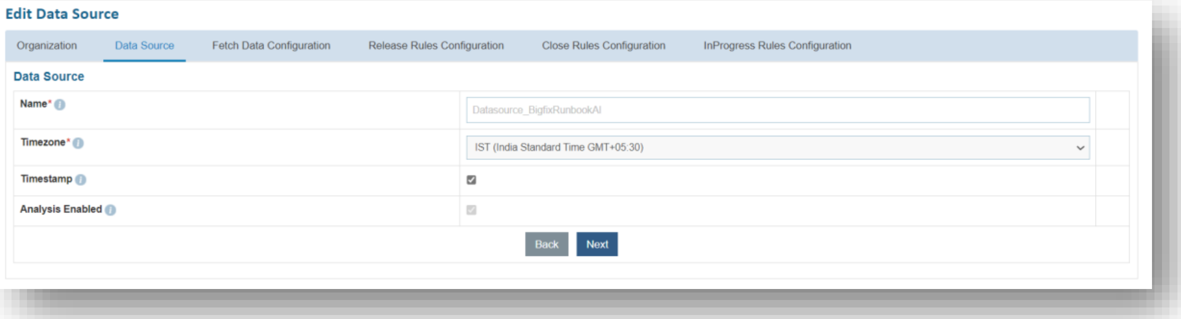
Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

Organization Details

Organization*	BigfixRunbookAI
Module*	Incident Management
Service*	ServiceNow
Integration Type*	REST API
Is ticket Closure Managed by product job	☑
Is ticket InProgress Managed by product job	☑

Figure 34 – Create Data Source – CMDB CI (cont.)

- On the **Data Source** tab:
    - Type the new data source in the **Name** field.
    - Select the **Timezone** to specify the time zone of the selected data source.
    - Select **Timestamp** to view the present data with date and time.
    - Select **Analysis Enabled** if user wants to analyze the data retrieved from the data source.
    - Click **Next**.



Edit Data Source

Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

Data Source

Name*	Datasource_BigfixRunbookAI
Timezone*	IST (India Standard Time GMT+05:30)
Timestamp	☑
Analysis Enabled	☑

Figure 35 – Create Data Source – CMDB CI (cont.)

- On the **Fetch Data Configuration** tab, type in the details as per the environment.
- In the **Connection Details** section, enter the following details:
    - **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.
    - **Sample URL -** https://URL.service-now.com/api/now/v1/table/cmdb_ci_server?sysparm_fields=#Columns#&sysparm_query=sys_updated_on>=#StartDate#^sys_updated_on<=#EndDate#^ORDERBYsys_updated_on
    - **Authentication Type** – Select one of the Authentication Types from Basic / Windows, JWT, OAuth 2.0

      Selection of **Basic / Windows** requires you to enter -

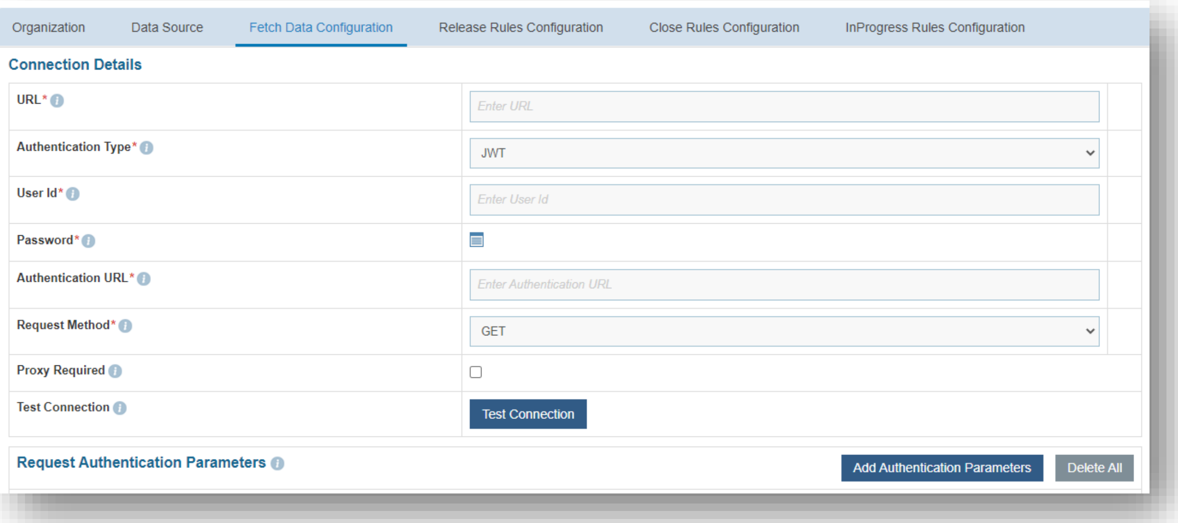
        - User Id
        - Password.

      Selection of **JWT / OAuth 2.0** requires you to enter -

        - User Id
        - Password

  - Authentication URL

- **Request Body -** Select GET, POST or PUT as Request Method as per the configured URL.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.


Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

Connection Details

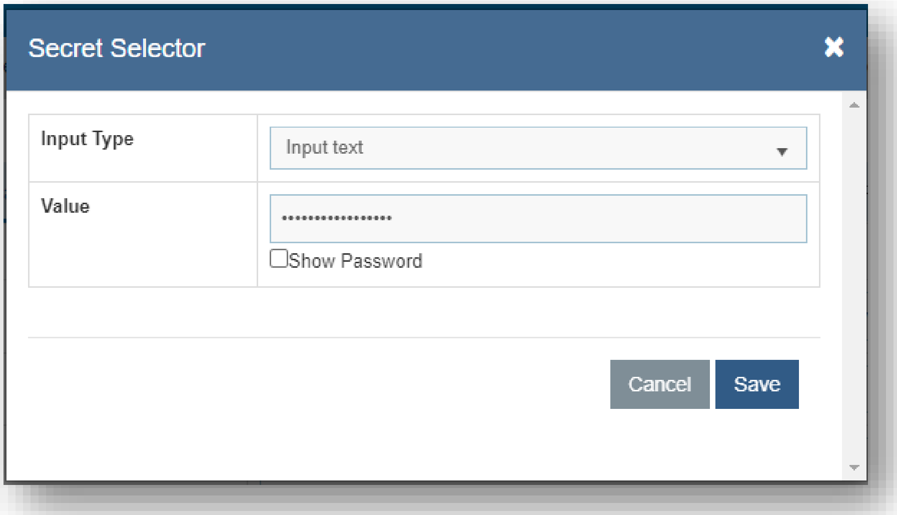
URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	▤
Authentication URL*	Enter Authentication URL
Request Method*	GET
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

Add Authentication Parameters
Delete All

Figure 36 – Create Data Source – CMDB CI (Connection Details)

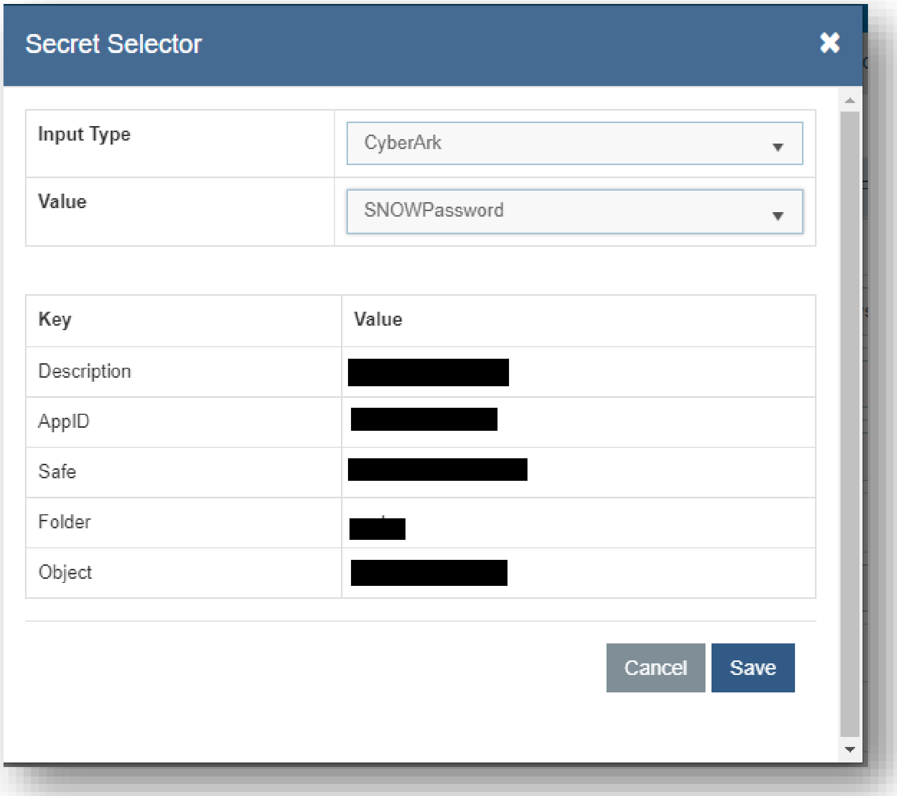
- **Password** – For password, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Secret Selector

Input Type	Input text
Value	•••••••••••••••• □ Show Password

Figure 37 – Password in plaintext



Secret Selector

Input Type	CyberArk
Value	SNOWPassword

Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

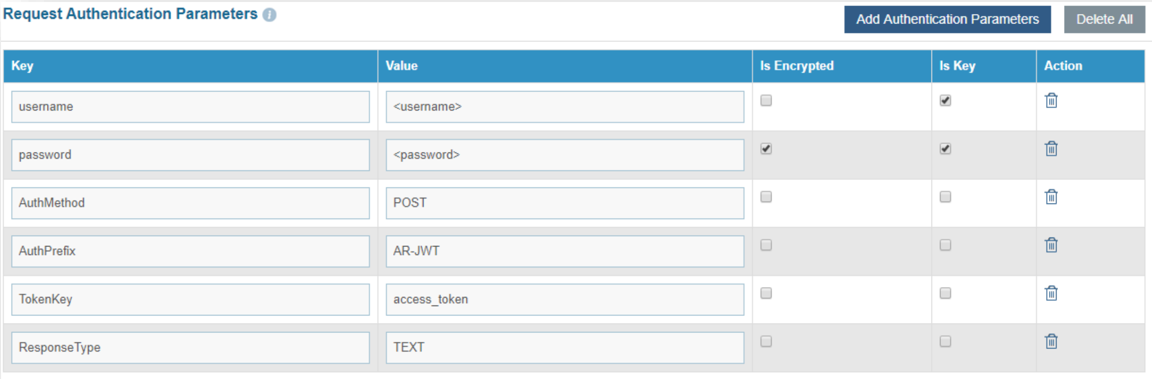
Figure 38 – Password from Key Vault (CyberArk)

- **Request Authentication Parameters** – If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.

– Based on the **Authentication Type**, add the parameters mentioned in the below table.

Table 8– Sample Authentication Parameters – CMDB CI

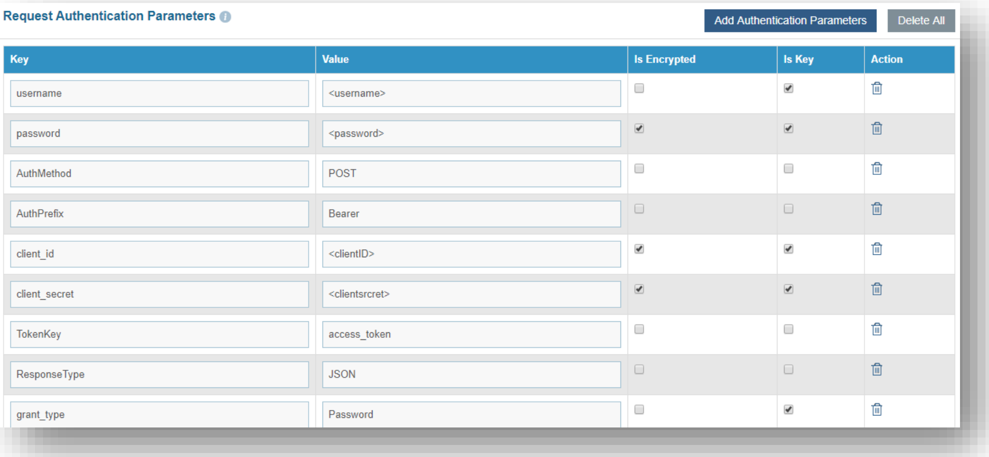
Authentication Type	Key	Value	Is Encrypted?	Is Key?
JWT	username	<username>	NO	YES
JWT	password	<password>	YES	YES
JWT	AuthMethod	POST	NO	NO
JWT	AuthPrefix	AR-JWT	NO	NO
JWT	TokenKey	access_token	NO	NO
JWT	ResponseType	TEXT	NO	NO
OAuth2.0	username	<username	NO	YES
OAuth2.0	password	<password>	YES	YES
OAuth2.0	AuthMethod	POST	NO	NO
OAuth2.0	AuthPrefix	Bearer	NO	NO
OAuth2.0	client_id	<clientID>	YES	YES
OAuth2.0	client_secret	<clientsecret>	YES	YES
OAuth2.0	TokenKey	access_token	NO	NO
OAuth2.0	ResponseType	JSON	NO	NO
OAuth2.0	grant_type	Password	NO	YES



Request Authentication Parameters
Add Authentication Parameters
Delete All

Key	Value	Is Encrypted	Is Key	Action
username	<username>	□	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	□	□	🗑
AuthPrefix	AR-JWT	□	□	🗑
TokenKey	access_token	□	□	🗑
ResponseType	TEXT	□	□	🗑

Figure 39 – Create Data Source – CMDB CI (Request Authentication Parameters for JWT)



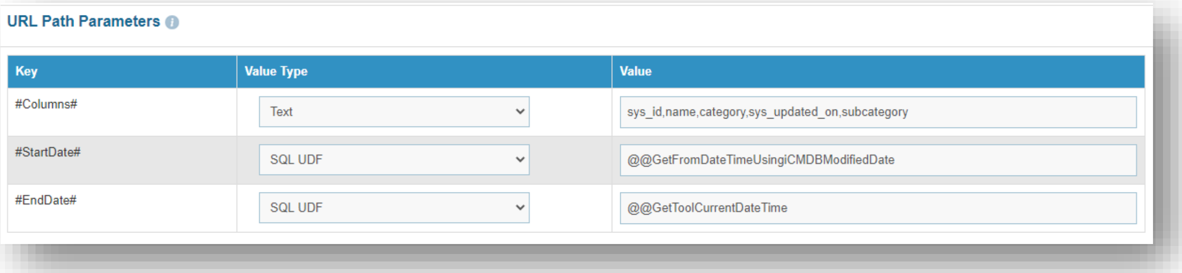
Request Authentication Parameters

Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	Bearer	☐	☐	🗑
client_id	<clientID>	☑	☑	🗑
client_secret	<clientsrcret>	☑	☑	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	JSON	☐	☐	🗑
grant_type	Password	☐	☑	🗑

Figure 40 – Create Data Source -CMDB CI (Request Authentication Parameters for OAuth2.0)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs.

```
Key: #Columns#

ValueType: Text

Value:

sys_id,name,category,sys_updated_on,subcategory


Note – These columns are mandatory. User can add more columns if
more data is required to be fetched from ITSM tool.


Key: #StartDate#

ValueType: SQL UDF

VALUE: @@GetFromDateTimeUsingiCMDBModifiedDate


Key: #EndDate#

ValueType: SQL UDF

VALUE: @@GetToolCurrentDateTime
```



URL Path Parameters

Key	Value Type	Value
#Columns#	Text	sys_id,name,category,sys_updated_on,subcategory
#StartDate#	SQL UDF	@@GetFromDateTimeUsingiCMDBModifiedDate
#EndDate#	SQL UDF	@@GetToolCurrentDateTime

Figure 41– URL Path Parameters – CMDB CI

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body -** Please enter the request body as per the configured URL, if applicable.
- **Response Body –** In this section, please enter the output of URL query for one of the incidents in JSON format. A sample response is mentioned below:

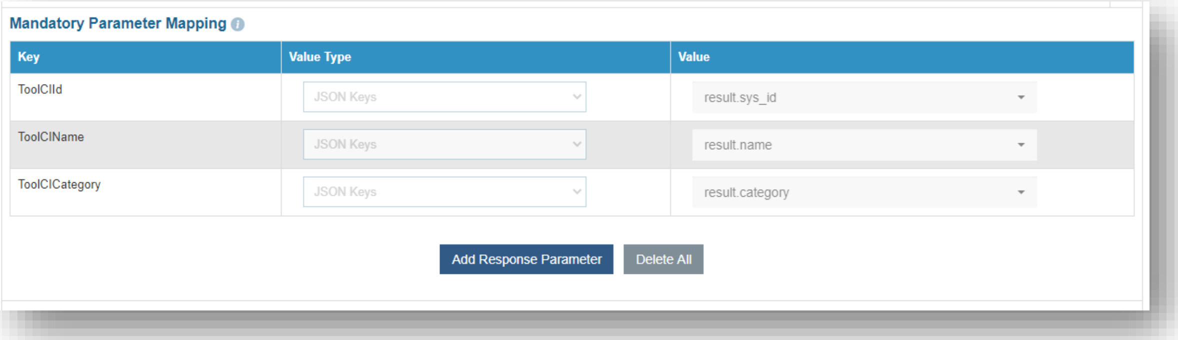
```
Response Body –
{
   "result": {
      "sys_id": "c8d2f53fdbcc1490e3bbde06f4961918",
      "name": "EC2AMAZ-FIHS9M1",
      "category": "Application",
      "subcategory": "Windows",
       "sys_updated_on":"2020-06-11 12:43:56"
   }
}
```

- After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section**.**
- **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below.

Table 9– Sample Mandatory Parameter Mapping – CMDB CI

Key	Value Type	Value

ToolCIId	JSON.Keys	result.sys_id
ToolCIName	JSON.Keys	result.name
ToolCICategory	JSON.Keys	result.category



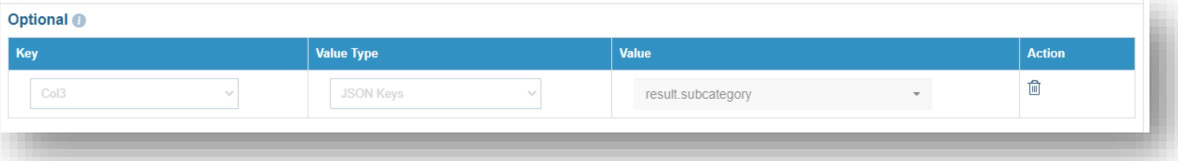
Key	Value Type	Value
ToolCIId	JSON Keys	result.sys_id
ToolCIName	JSON Keys	result.name
ToolCICategory	JSON Keys	result.category

Figure 42 – Mandatory Parameter Mapping – CMDB CI

– If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 10– Sample Optional Parameters – CMDB CI

Key	Value Type	Value
Col3	JSON.Keys	result. subcategory



Key	Value Type	Value	Action
Col3	JSON Keys	result.subcategory	🗑

Figure 43 – Optional Parameter Mapping – CMDB CI

– Click Next to proceed to Release Rules Configuration.

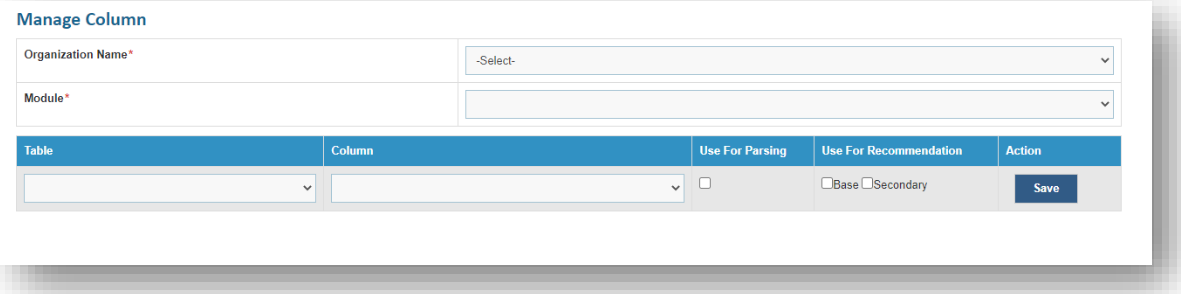
– Click **Submit** to add the data source.

4.2.1.3 Configuration of additional parameters for Recommendation and Parsing

To use the field values of CMDB CI for the purpose of Recommendation and Parsing by BigFix Runbook AI services, they need to be mapped to Incident Management.

To do so, perform the following steps -

- On the main menu bar, click Advance Configuration →Parameter → Manage Column.



Manage Column

Organization Name*	-Select-
Module*	

Table	Column	Use For Parsing	Use For Recommendation	Action
		☐	☐Base ☐Secondary	Save

Figure 44 – Map CMDB CI to Incident Management

- Select **Organization Name** from dropdown. Select I**ncident Management** as the **Module**.

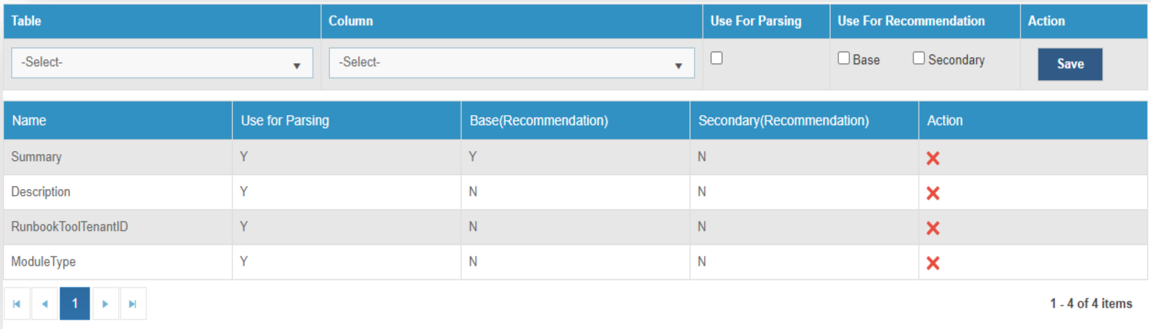


Table	Column	Use For Parsing	Use For Recommendation	Action
-Select-	-Select-	☐	☐ Base ☐ Secondary	Save

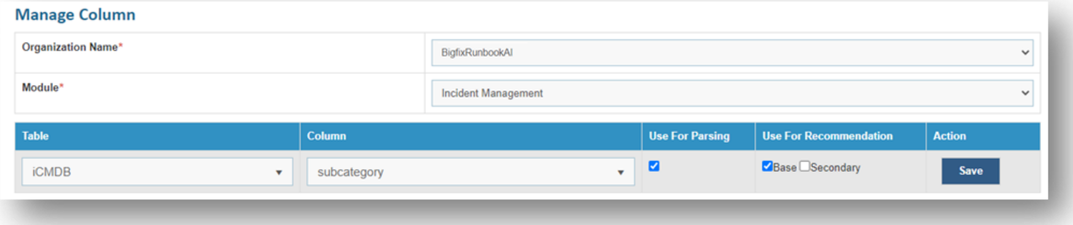
Name	Use for Parsing	Base(Recommendation)	Secondary(Recommendation)	Action
Summary	Y	Y	N	✗
Description	Y	N	N	✗
RunbookToolTenantID	Y	N	N	✗
ModuleType	Y	N	N	✗

1 - 4 of 4 items

Figure 45 – Map CMDB CI to Incident Management (cont.)

Summary, Description, RunbookToolTenantID, ModuleType are the default entries.

- Select **iCMDB** in Table dropdown.
- Select the column of CMDB which has to be mapped to incident in the **Column** dropdown. In this case, we are selecting **subcategory**.
- Check the fields Use For Parsing and 'Base' in Use For Recommendation.



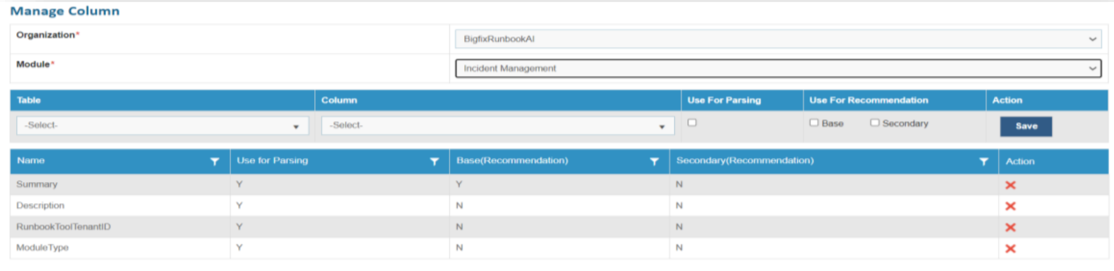
Manage Column

Organization Name*	BigfixRunbookAI
Module*	Incident Management

Table	Column	Use For Parsing	Use For Recommendation	Action
iCMDB	subcategory	☑	☑Base ☐Secondary	Save

Figure 46 – Map CMDB CI to Incident Management (cont.)

- Click **Save.** The page lists one additional entry i.e. '**Subcategory**', as depicted below:



Manage Column

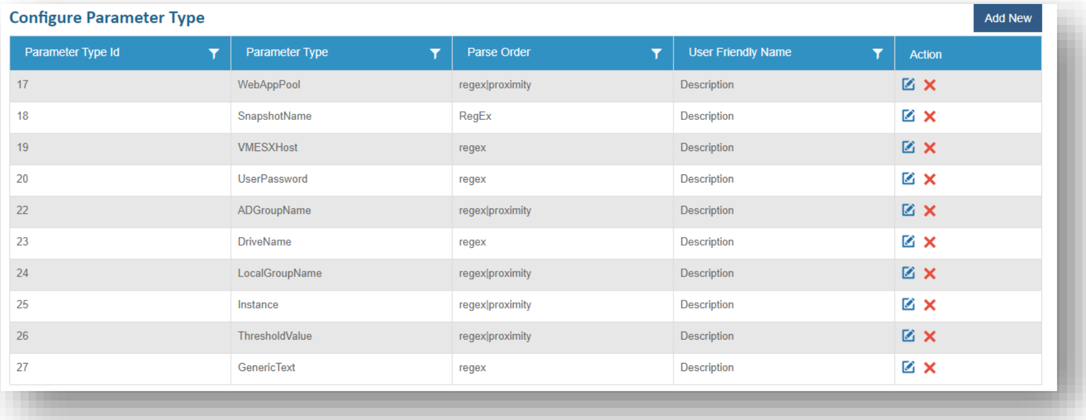
Organization*	BigfixRunbookAI
Module*	Incident Management

Table	Column	Use For Parsing	Use For Recommendation	Action
-Select-	-Select-	☐	☐ Base ☐ Secondary	Save

Name	Use for Parsing	Base(Recommendation)	Secondary(Recommendation)	Action
Summary	Y	Y	N	✕
Description	Y	N	N	✕
RunbookToolTenantID	Y	N	N	✕
ModuleType	Y	N	N	✕

Figure 47 – Map CMDB CI to Incident Management (cont.)

- For Recommendation, above steps are sufficient. But for Parsing, additional steps are required to be performed.
- On the main menu bar, click on **Advance configuration -> Parameter**.
- Click **Configure Parameter Type**. By default, there are several entries already defined.

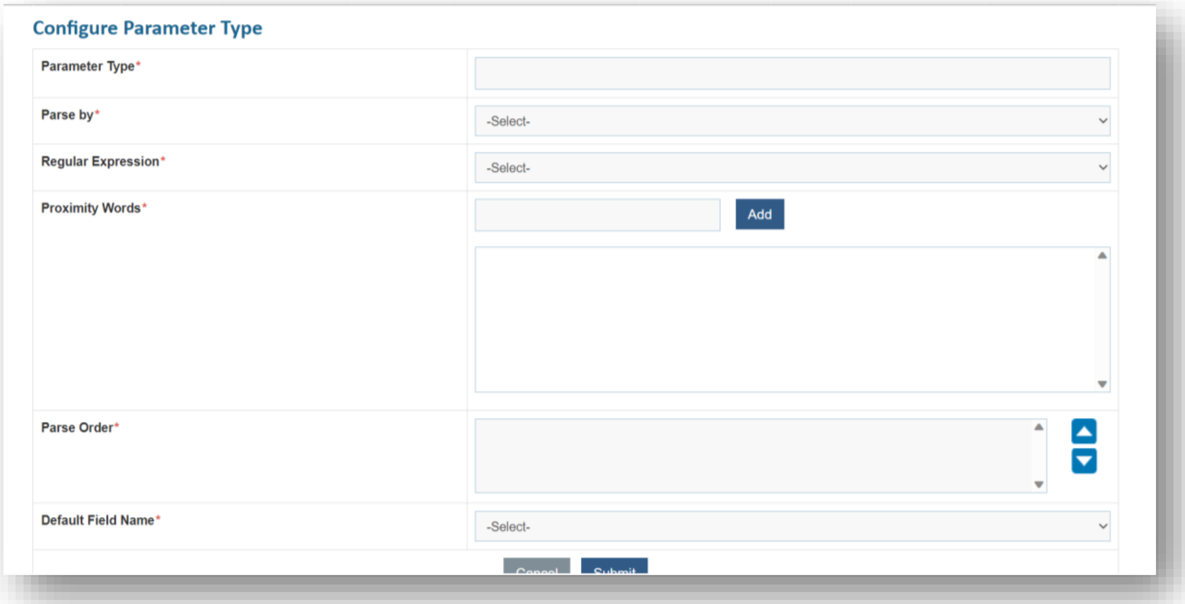


Configure Parameter Type

Parameter Type Id	Parameter Type	Parse Order	User Friendly Name	Action
17	WebAppPool	regex|proximity	Description	✎ ✕
18	SnapshotName	RegEx	Description	✎ ✕
19	VMESXHost	regex	Description	✎ ✕
20	UserPassword	regex	Description	✎ ✕
22	ADGroupName	regex|proximity	Description	✎ ✕
23	DriveName	regex	Description	✎ ✕
24	LocalGroupName	regex|proximity	Description	✎ ✕
25	Instance	regex|proximity	Description	✎ ✕
26	ThresholdValue	regex|proximity	Description	✎ ✕
27	GenericText	regex	Description	✎ ✕

Figure 48 – Map CMDB CI to Incident Management (cont.)

- Click **Add New**.

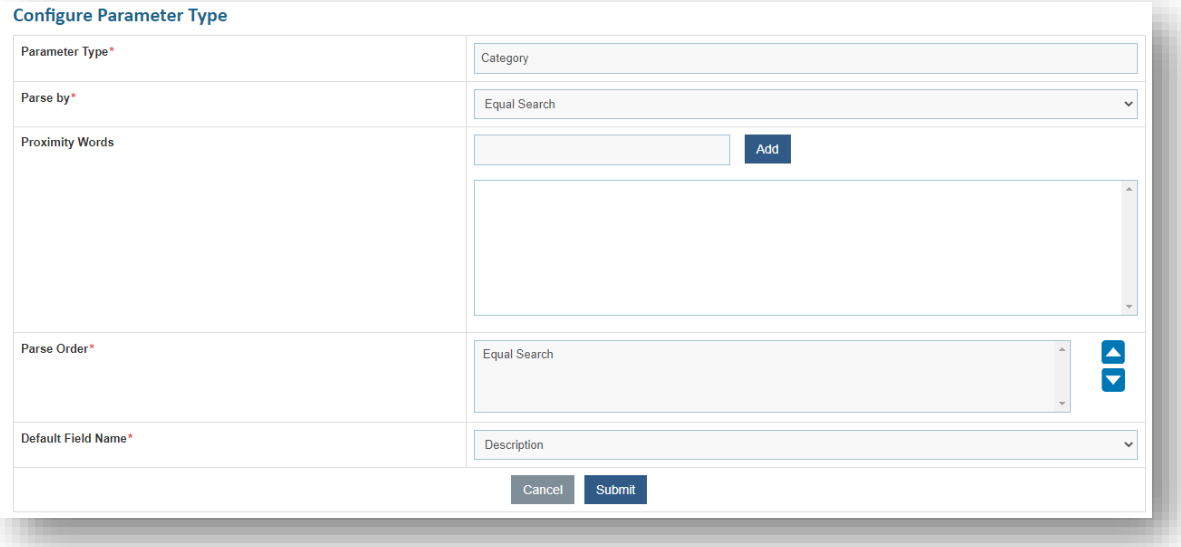


Configure Parameter Type

Parameter Type*	
Parse by*	-Select-
Regular Expression*	-Select-
Proximity Words*	
Parse Order*	
Default Field Name*	-Select-

Figure 49 – Map CMDB CI to Incident Management (cont.)

- Mention **Parameter Type**, for e.g. Category
- Select ‘Equal Search’ in the **Parse By** field.
- Select ‘Description’ in the **Default Field Name** field.
- Click **Submit**.

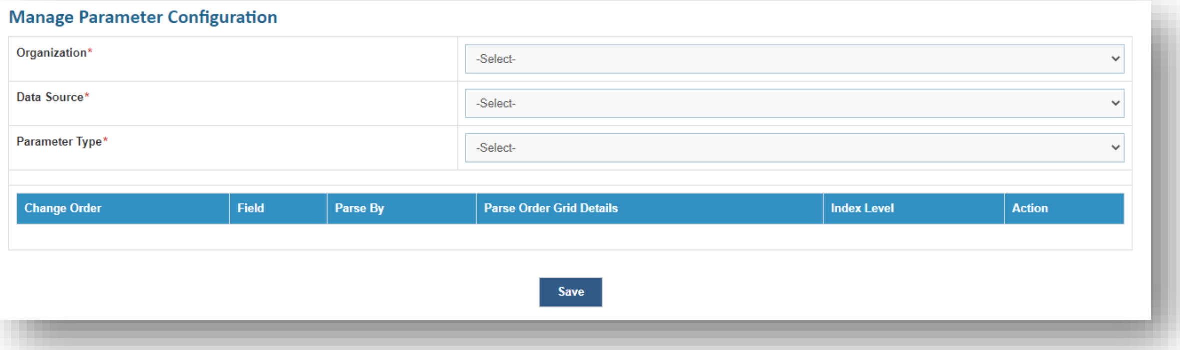


Configure Parameter Type

Parameter Type*	Category
Parse by*	Equal Search
Proximity Words	
Parse Order*	Equal Search
Default Field Name*	Description

Figure 50 – Map CMDB CI to Incident Management (cont.)

- Next step is to map this **Parameter Type** i.e. ‘**Category’**, to the one that was created via **Manage Columns** in earlier step by the name **subcategory**. To do that, perform the following steps:
- On the main menu bar, click Advance Configuration → Parameter.
- Click Manage Parameter Configuration.



Manage Parameter Configuration

Organization* -Select-

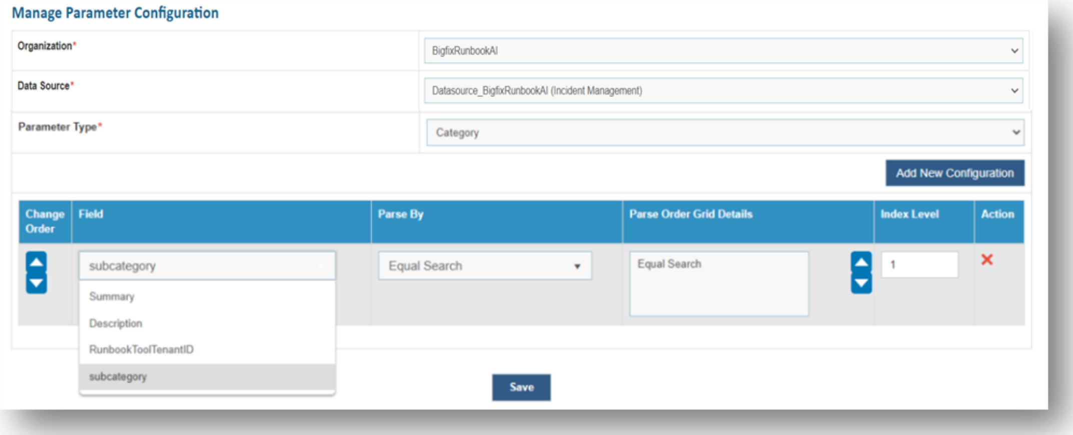
Data Source* -Select-

Parameter Type* -Select-

Change Order	Field	Parse By	Parse Order Grid Details	Index Level	Action

Figure 51 – Map CMDB CI to Incident Management (cont.)

- Select Organization.
- Select ‘Incident Management’ as the **Data Source**.
- Select the newly created parameter ‘Category’ from **Parameter Type** dropdown**.**
- From the **Field** dropdown, select ‘subcategory’**,** the parameter that has been mapped via **Manage Columns.**



Manage Parameter Configuration

Organization* BigfixRunbookAI

Data Source* Datasource_BigfixRunbookAI (Incident Management)

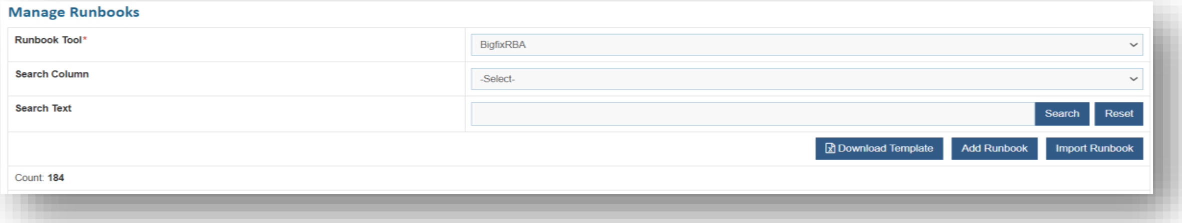
Parameter Type* Category

Change Order	Field	Parse By	Parse Order Grid Details	Index Level	Action
	subcategory	Equal Search	Equal Search	1	✗

Figure 52 – Map CMDB CI to Incident Management (cont.)

- Click **Save**.
- To verify whether this parameter is successfully parsed or not, perform the following steps -

- On the main menu bar, click **Runbooks.**
- Click Manage Runbooks.
- Select the **Runbook Tool** mapped with the organization.



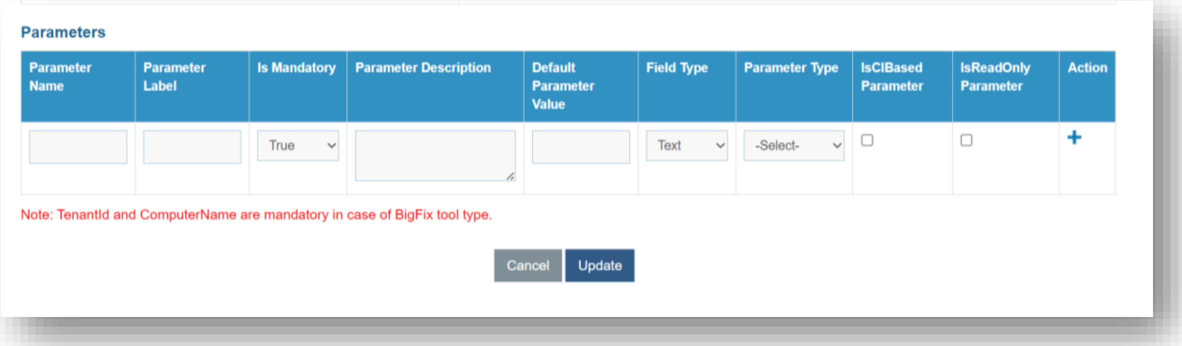
Manage Runbooks

Runbook Tool*	BigfixRBA
Search Column	-Select-
Search Text	

Count: 184

Figure 53 – Map CMDB CI to Incident Management (cont.)

- The parameter, **Category**, which was created in earlier steps, has to be added as one of the parameters to the existing runbook. You can also create a new runbook with **Category** as one of the parameters.
- Click the **Edit** icon to edit the runbook.
- In the Parameters section, add a new parameter with any relevant **Parameter Name**, **Parameter Label**, **Parameter Description**, **Default Parameter Value**. Ensure that Parameter Type is selected as **Category**.



Parameters

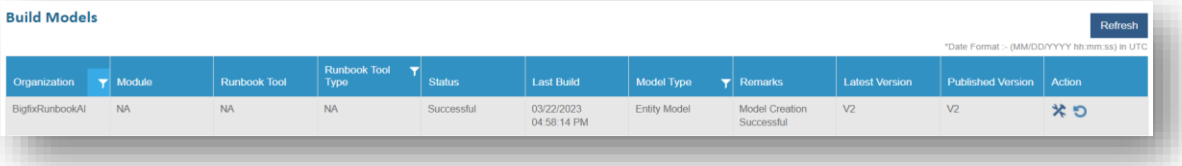
Parameter Name	Parameter Label	Is Mandatory	Parameter Description	Default Parameter Value	Field Type	Parameter Type	IsCIBased Parameter	IsReadOnly Parameter	Action
		True			Text	-Select-	□	□	+

Note: TenantId and ComputerName are mandatory in case of BigFix tool type.

Figure 54 – Map CMDB CI to Incident Management (cont.)

- Add the parameter and click **Update**.
- Ensure that the runbook in which the parameter is added is mapped with the organization.
- Next step is to build the Recommendation model and to do that perform the following steps:
- On the main menu bar, click **Actions -> Build Model**.

– ReBuild / Re-build the model for the Organization under **Incident Management** module for the mapped runbook tool.



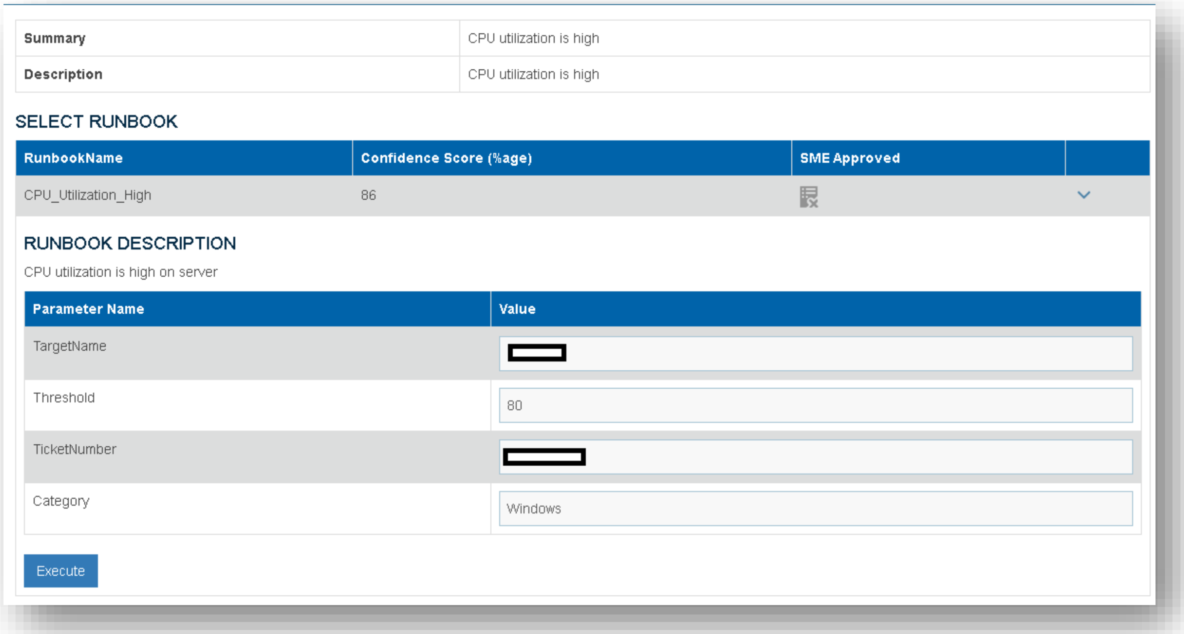
Build Models

*Date Format :- (MM/DD/YYYY hh:mm:ss) in UTC

Organization	Module	Runbook Tool	Runbook Tool Type	Status	Last Build	Model Type	Remarks	Latest Version	Published Version	Action
BigfixRunbookAI	NA	NA	NA	Successful	03/22/2023 04:58:14 PM	Entity Model	Model Creation Successful	V2	V2	⚒ ↻

Figure 55 – Map CMDB CI to Incident Management (cont.)

– Run the entire flow and see if the runbook recommended for the ticket in which the parameter was added has the parameter **Category** with its expected value.



Summary	CPU utilization is high
Description	CPU utilization is high

SELECT RUNBOOK

RunbookName	Confidence Score (%age)	SME Approved	
CPU_Utilization_High	86		⌄

RUNBOOK DESCRIPTION

CPU utilization is high on server

Parameter Name	Value
TargetName	[redacted]
Threshold	80
TicketNumber	[redacted]
Category	Windows

Figure 56 – Map CMDB CI to Incident Management (cont.)

4.2.2 Service Request Management

To fetch information about Service Requests, usually, creation of a data source for Service Request Tasks should suffice. However, there could be scenarios where some additional fields / values are required for processing the tickets – recommending the relevant runbooks and parsing the tickets to extract relevant parameters, for which separate data sources for Service Request and Service

Request Item must be created. Here, we will cover the procedure for creating all 3 kinds of data sources.

4.2.2.1 Create Data Source for Service Request

To create a data source for Service Requests, perform the following steps:

- On the main menu bar, click **Action Tab → Manage Data Sources**.
- The **Create Data Source** page appears with the following tabs:
  - Organization
  - Data Source
  - Fetch Data Configuration

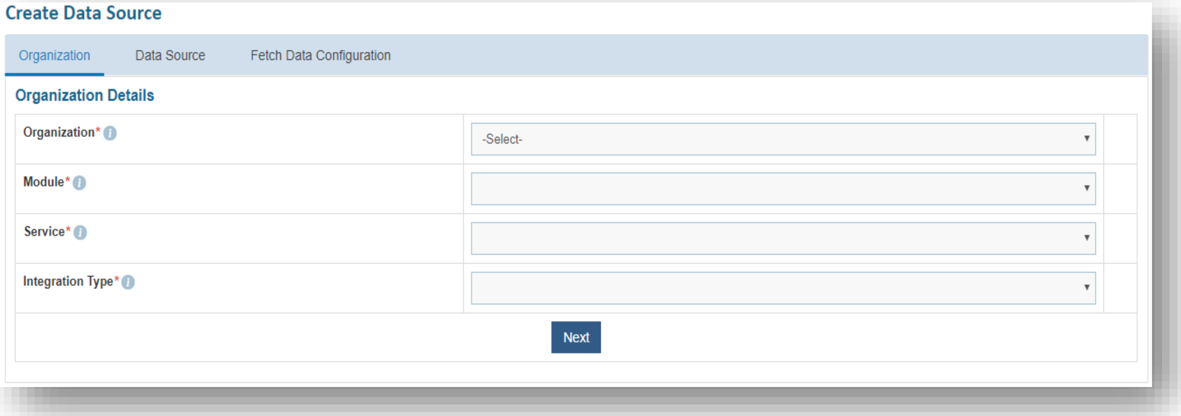
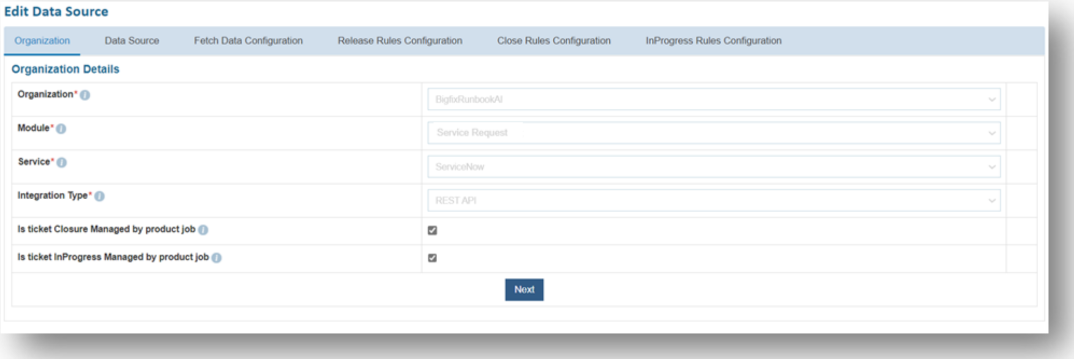


Figure 57 - Create Data Source – Service Request

Release Rules Configuration is only applicable for the following **Module** types- **Incident Management, Change Request Task and Service Request Task.** This tab will not be activated for other module types.

- On the **Organization** tab:
  - Select the **Organization Name** from the dropdown.
  - In the **Module** field, select ‘Service Request’, since we are using this data source for using its field value for the **Service Request Tasks**.
  - In the **Service** field, select **Service Now Tool** as we are configuring the data source for ServiceNow.
  - In the **Integration Type** field, select **REST API**, since we will be integrating through REST APIs.
  - Click **Next**.



Edit Data Source

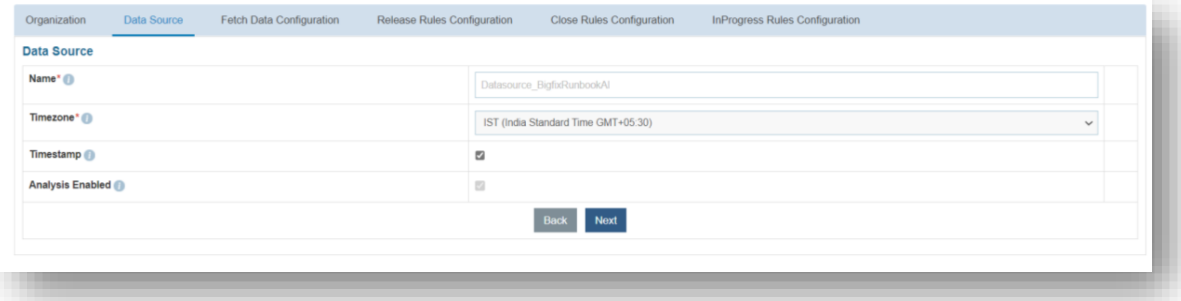
Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

Organization Details

Organization*	BigfixRunbookAI
Module*	Service Request
Service*	ServiceNow
Integration Type*	REST API
Is ticket Closure Managed by product job	☑
Is ticket InProgress Managed by product job	☑

Figure 58 - Create Data Source – Service Request (cont.)

- On the **Data Source** tab,
  - Type the new data source in the **Name** field.
  - Select the **Timezone** to specify the time zone of the selected data source.
  - Select **Is Datetime** to view the present data with date and time.
  - Select **Analysis Enabled** if user wants to analyze the data retrieved from the data source.
  - Click **Next**.



Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

Data Source

Name*	Datasource_BigfixRunbookAI
Timezone*	IST (India Standard Time GMT+05:30)
Timestamp	☑
Analysis Enabled	☑

Figure 59 - Create Data Source – Service Request (cont.)

- On the **Fetch Data Configuration** tab, populate the details as per the environment.
- In the **Connection Details** section, enter the following details:
  - **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.
  - **Sample URL** - https://URL.service-now.com/api/now/v1/table/sc_request?sysparm_fields=#Columns#&sysparm_query=sys_updated_on>=#StartDate#^sys_updated_on<=#EndDate#^ORDERBYsys_updated_on

- **Authentication Type** – Select one of the Authentication Types from Basic / Windows, JWT, OAuth 2.0

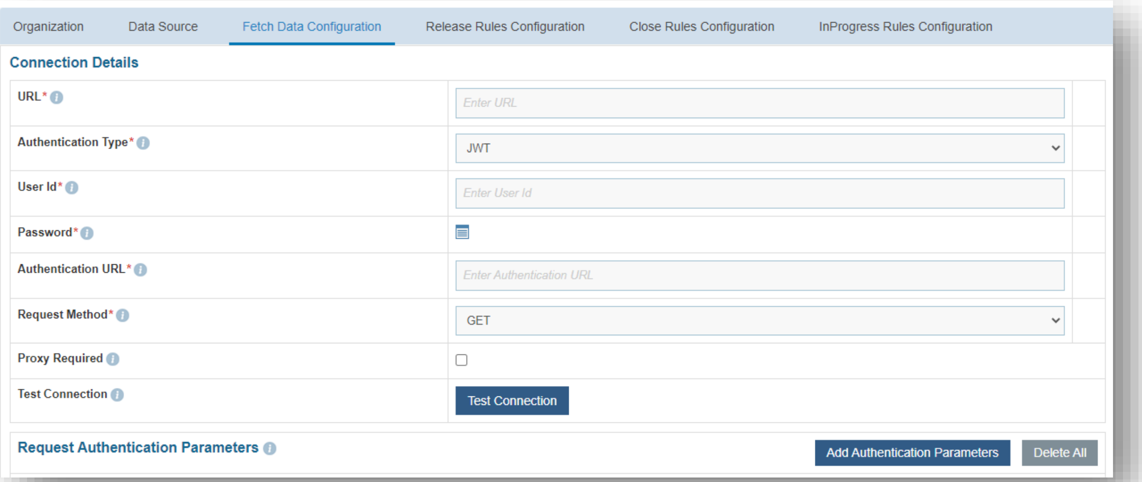
  Selection of **Basic / Windows** requires you to enter -

  - User Id
  - Password

  Selection of **JWT / OAuth 2.0** requires you to enter -

  - User Id
  - Password
  - Authentication URL

- **Request Body –** Select the **GET**, **POST** or **PUT** as Request Method as per the configured URL.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.


Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

Connection Details

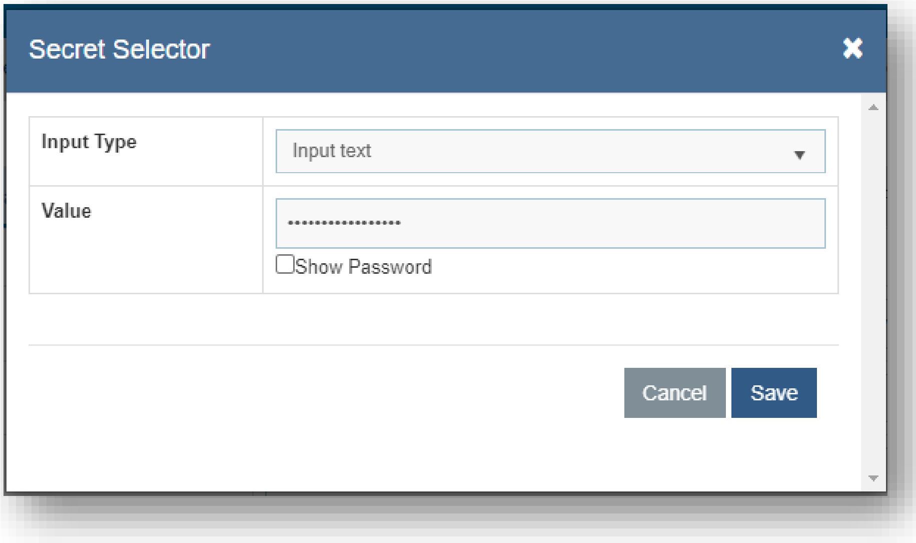
URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	GET
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

Add Authentication Parameters
Delete All

Figure 60 – Create Data Source – Service Request (Connection Details)

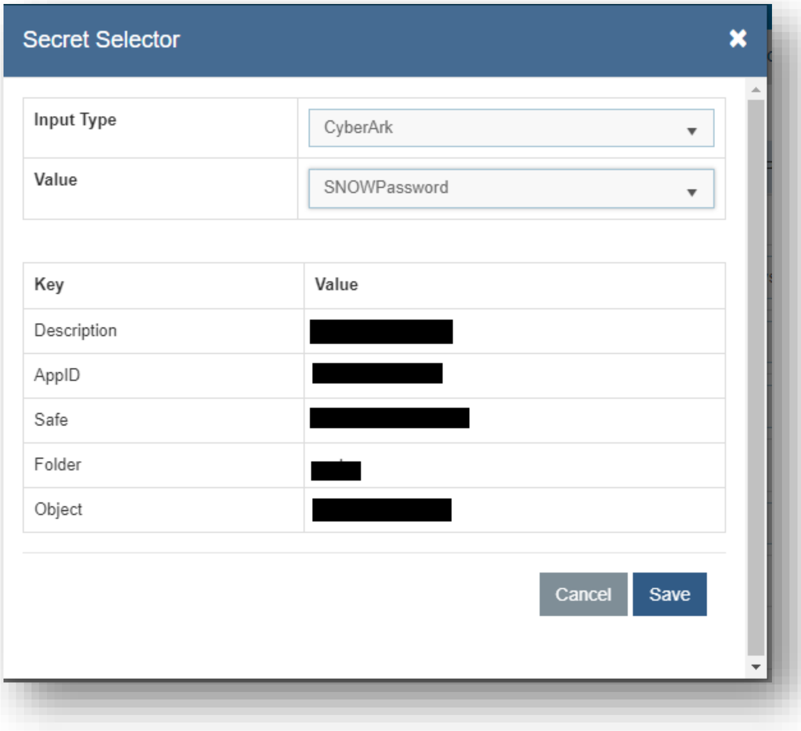
- **Password** – For password, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Secret Selector

Input Type	Input text
Value	•••••••••••••••••• □ Show Password

Figure 61 – Password in plaintext



Secret Selector

Input Type	CyberArk
Value	SNOWPassword

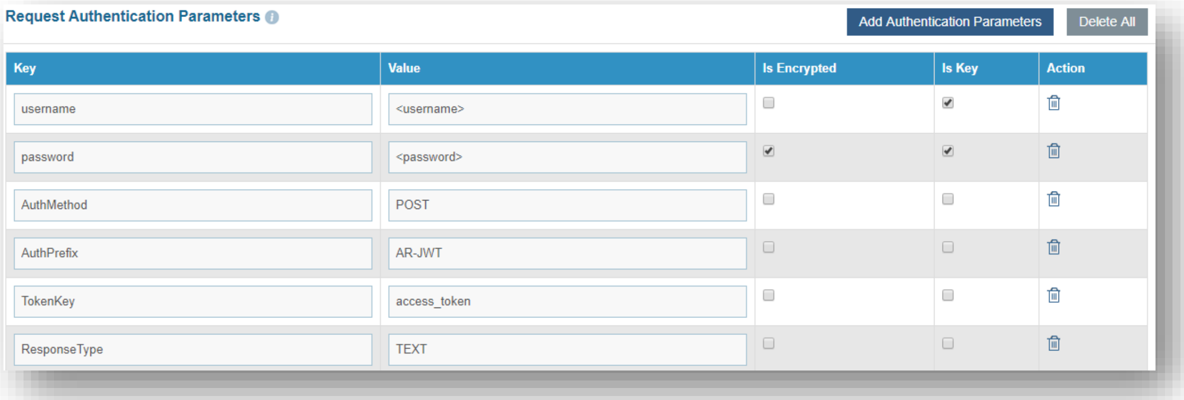
Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 62 – Password from Key Vault (CyberArk)

- **Request Authentication Parameters** – If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.
- Based on the **Authentication Type**, add the parameters mentioned in the below table:

Table 11 – Sample Authentication Parameters – Service Request

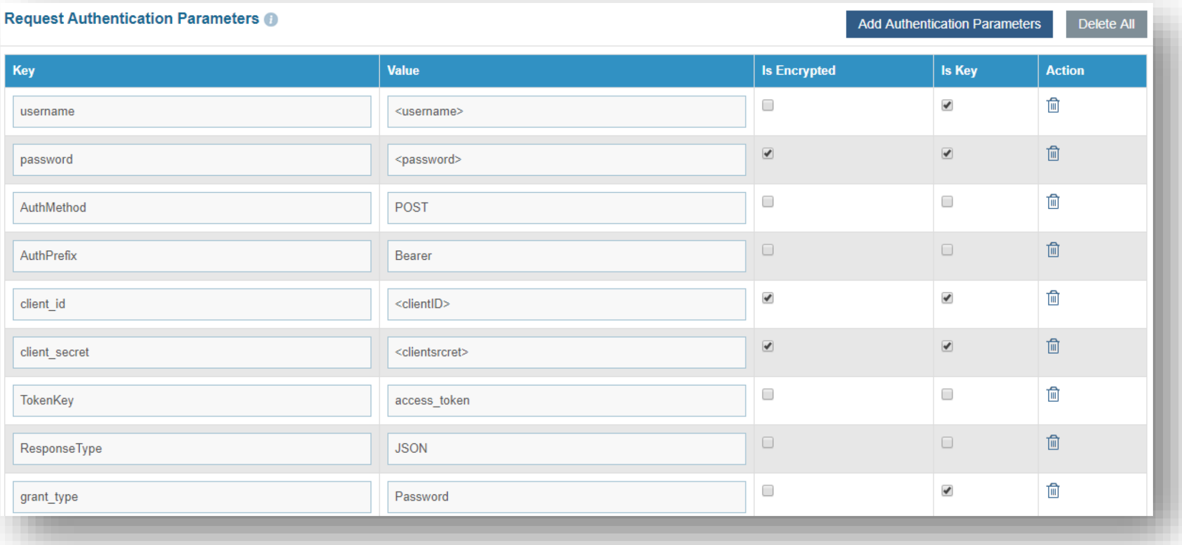
Authentication Type	Key	Value	Is Encrypted?	Is Key?
JWT	username	<username>	NO	YES
JWT	password	<password>	YES	YES
JWT	AuthMethod	POST	NO	NO
JWT	AuthPrefix	AR-JWT	NO	NO
JWT	TokenKey	access_token	NO	NO
JWT	ResponseType	TEXT	NO	NO
OAuth2.0	username	<username	NO	YES
OAuth2.0	password	<password>	YES	YES
OAuth2.0	AuthMethod	POST	NO	NO
OAuth2.0	AuthPrefix	Bearer	NO	NO
OAuth2.0	client_id	<clientID>	YES	YES
OAuth2.0	client_secret	<clientsecret>	YES	YES
OAuth2.0	TokenKey	access_token	NO	NO
OAuth2.0	ResponseType	JSON	NO	NO
OAuth2.0	grant_type	Password	NO	YES



Request Authentication Parameters

Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	AR-JWT	☐	☐	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	TEXT	☐	☐	🗑

Figure 63 – Create Data Source – Service Request (Request Authentication Parameters for JWT)



Request Authentication Parameters

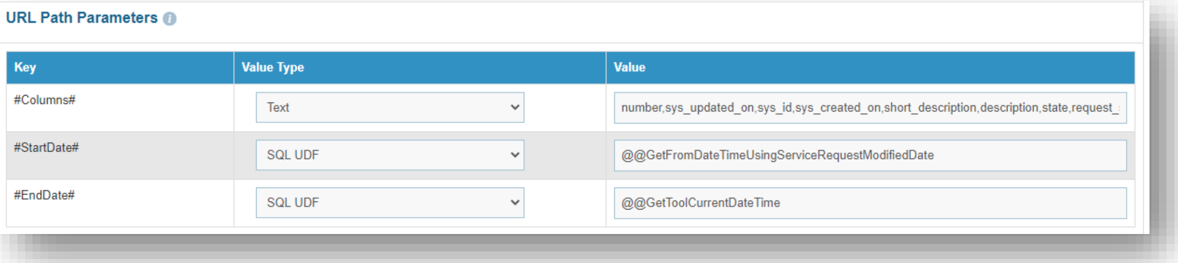
Key	Value	Is Encrypted	Is Key	Action
username	<username>	□	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	□	□	🗑
AuthPrefix	Bearer	□	□	🗑
client_id	<clientID>	☑	☑	🗑
client_secret	<clientsrcret>	☑	☑	🗑
TokenKey	access_token	□	□	🗑
ResponseType	JSON	□	□	🗑
grant_type	Password	□	☑	🗑

Figure 64 – Create Data Source – Service Request (Request Authentication Parameters for OAuth2.0)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #Columns#

ValueType: Text

Value:

number,sys_updated_on,sys_id,sys_created_on,short_description,description,state,request_state


Note – These columns are mandatory. User can add more columns if more data is required to be fetched from ITSM tool.


Key: #StartDate#

ValueType: SQL UDF

VALUE: @@GetFromDateTimeUsingServiceRequestModifiedDate


Key: #EndDate#
```

```
ValueType: SQL UDF

VALUE: @@GetToolCurrentDateTime
```



URL Path Parameters

Key	Value Type	Value
#Columns#	Text	number,sys_updated_on,sys_id,sys_created_on,short_description,description,state,request_
#StartDate#	SQL UDF	@@GetFromDateTimeUsingServiceRequestModifiedDate
#EndDate#	SQL UDF	@@GetToolCurrentDateTime

Figure 65 – URL Path Parameters – Service Request (Service Request Task Management)

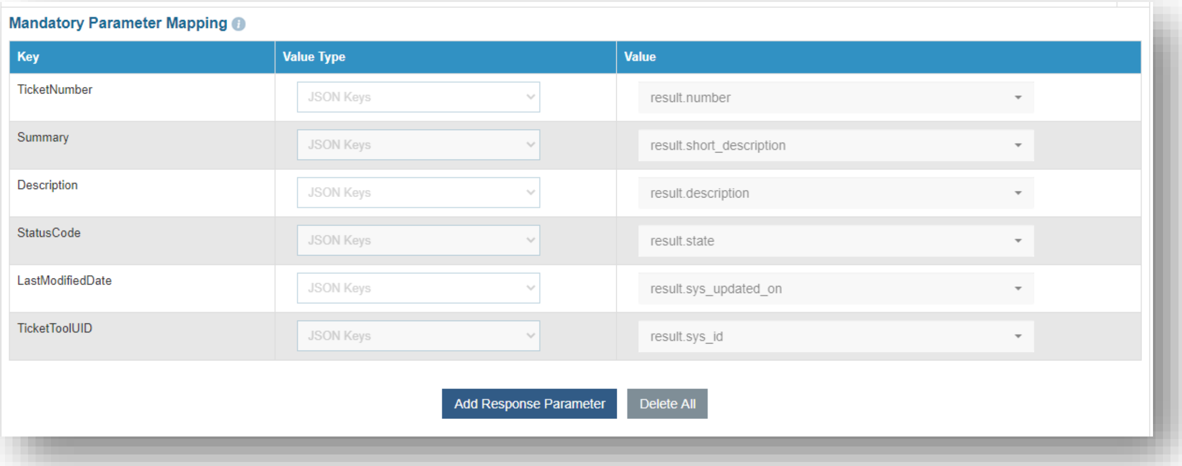
- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body -** Enter the request body in JSON format as per the configured URL, if applicable.
- **Response Body –** In this section, please enter the output of URL query for one of the service request tasks in JSON format. A sample response is mentioned below.

```
Response Body –

{

   "result": {

      "number": "REQ0011787",

      "sys_id": "2ae764d5db199c14e3bbde06f496195a",

      "short_description": "Test",

      "request_state": "in_process",

      "sys_created_on": "2020-06-08 10:34:54",

      "description": "test",

      "sys_updated_on": "2020-06-08 10:34:56",

      "state": "2"

   }

}
```

- After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section**.**
- **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below.

Table 12– Sample Mandatory Mapping Parameters – Service Request

Key	Value Type	Value
TicketNumber	JSON.Keys	result.number
Summary	JSON.Keys	result.short_description
Description	JSON.Keys	result.description
StatusCode	JSON.Keys	result.state
LastModifiedDate	JSON.Keys	result.sys_updated_on
TicketToolUID	JSON.Keys	result.sys_id



Mandatory Parameter Mapping

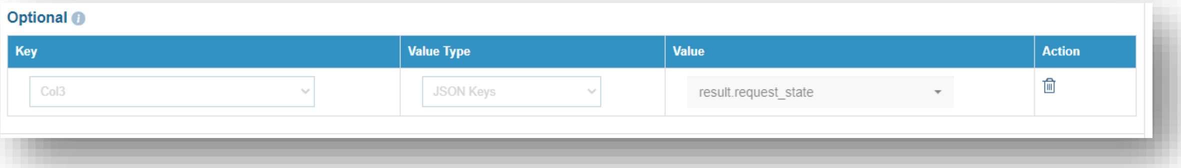
Key	Value Type	Value
TicketNumber	JSON Keys	result.number
Summary	JSON Keys	result.short_description
Description	JSON Keys	result.description
StatusCode	JSON Keys	result.state
LastModifiedDate	JSON Keys	result.sys_updated_on
TicketToolUID	JSON Keys	result.sys_id

Figure 66 – Mandatory Parameter Mapping (Service Request Management)

- If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 13– Sample Optional Mapping Parameters – Service Request

Key	Value Type	Value
Col3	JSON.Keys	result.request_state



Optional

Key	Value Type	Value	Action
Col3	JSON Keys	result.request_state	🗑

Figure 67 – Optional Parameter Mapping (Service Request Management)

– Click **Submit** to add the data source.

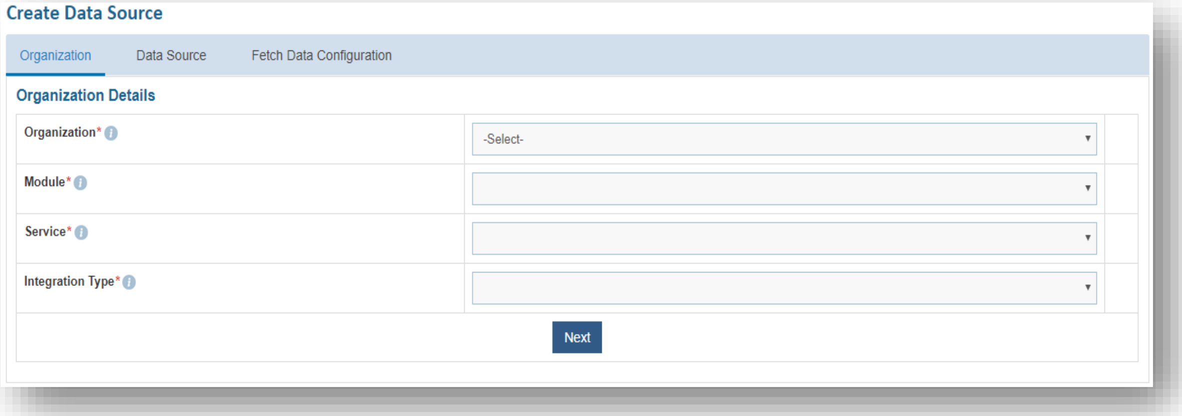
4.2.2.2 Create Data Source for Service Request Tasks

To create a data source for Service Requests Tasks Management, perform the following steps:

– On the main menu bar, click Actions Tab → Manage Data Sources.

– The **Create Data Source** page appears with the following tabs:

- Organization
- Data Source
- Fetch Data Configuration
- Release Rules Configuration



Create Data Source

Organization Details

Organization*	-Select-
Module*	
Service*	
Integration Type*	

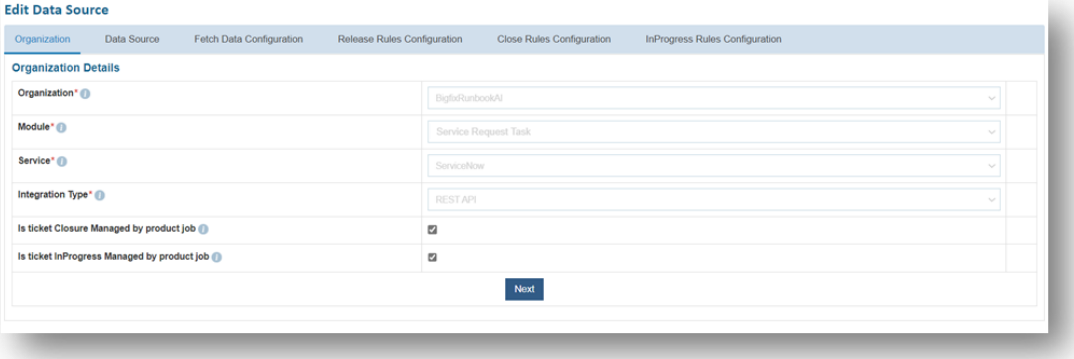
Figure 68 - Create Data Source – Service Request Tasks

Release Rules Configuration is only applicable for the following **Module** types- **Incident Management, Change Request Task and Service Request Task.** This tab will not be activated for other module types.

– On the **Organization** tab,

- Select the **Organization Name** from the dropdown.

- In the **Module** field, select 'Service Request Task'**,** since we are configuring this data source for pulling the service requests tasks.
- In the **Service** field, select **Service Now Tool** as we are configuring the data source for ServiceNow
- In the **Integration Type** field, select **REST**, since we will be integrating through REST APIs.
- Click **Next**.



Edit Data Source

Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

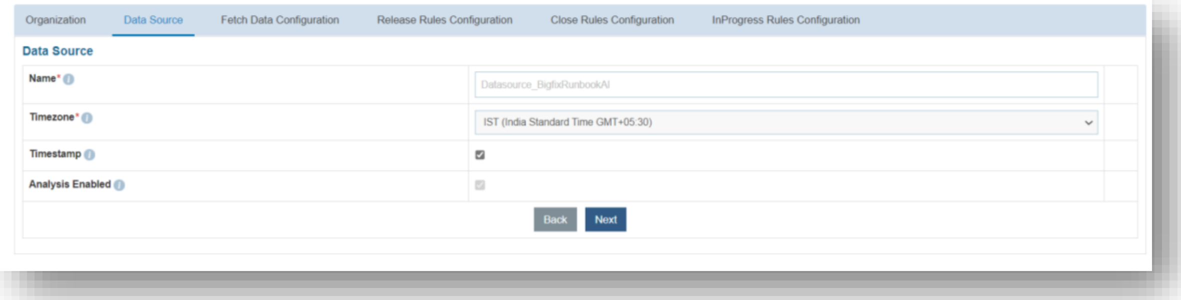
Organization Details

Organization*	BigfixRunbookAI
Module*	Service Request Task
Service*	ServiceNow
Integration Type*	REST API
Is ticket Closure Managed by product job	☑
Is ticket InProgress Managed by product job	☑

Figure 69 - Create Data Source – Service Request Tasks (cont.)

– On the **Data Source** tab,

- Type the new data source in the **Name** field.
- Select the **Timezone** to specify the time zone of the selected data source.
- Select **Timestamp** to view the present data with date and time.
- Select **Analysis Enabled,** if user wants to analyze the data retrieved from the data source.
- Click **Next**.



Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

Data Source

Name*	Datasource_BigfixRunbookAI
Timezone*	IST (India Standard Time GMT+05:30)
Timestamp	☑
Analysis Enabled	☑

Figure 70 - Create Data Source – Service Request Tasks (cont.)

– On the **Fetch Data Configuration** tab, type in the details as per the environment.

– In the **Connection Details** section, enter the following details:

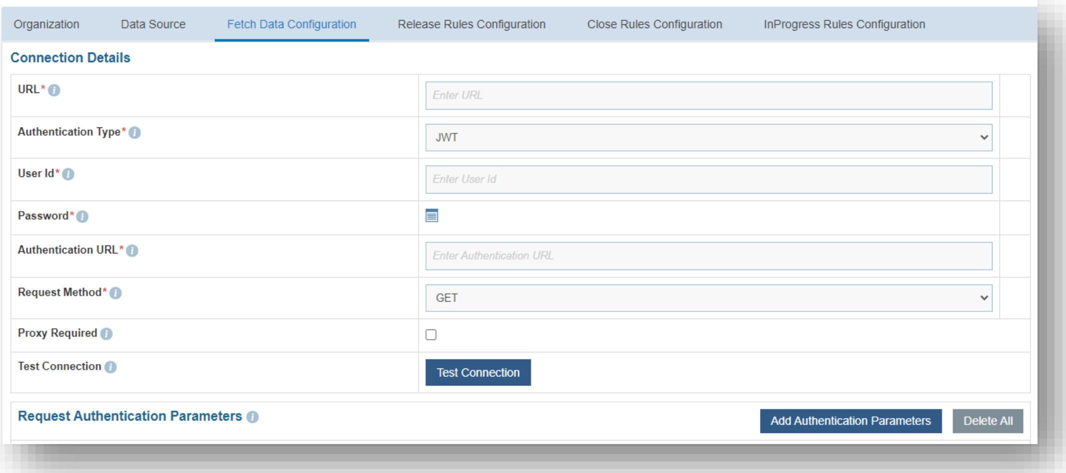
- **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.
- **Sample URL** - https://URL.service-now.com/api/now/v1/table/sc_task?sysparm_fields=#Columns#&sysparm_query=active=true^sys_updated_on>=#StartDate#^sys_updated_on<=#EndDate#^ORDERBYsys_updated_on
- **Authentication Type** – Select one of the Authentication Types from Basic / Windows, JWT, OAuth 2.0

  Selection of **Basic / Windows** requires you to enter -

  - User Id
  - Password

  Selection of **JWT / OAuth 2.0** requires you to enter -

  - User Id
  - Password
  - Authentication URL
- **Request Method -** Enter the request method as **GET**, **POST** or **PUT** as per the configured URL.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



Organization | Data Source | Fetch Data Configuration | Release Rules Configuration | Close Rules Configuration | InProgress Rules Configuration

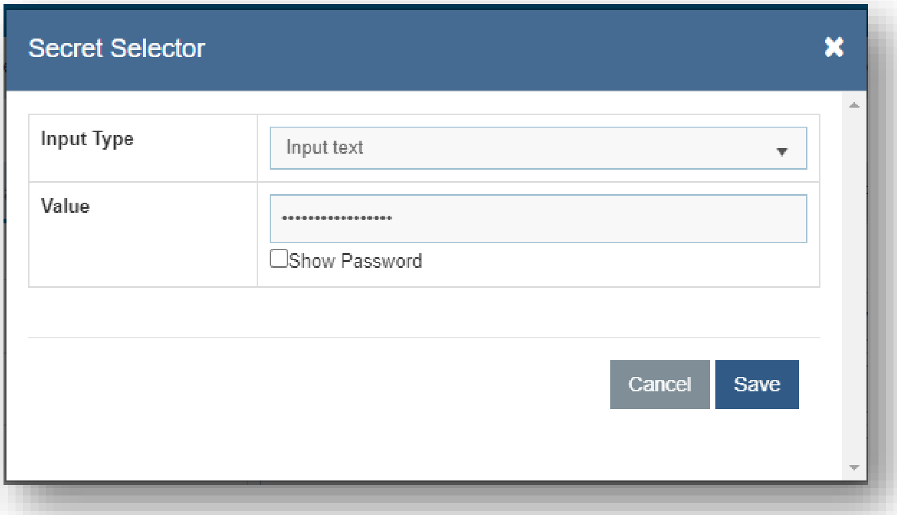
Connection Details

URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	GET
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters — Add Authentication Parameters | Delete All

Figure 71 – Create Data Source – Service Request Tasks (Connection Details)

- **Password** – For password, click on icon next to it. If the password is available in plaintext then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.

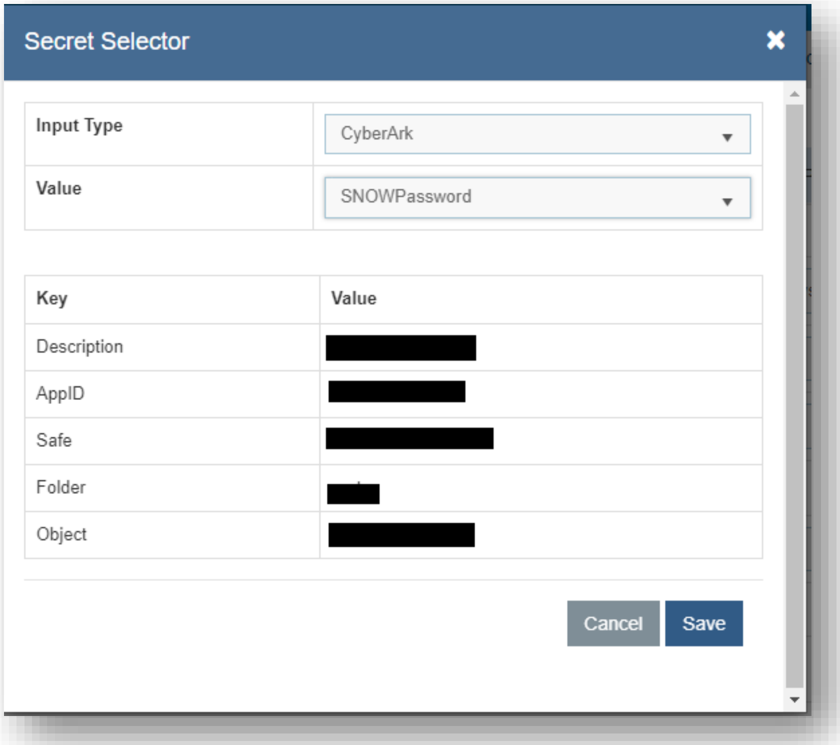


Secret Selector

Input Type	Input text
Value	••••••••••••••••• □ Show Password

Cancel | Save

Figure 72 – Password in plaintext



Input Type	CyberArk
Value	SNOWPassword

Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

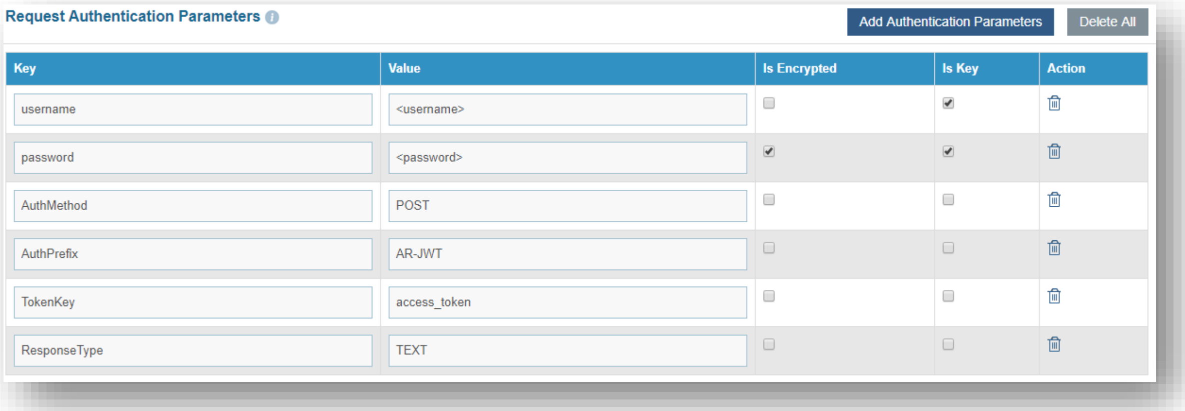
Figure 73 – Password from Key Vault (CyberArk)

- **Request Authentication Parameters** – If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.
- Based on the **Authentication Type**, add the parameters mentioned in the below table:

Table 14 – Sample Authentication Parameters – Service Request Tasks

Authentication Type	Key	Value	Is Encrypted?	Is Key?
JWT	username	<username>	NO	YES
JWT	password	<password>	YES	YES
JWT	AuthMethod	POST	NO	NO
JWT	AuthPrefix	AR-JWT	NO	NO
JWT	TokenKey	access_token	NO	NO
JWT	ResponseType	TEXT	NO	NO
OAuth2.0	username	<username	NO	YES
OAuth2.0	password	<password>	YES	YES
OAuth2.0	AuthMethod	POST	NO	NO
OAuth2.0	AuthPrefix	Bearer	NO	NO
OAuth2.0	client_id	<clientID>	YES	YES
OAuth2.0	client_secret	<clientsecret>	YES	YES

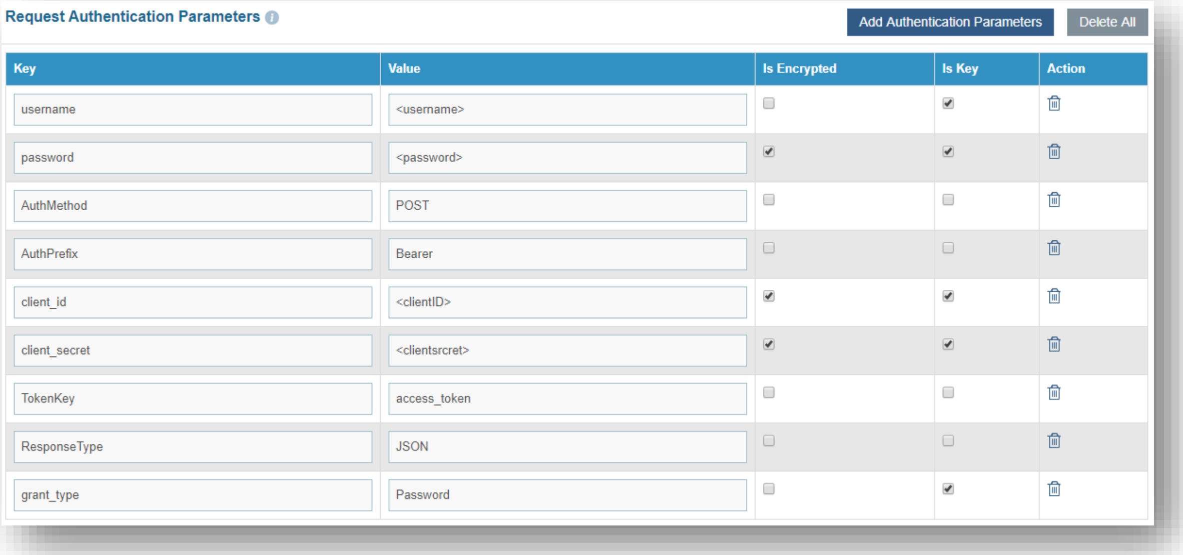
OAuth2.0	TokenKey	access_token	NO	NO
OAuth2.0	ResponseType	JSON	NO	NO
OAuth2.0	grant_type	Password	NO	YES



Request Authentication Parameters

Key	Value	Is Encrypted	Is Key	Action
username	<username>	□	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	□	□	🗑
AuthPrefix	AR-JWT	□	□	🗑
TokenKey	access_token	□	□	🗑
ResponseType	TEXT	□	□	🗑

Figure 74 – Create Data Source – Service Request Tasks (Request Authentication Parameters for JWT)



Request Authentication Parameters

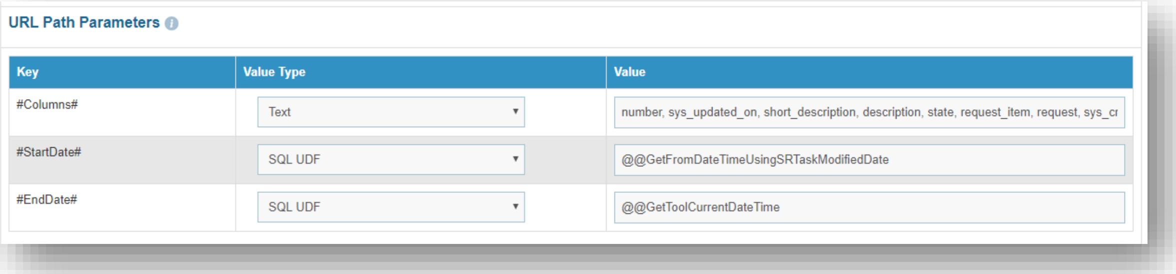
Key	Value	Is Encrypted	Is Key	Action
username	<username>	□	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	□	□	🗑
AuthPrefix	Bearer	□	□	🗑
client_id	<clientID>	☑	☑	🗑
client_secret	<clientsrcret>	☑	☑	🗑
TokenKey	access_token	□	□	🗑
ResponseType	JSON	□	□	🗑
grant_type	Password	□	☑	🗑

Figure 75 – Request Authentication Parameters for OAuth2.0

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #Columns#

ValueType: Text

Value:
```

```
number, sys_updated_on, short_description, description,
assignment_group,closed_at,category,dv_assigned_to,sys_id,sys_crea
ted_on,state,request,request_item,sys_id


Note – These columns are mandatory. User can add more columns if
more data is required to be fetched from ITSM tool.


Key: #StartDate#

ValueType: SQL UDF

VALUE: @@GetFromDateTimeUsingSRTaskModifiedDate


Key: #EndDate#

ValueType: SQL UDF

VALUE: @@GetToolCurrentDateTime
```



URL Path Parameters

Key	Value Type	Value
#Columns#	Text	number, sys_updated_on, short_description, description, state, request_item, request, sys_cr
#StartDate#	SQL UDF	@@GetFromDateTimeUsingSRTaskModifiedDate
#EndDate#	SQL UDF	@@GetToolCurrentDateTime

Figure 76 – URL Path Parameters (Service Request Task)

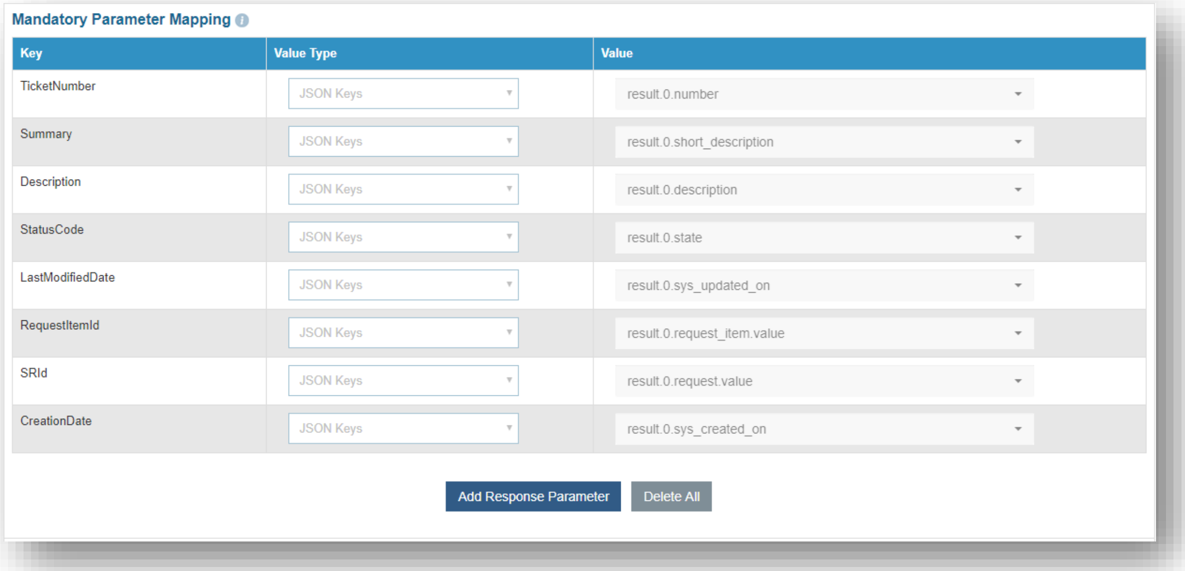
- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body -** Enter the request body in JSON format as per the configured URL, if applicable.
- **Response Body –** In this section, please enter the output of URL query for one of the service request tasks in JSON format. A sample response is mentioned below.

```
Response Body –

{
   "result": [{"number": "TASK2190188","short_description": "For
fullfillment","description": "Test",        "state": "1","active":
"true","sys_created_on":"2019-12-31 05:45:39","sys_id":
"0701e746db9a0450b773f3731d9619ab","approval": "not
requested","sys_updated_on":"2020-01-31 05:45:39”,"request": {

"link":"https://hclmtdev.servicenow.com/api/now/v1/table/sc_reques
t/be702706db9a0450b773f3731d961907",  "value":
"be702706db9a0450b773f3731d961907"},

 "request_item": {"link": "https://hclmtdev.service-
now.com/api/now/v1/table/sc_req_item/32702706db9a0450b773f3731d961
908", "value": "32702706db9a0450b773f3731d961908"

      }

   }]

}
```

- After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section**.**
- **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below.

Table 15– Sample Mandatory Mapping Parameters – Service Request Tasks

Key	Value Type	Value
TicketNumber	JSON.Keys	result.0.number
Summary	JSON.Keys	result.0.short_description
Description	JSON.Keys	result.0.description
StatusCode	JSON.Keys	result.0.state
LastModifiedDate	JSON.Keys	result.0.sys_updated_on
RequestItemId	JSON.Keys	result.0.request_item.value
SRId	JSON.Keys	result.0.request.value
CreationDate	JSON.Keys	result.0.sys_created_on



Mandatory Parameter Mapping

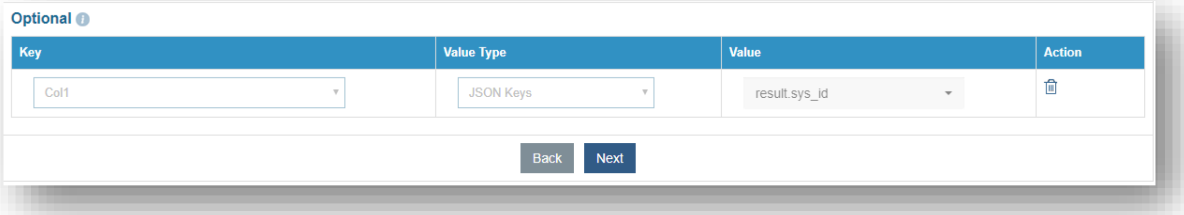
Key	Value Type	Value
TicketNumber	JSON Keys	result.0.number
Summary	JSON Keys	result.0.short_description
Description	JSON Keys	result.0.description
StatusCode	JSON Keys	result.0.state
LastModifiedDate	JSON Keys	result.0.sys_updated_on
RequestItemId	JSON Keys	result.0.request_item.value
SRId	JSON Keys	result.0.request.value
CreationDate	JSON Keys	result.0.sys_created_on

Figure 77 – Mandatory Parameter Mapping (Service Request Task)

– If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 16– Sample Optional Mapping Parameters – Service Request Tasks

Key	Value Type	Value
Col1	JSON.Keys	result.sys_id



Optional

Key	Value Type	Value	Action
Col1	JSON Keys	result.sys_id	🗑

Figure 78 – Optional Parameter Mapping (Service Request Task)

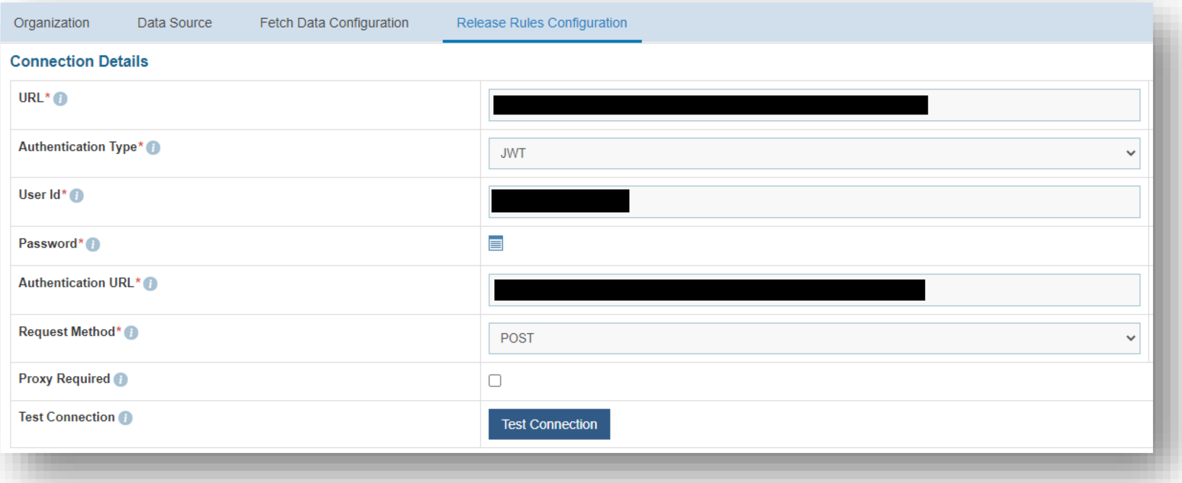
– Click Next to proceed to Release Rules Configuration.

– On **Release Rules Configuration** tab, type in the details as per the requirement.

– In the **Connection Details** section, enter the following details:

- **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.

- **Sample URL** - https://<url>.service-now.com/api/now/table/sc_task/#incident#
- **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously.
- Request Method – Select Request Method as PUT from the drop-down.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.

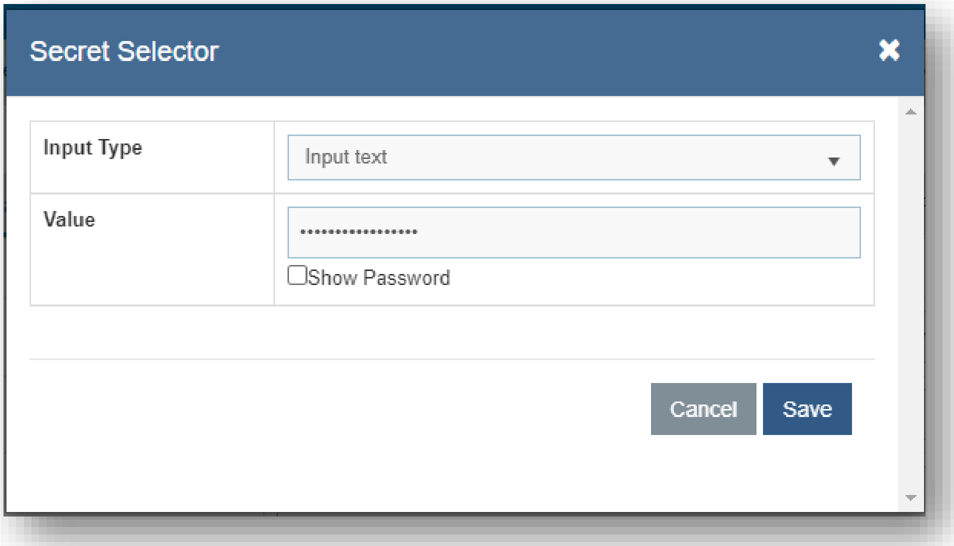


Connection Details

URL*	[redacted]
Authentication Type*	JWT
User Id*	[redacted]
Password*	
Authentication URL*	[redacted]
Request Method*	POST
Proxy Required	☐
Test Connection	Test Connection

Figure 79 – Release Rules Configuration – Service Request Tasks (Connection Details)

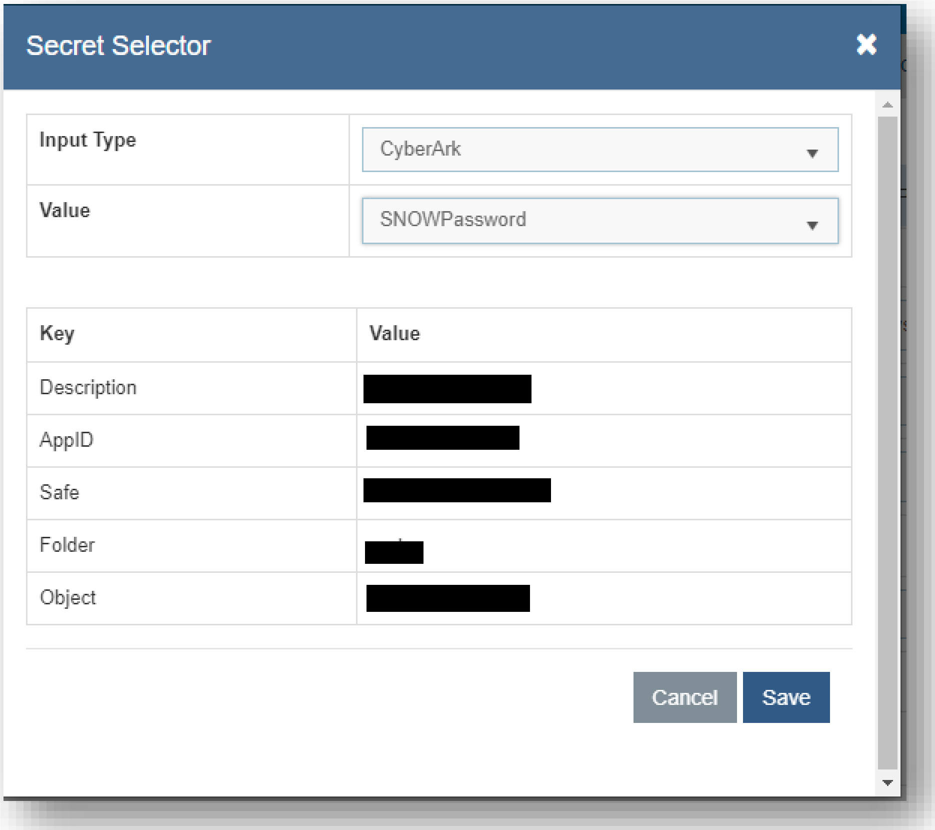
- **Password** – For password, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Secret Selector

Input Type	Input text
Value	••••••••••••••••• ☐ Show Password

Figure 80 – Password in plaintext



Secret Selector

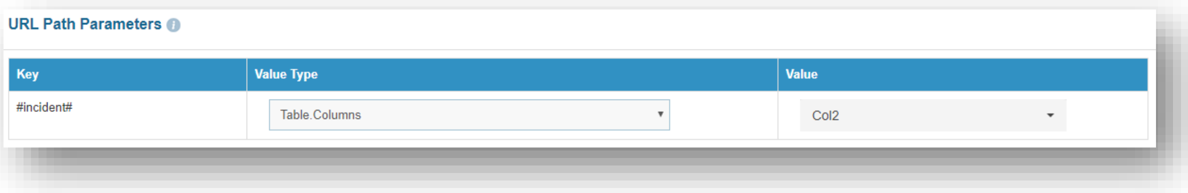
Input Type	CyberArk
Value	SNOWPassword

Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 81 – Password from Key Vault (CyberArk)

- **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #incident#

ValueType: Table Columns

Value:

Select from dropdown that mapped to sys_id from previous screen
"Col2"
```



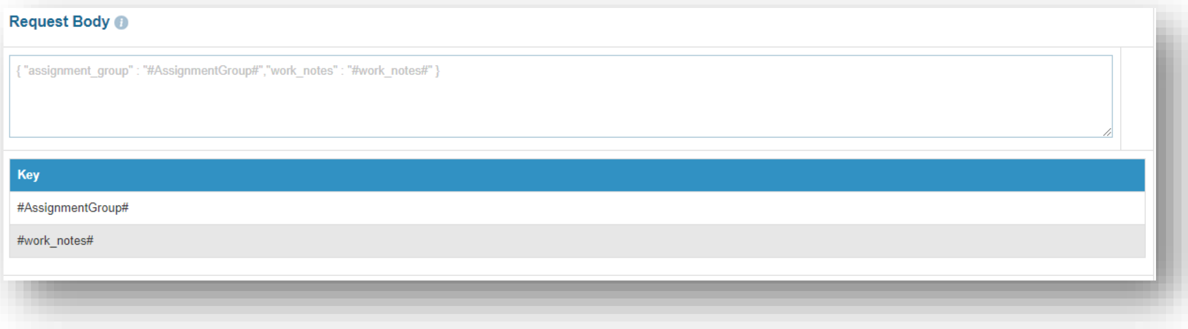
URL Path Parameters

Key	Value Type	Value
#incident#	Table.Columns	Col2

Figure 82 – Release Rules Configuration – Service Request Tasks (URL Path Parameters)

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body –** In this section, please enter the request body in JSON format. A sample request is mentioned below.

```
Request Body –

{ "assignment_group" : "#AssignmentGroup#","work_notes" :
"#work_notes#" }
```



Request Body

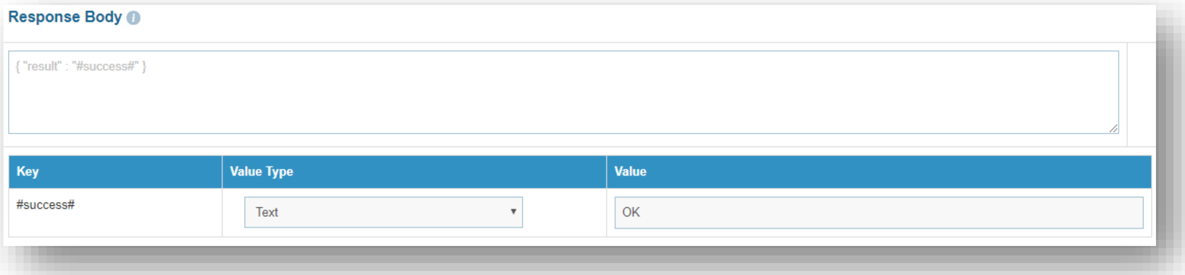
```
{ "assignment_group" : "#AssignmentGroup#","work_notes" : "#work_notes#" }
```

Key
#AssignmentGroup#
#work_notes#

Figure 83 – Release Rules Configuration – Service Request Tasks (Request Body)

- **Response Body –** In this section, please enter the response body in JSON format. A sample response is mentioned below.

```
Response Body –
{ "result" : "#success#" }
```



Response Body

{ "result" : "#success#" }

Key	Value Type	Value
#success#	Text	OK

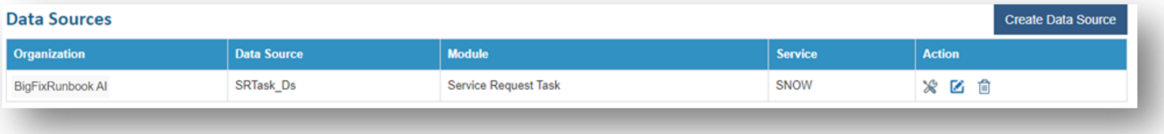
Figure 84 – Release Rules Configuration – Service Request Tasks (Response Body)

- **Response Key Value** mapping can be done as per the below table.

Table 17– Sample Response Key Value Mapping – Service Request Tasks

#success#	Text	OK

- Click **Submit** to add the data source.
- In order to bring the tickets within BigFix Runbook AI scope, a specific queue needs to be configured in the ITSM tool and same has to be configured in BigFix Runbook AI. This is achieved through **Manage the Entry Criteria**. Please perform the below steps –
- Go to Actions tab and click Manage Data Sources.
- On the **Data Sources** tab, click 🛠 next to the data source user wants to manage. **Manage Entry Criteria** screen appears.



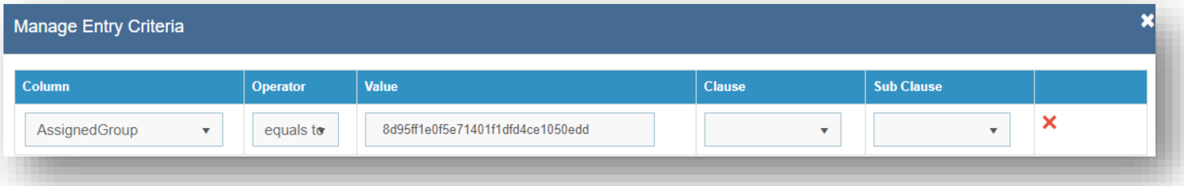
Data Sources

Organization	Data Source	Module	Service	Action
BigFixRunbook AI	SRTask_Ds	Service Request Task	SNOW	🛠 ✎ 🗑

Figure 85 – Manage Entry Criteria (Service Request Task)

- Select 'AssignedGroup' for the **Column** field and 'equals to' for the **Operator** field.
- Enter the sys_id of the assignment group in ServiceNow in the **Value** field.

- **Clause** and **Sub-Clause** fields can also be added based on requirement.



Manage Entry Criteria

Column	Operator	Value	Clause	Sub Clause	
AssignedGroup	equals to	8d95ff1e0f5e71401f1dfd4ce1050edd			✕

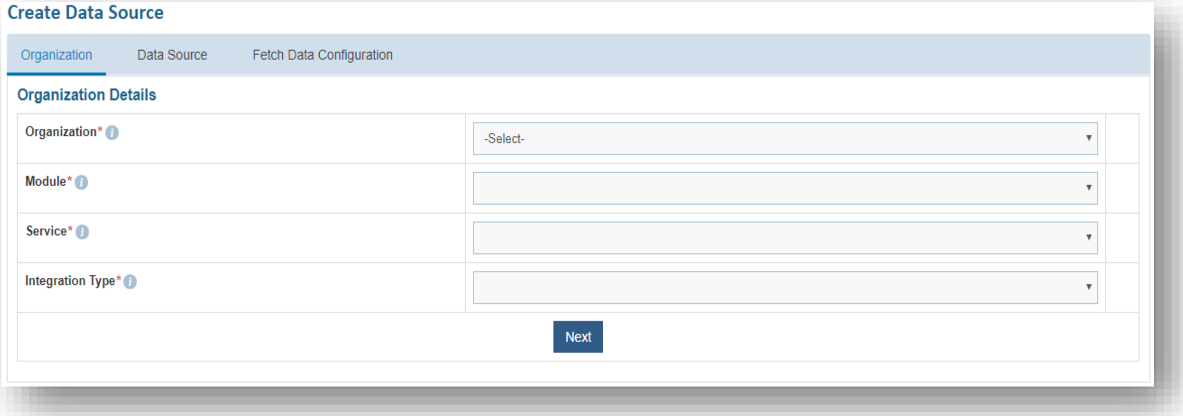
Figure 86 – Manage Entry Criteria (Service Request Task) cont.

- Click **Save**.

4.2.2.3 Create Data Source for Service Request Item

To create a data source for Service Requests Items, perform the following steps:

- On the main menu bar, click **Actions Tab → Manage Data Sources**.
- The **Create Data Source** page appears with the following tabs:
    - Organization
    - Data Source
    - Fetch Data Configuration



Create Data Source

Organization Details

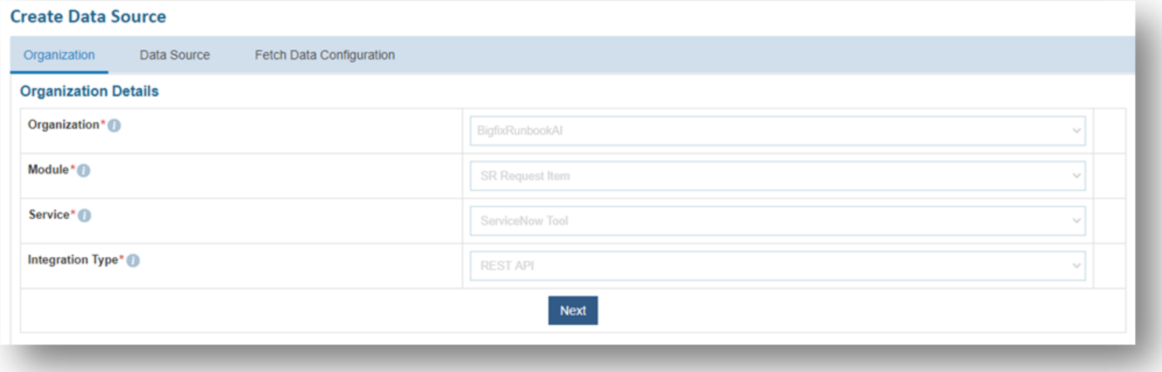
Organization*	-Select-
Module*	
Service*	
Integration Type*	

Figure 87 - Create Data Source – Service Request Item

Release Rules Configuration is only applicable for the following **Module** types- **Incident Management, Change Request Task and Service Request Task.** This tab will not be activated for other module types.

- On the **Organization** tab:
    - Select the **Organization Name** from the dropdown.

- In the **Module** field, select 'Service Request Item', since we are using this data source for using its field value for the Service Request Tasks.
- In the **Service** field, select 'Service Now Tool' as we are configuring the data source for ServiceNow.
- In the **Integration Type** field, select 'REST API', since we will be integrating through REST APIs.
- Click **Next**.



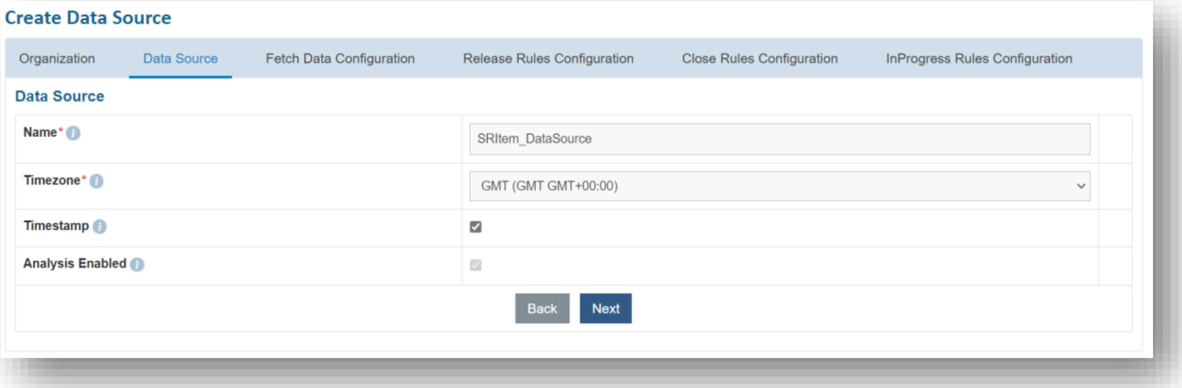
Organization Details

Organization*	BigfixRunbookAI
Module*	SR Request Item
Service*	ServiceNow Tool
Integration Type*	REST API

Figure 88 - Create Data Source – Service Request Item (cont.)

– On the **Data Source** tab:

- Type the new data source in the **Name** field.
- Select the **Timezone** to specify the time zone of the selected data source.
- Select **Timestamp** to view the present data with date and time.
- Select **Analysis Enabled?** if user wants to analyze the data retrieved from the data source.
- Click Next.


Create Data Source
Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration
Back
Next

Data Source

Name*	SRItem_DataSource
Timezone*	GMT (GMT GMT+00:00)
Timestamp	☑
Analysis Enabled	☑

Figure 89 - Create Data Source – Service Request Item (cont.)

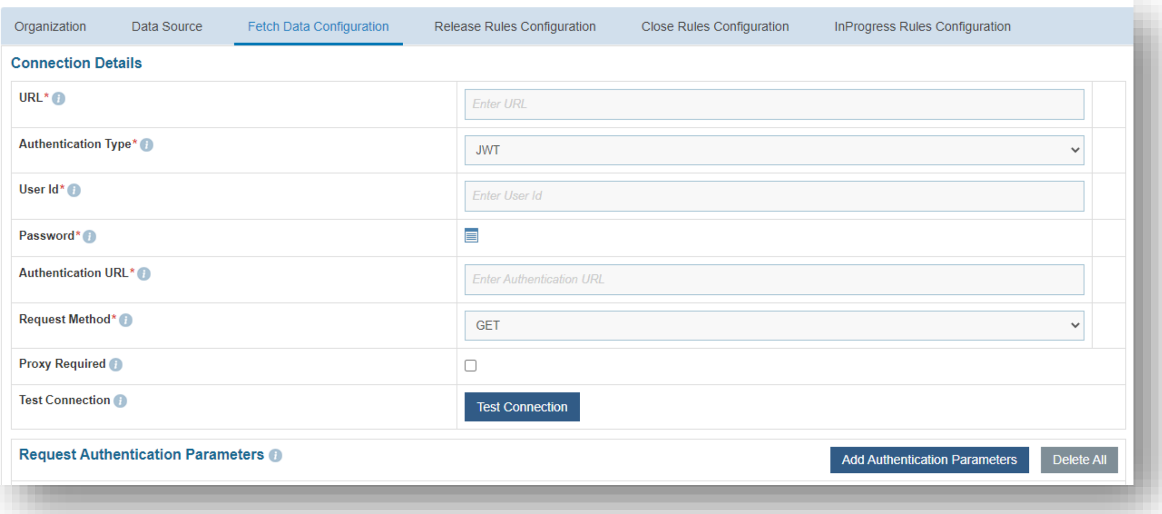
- On the **Fetch Data Configuration** tab, type in the details as per the environment.
- In the **Connection Details** section, enter the following details:
  - **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.
  - **Sample URL** - https://URL.service-now.com/api/now/v1/table/sc_req_item?sysparm_fields=#Columns#&sysparm_query=sys_updated_on>=#StartDate#^sys_updated_on<=#EndDate#^ORDERBYsys_updated_on
  - **Authentication Type** – Select one of the Authentication Types from Basic / Windows, JWT, OAuth 2.0

    Selection of **Basic / Windows** requires you to enter -

    - User Id
    - Password.

    Selection of **JWT / OAuth 2.0** requires you to enter -

    - User Id
    - Password
    - Authentication URL
  - **Request Body –** Select the request method as **GET**, **POST** or **PUT** as per the configured URL.
  - **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
  - Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.


Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

Connection Details

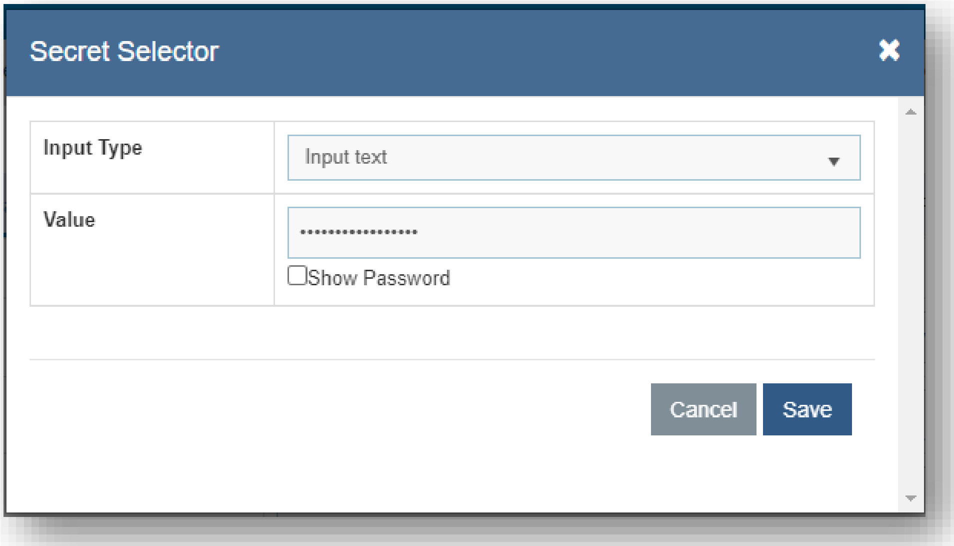
URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	GET
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

Add Authentication Parameters
Delete All

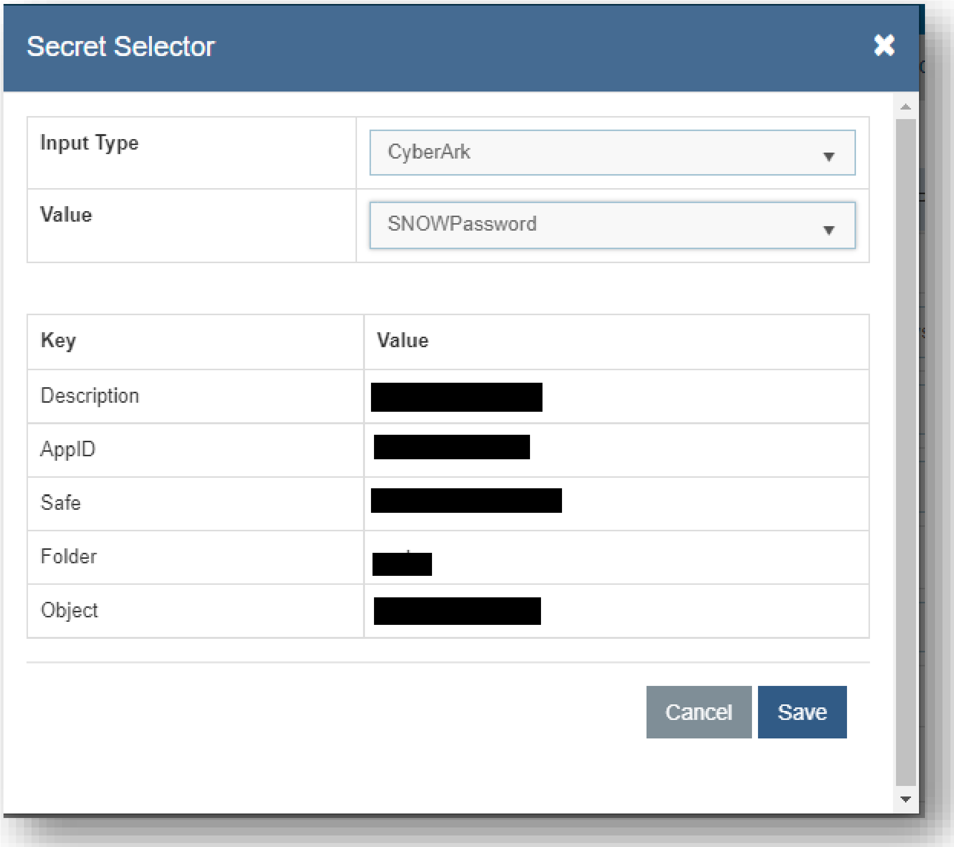
Figure 90 – Create Data Source – Service Request Item (Connection Details)

- **Password** – For password, click on icon next to it. If the password is available in plaintext then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.


Secret Selector

Input Type	Input text
Value	•••••••••••••••••• □Show Password

Figure 91 – Password in plaintext



Secret Selector

Input Type: CyberArk

Value: SNOWPassword

Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

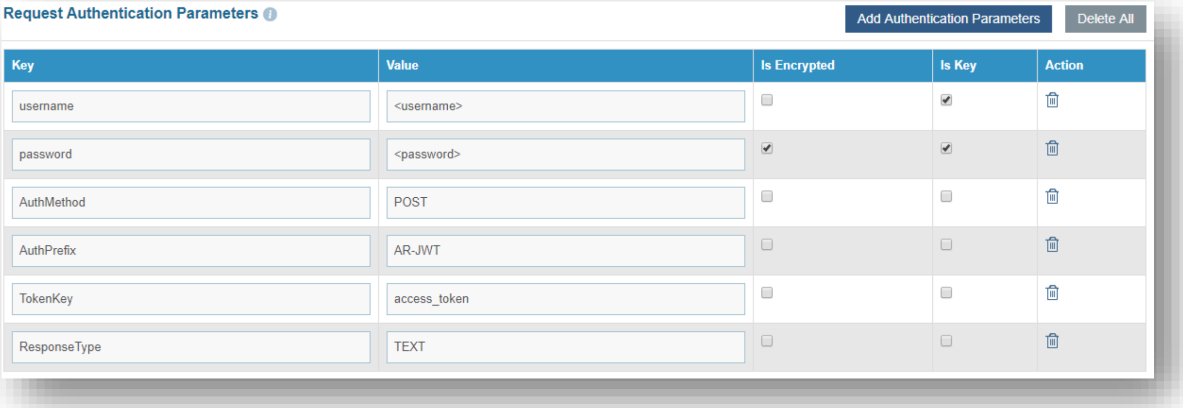
Figure 92 – Password from Key Vault (CyberArk)

- **Request Authentication Parameters** – If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.
- Based on the **Authentication Type**, add the parameters mentioned in the below table.

Table 18 – Sample Authentication Parameters – Service Request Item

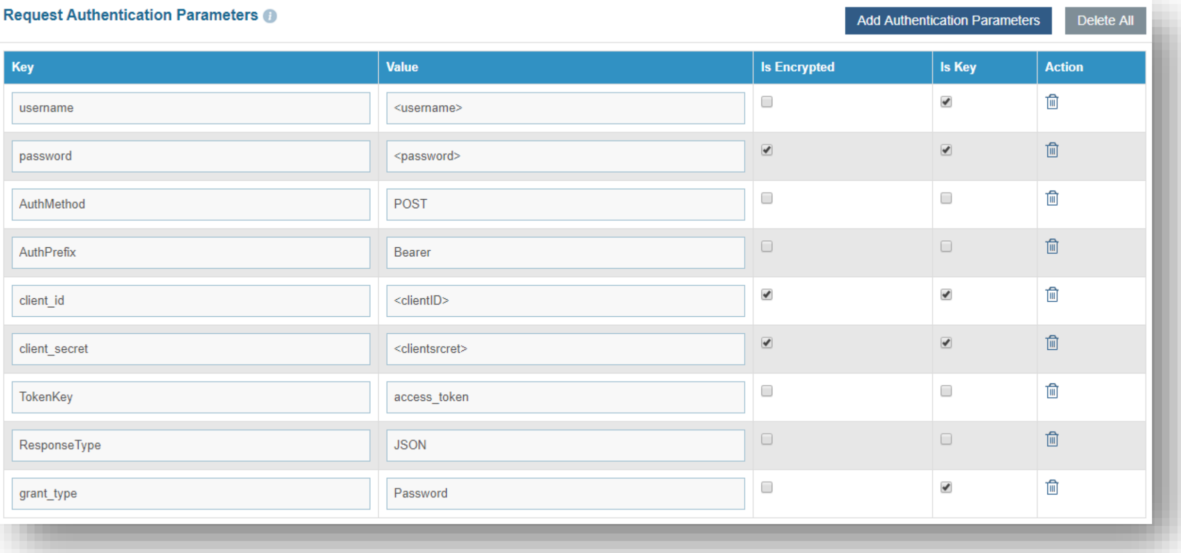
Authentication Type	Key	Value	Is Encrypted?	Is Key?
JWT	username	<username>	NO	YES
JWT	password	<password>	YES	YES
JWT	AuthMethod	POST	NO	NO
JWT	AuthPrefix	AR-JWT	NO	NO
JWT	TokenKey	access_token	NO	NO
JWT	ResponseType	TEXT	NO	NO
OAuth2.0	username	<username	NO	YES
OAuth2.0	password	<password>	YES	YES
OAuth2.0	AuthMethod	POST	NO	NO
OAuth2.0	AuthPrefix	Bearer	NO	NO

OAuth2.0	client_id	<clientID>	YES	YES
OAuth2.0	client_secret	<clientsecret>	YES	YES
OAuth2.0	TokenKey	access_token	NO	NO
OAuth2.0	ResponseType	JSON	NO	NO
OAuth2.0	grant_type	Password	NO	YES


Request Authentication Parameters
Add Authentication Parameters
Delete All

Key	Value	Is Encrypted	Is Key	Action
username	<username>	□	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	□	□	🗑
AuthPrefix	AR-JWT	□	□	🗑
TokenKey	access_token	□	□	🗑
ResponseType	TEXT	□	□	🗑

Figure 93 – Create Data Source – Service Request Item (Request Authentication Parameters for JWT)


Request Authentication Parameters
Add Authentication Parameters
Delete All

Key	Value	Is Encrypted	Is Key	Action
username	<username>	□	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	□	□	🗑
AuthPrefix	Bearer	□	□	🗑
client_id	<clientID>	☑	☑	🗑
client_secret	<clientsrcret>	☑	☑	🗑
TokenKey	access_token	□	□	🗑
ResponseType	JSON	□	□	🗑
grant_type	Password	□	☑	🗑

Figure 94 – Service Request Item (Request Authentication Parameters for OAuth2.0)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #Columns#
```

ValueType: Text

Value:

number,sys_updated_on,sys_id,sys_created_on,short_description,description,state,request,approval

Note – These columns are mandatory. User can add more columns if more data is required to be fetched from ITSM tool.

Key: #StartDate#

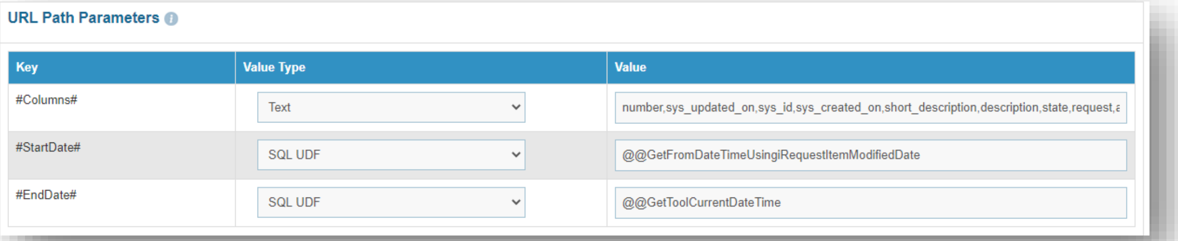
ValueType: SQL UDF

VALUE: @@GetFromDateTimeUsingiRequestItemModifiedDate

Key: #EndDate#

ValueType: SQL UDF

VALUE: @@GetToolCurrentDateTime



URL Path Parameters

Key	Value Type	Value
#Columns#	Text	number,sys_updated_on,sys_id,sys_created_on,short_description,description,state,request,a
#StartDate#	SQL UDF	@@GetFromDateTimeUsingiRequestItemModifiedDate
#EndDate#	SQL UDF	@@GetToolCurrentDateTime

Figure 95 – URL Path Parameters – Service Request Item (Service Request Task Management)

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body -** Enter the request body in JSON format as per the configured URL, if applicable.
- **Response Body –** In this section, please enter the output of URL query for one of the service request tasks in JSON format. A sample response is mentioned below:

Response Body –

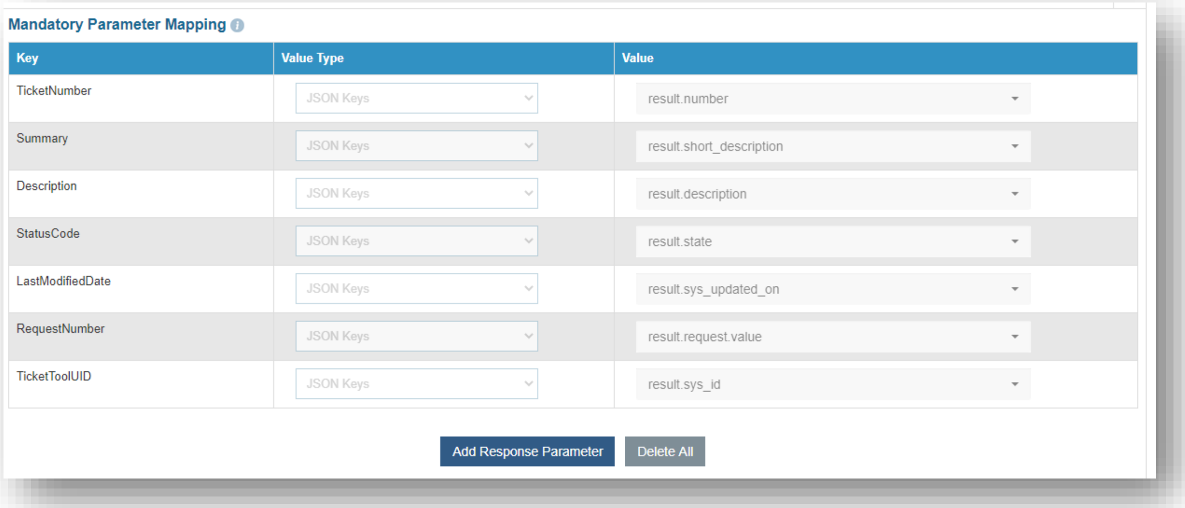
```
{
   "result": {
      "number": "RITM0011964",
      "sys_id": "6ee764d5db199c14e3bbde06f496195a",
      "short_description": "Can't find the right request?TEST",
      "request": {
         "link":"https://dryicegbpdevdemo.service-now.com/api/now/v1/table/sc_request/2ae764d5db199c14e3bbde06f496195a",
         "value": "2ae764d5db199c14e3bbde06f496195a"
      },
      "sys_created_on": "2020-06-08 10:34:54",
      "approval": "approved",
      "description": "Test",
      "sys_updated_on": "2020-06-08 10:35:17",
      "state": "2"
   }
}
```

- After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section**.**
- **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below:

Table 19– Sample Mandatory Mapping Parameters – Service Request Item

Key	Value Type	Value
TicketNumber	JSON.Keys	result.number
Summary	JSON.Keys	result.short_description
Description	JSON.Keys	result.description
StatusCode	JSON.Keys	result.state
LastModifiedDate	JSON.Keys	result.sys_updated_on

RequestNumber	JSON.Keys	result.request.value
TicketToolUID	JSON.Keys	result.sys_id



Mandatory Parameter Mapping

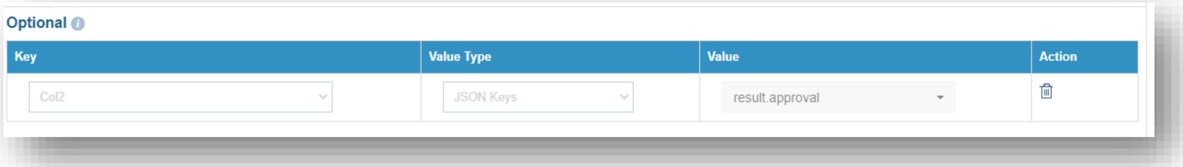
Key	Value Type	Value
TicketNumber	JSON Keys	result.number
Summary	JSON Keys	result.short_description
Description	JSON Keys	result.description
StatusCode	JSON Keys	result.state
LastModifiedDate	JSON Keys	result.sys_updated_on
RequestNumber	JSON Keys	result.request.value
TicketToolUID	JSON Keys	result.sys_id

Figure 96 – Mandatory Parameter Mapping (Service Request Item)

– If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 20– Sample Optional Mapping Parameters – Service Request Item

Key	Value Type	Value
Col2	JSON.Keys	result.approval



Optional

Key	Value Type	Value	Action
Col2	JSON Keys	result.approval	🗑

Figure 97 – Optional Parameter Mapping (Service Request Item)

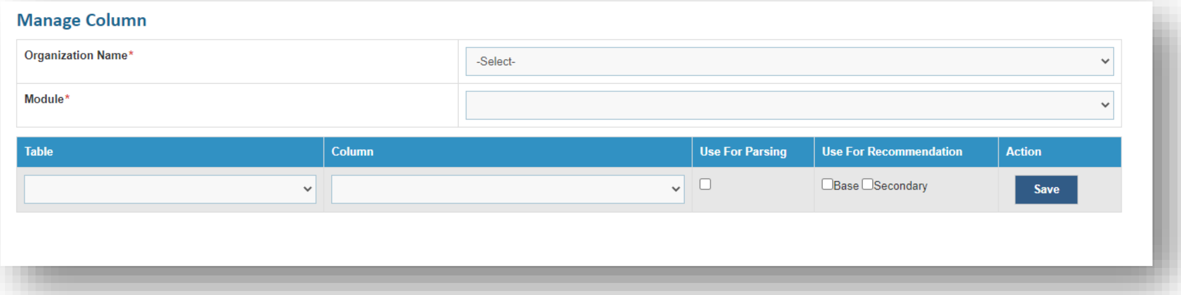
– Click **Submit** to add the data source.

4.2.2.4 Configuration of additional parameters for Recommendation and Parsing

To use the field values of Service Request and Service Request Item for the purpose of Recommendation and Parsing by BigFix Runbook AI services, they need to be mapped to Service Request Task.

To do so, perform the following steps -

– On the main menu bar, click Advance Configuration → Parameter → Manage Column.



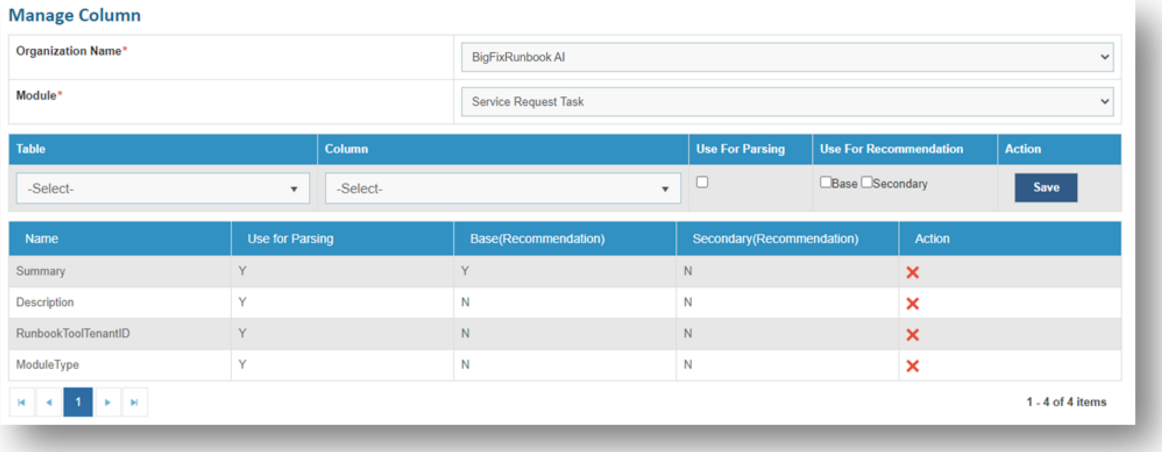
Manage Column

Organization Name*	-Select-
Module*	

Table	Column	Use For Parsing	Use For Recommendation	Action
		□	□Base □Secondary	Save

Figure 98 – Map fields of Service Request and Service Request Item to Service Request Task

– Select Organization Name from dropdown. Select Service Request Task as the Module.



Manage Column

Organization Name*	BigFixRunbook AI
Module*	Service Request Task

Table	Column	Use For Parsing	Use For Recommendation	Action
-Select-	-Select-	□	□Base □Secondary	Save

Name	Use for Parsing	Base(Recommendation)	Secondary(Recommendation)	Action
Summary	Y	Y	N	✕
Description	Y	N	N	✕
RunbookToolTenantID	Y	N	N	✕
ModuleType	Y	N	N	✕

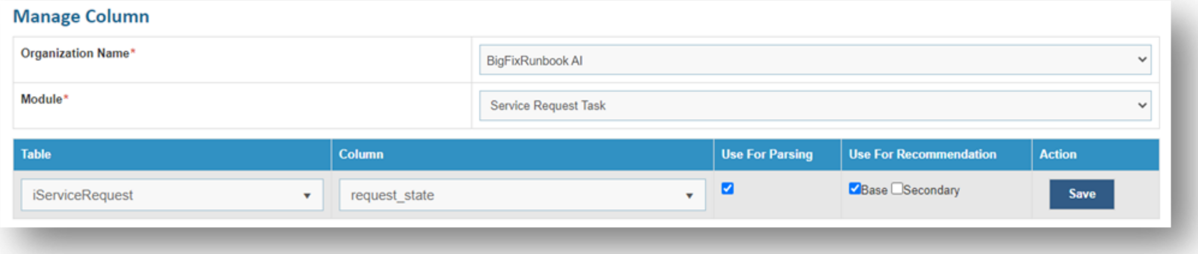
1 - 4 of 4 items

Figure 99 – Map fields of Service Request and Service Request Item to Service Request Task (cont.)

Summary, Description, RunbookToolTenantID, ModuleType are the default entries.

– To map the column of Service Request, select **iServiceRequest** in Table dropdown.

- Select the column of Service Request which has to be mapped to Service Request Task in the Column dropdown. In this case, we are selecting **request_state**.
- Check the fields **Use For Parsing** and select 'Base' in **Use For Recommendation**.



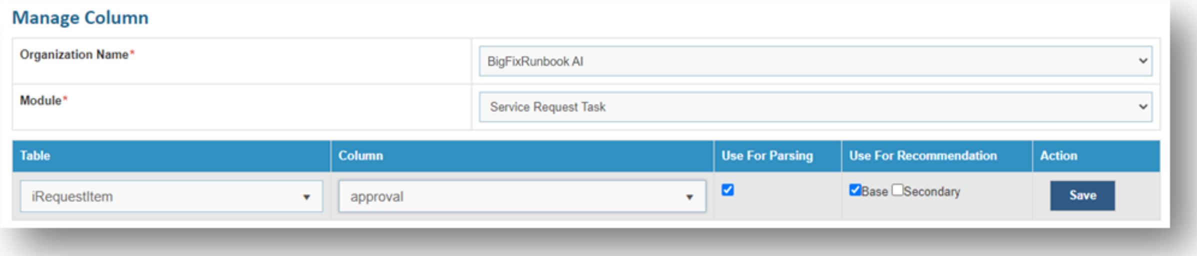
Manage Column

Organization Name*	BigFixRunbook AI
Module*	Service Request Task

Table	Column	Use For Parsing	Use For Recommendation	Action
iServiceRequest	request_state	☑	☑Base ☐Secondary	Save

Figure 100 – Map fields of Service Request and Service Request Item to Service Request Task (cont.)

- Click **Save.**
- To map the column of Service Request Item, select **iRequestItem** in Table dropdown.
- Select the column of Service Request Item which has to be mapped to Service Request Task in the Column dropdown. In this case, we are selecting **approval**.
- Check the fields **Use For Parsing** and select 'Base' in **Use For Recommendation**.



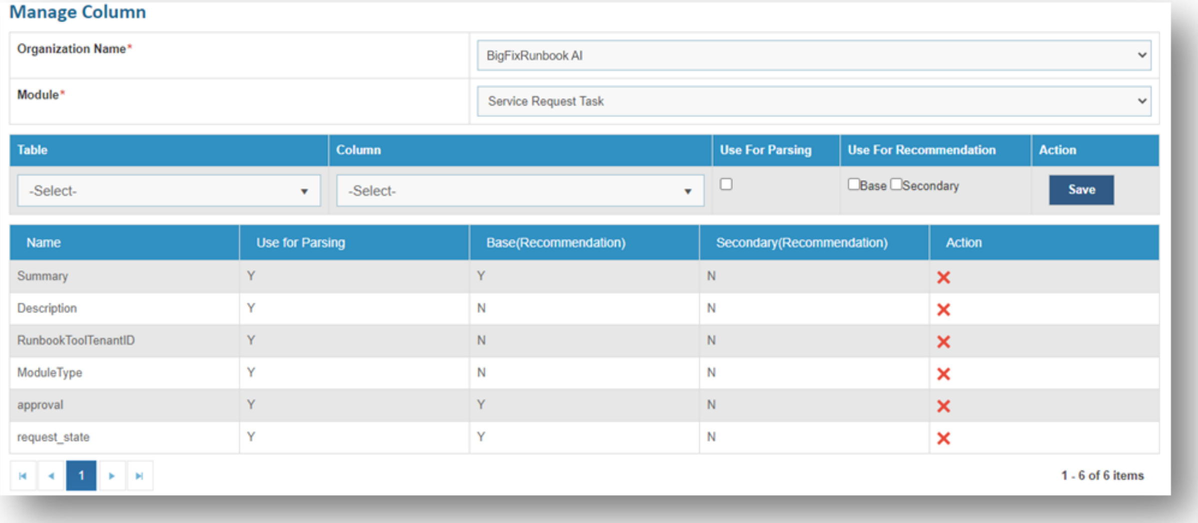
Manage Column

Organization Name*	BigFixRunbook AI
Module*	Service Request Task

Table	Column	Use For Parsing	Use For Recommendation	Action
iRequestItem	approval	☑	☑Base ☐Secondary	Save

Figure 101 – Map fields of Service Request and Service Request Item to Service Request Task (cont.)

- Click **Save**. The page lists two additional entries, **request_state** and **approval**, as depicted below.



Manage Column

Organization Name*	BigFixRunbook AI
Module*	Service Request Task

Table	Column	Use For Parsing	Use For Recommendation	Action
-Select-	-Select-	□	□Base □Secondary	Save

Name	Use for Parsing	Base(Recommendation)	Secondary(Recommendation)	Action
Summary	Y	Y	N	✕
Description	Y	N	N	✕
RunbookToolTenantID	Y	N	N	✕
ModuleType	Y	N	N	✕
approval	Y	Y	N	✕
request_state	Y	Y	N	✕

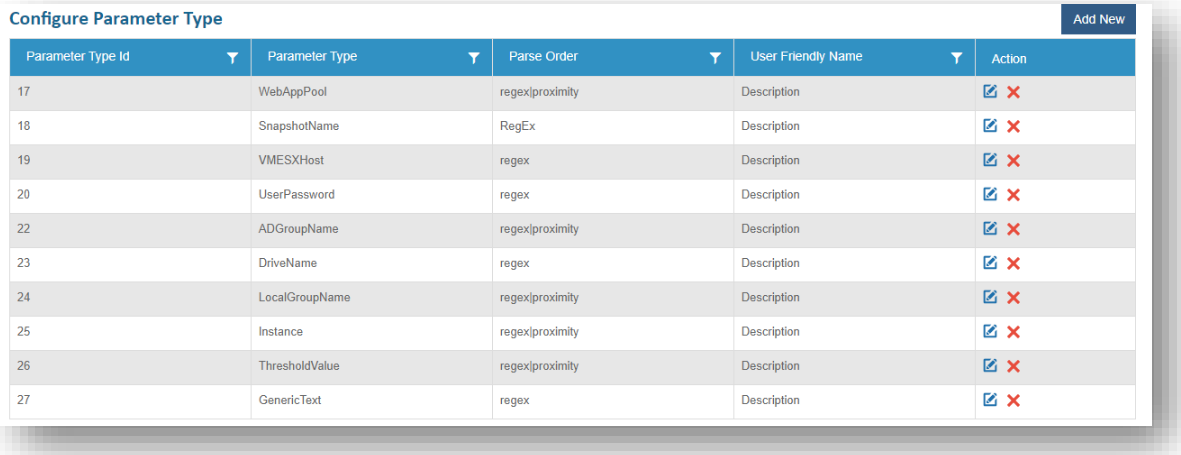
1 - 6 of 6 items

Figure 102 – Map fields of Service Request and Service Request Item to Service Request Task (cont.)

– For Recommendation, above steps are sufficient. But for Parsing, additional steps are required to be performed.

– On the main menu bar, click **Environment.**

– Click **Configure Parameter Type**. By default, there are several entries already defined.



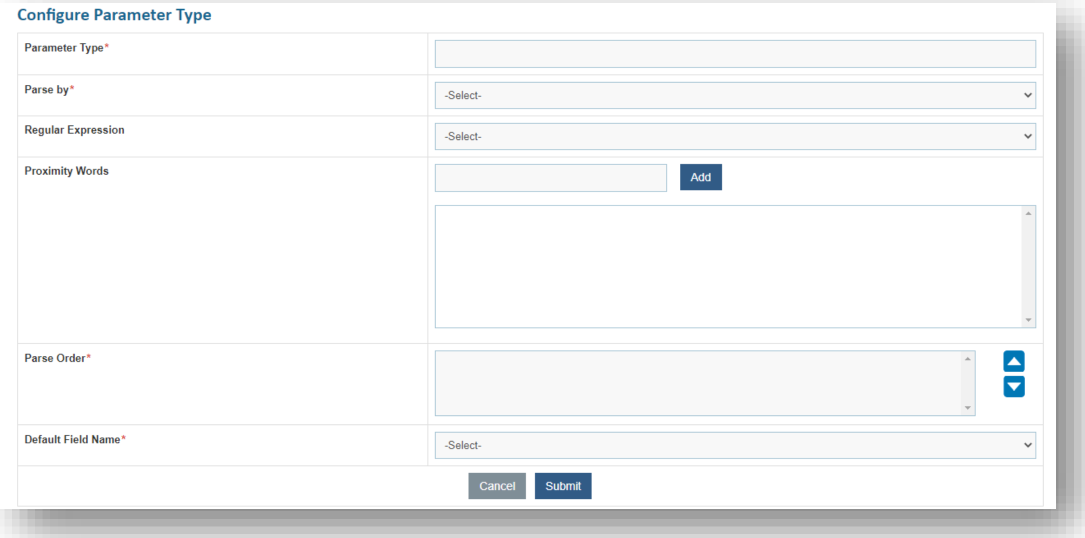
Configure Parameter Type

Add New

Parameter Type Id	Parameter Type	Parse Order	User Friendly Name	Action
17	WebAppPool	regex|proximity	Description	✎ ✕
18	SnapshotName	RegEx	Description	✎ ✕
19	VMESXHost	regex	Description	✎ ✕
20	UserPassword	regex	Description	✎ ✕
22	ADGroupName	regex|proximity	Description	✎ ✕
23	DriveName	regex	Description	✎ ✕
24	LocalGroupName	regex|proximity	Description	✎ ✕
25	Instance	regex|proximity	Description	✎ ✕
26	ThresholdValue	regex|proximity	Description	✎ ✕
27	GenericText	regex	Description	✎ ✕

Figure 103 – Map fields of Service Request and Service Request Item to Service Request Task (cont.)

– Click **Add New**.

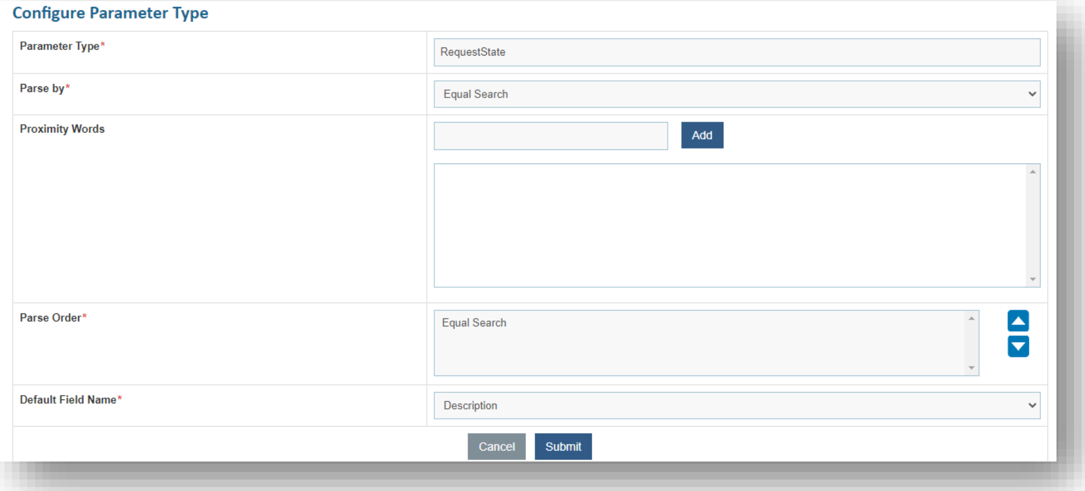


Configure Parameter Type

Parameter Type*	
Parse by*	-Select-
Regular Expression	-Select-
Proximity Words	Add
Parse Order*	
Default Field Name*	-Select-

Figure 104 – Map fields of Service Request and Service Request Item to Service Request Task (cont.)

- Mention **Parameter Type** for Service Request column, for e.g. **RequestState**
- Select 'Equal Search' in the **Parse By** field.
- Select 'Description' in the **Default Field Name** field.
- Click **Submit**.



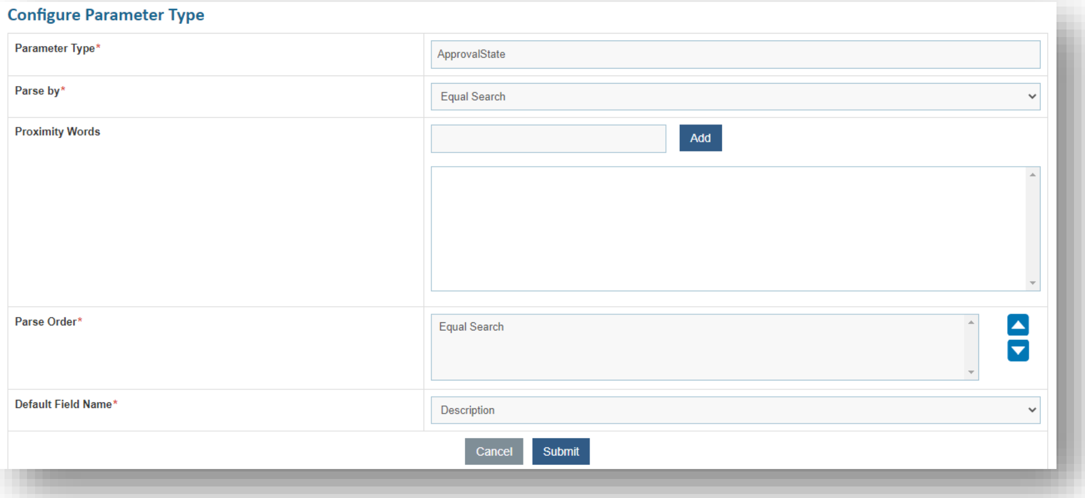
Configure Parameter Type

Parameter Type*	RequestState
Parse by*	Equal Search
Proximity Words	Add
Parse Order*	Equal Search
Default Field Name*	Description

Figure 105 – Map fields of Service Request and Service Request Item to Service Request Task (cont.)

- Click **Add New**.
- Mention **Parameter Type** for Service Request Item column, for e.g. **ApprovalState.**

- In the **Parse By** field, select ‘Equal Search’.
- In the **Default Field Name** field, select ‘Description’.
- Click **Submit**.

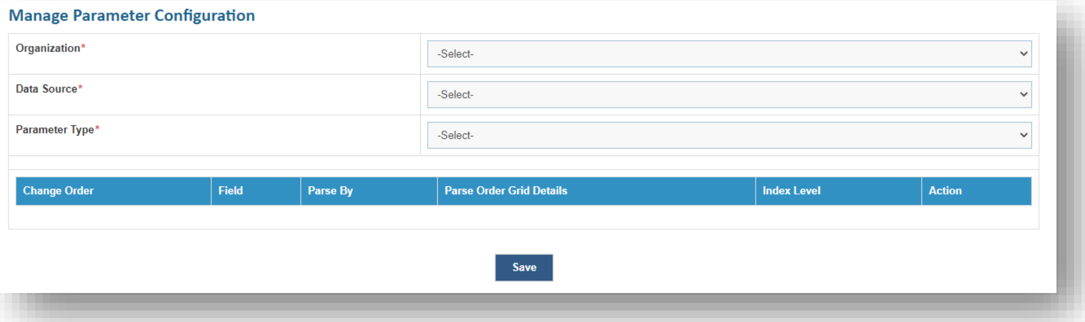


Configure Parameter Type

Parameter Type*	ApprovalState
Parse by*	Equal Search
Proximity Words	
Parse Order*	Equal Search
Default Field Name*	Description

Figure 106 – Map fields of Service Request and Service Request Item to Service Request Task (cont.)

- Next step is to map this **Parameter Type** i.e. ‘RequestState’ and ‘ApprovalState’, to the one that was created via **Manage Columns** in earlier step by the name ‘request_state’ and ‘approval’, respectively. To do that, perform the following steps:
- On the main menu bar, click Advance Configurations → Parameter.
- Click Manage Parameter Configuration.



Manage Parameter Configuration

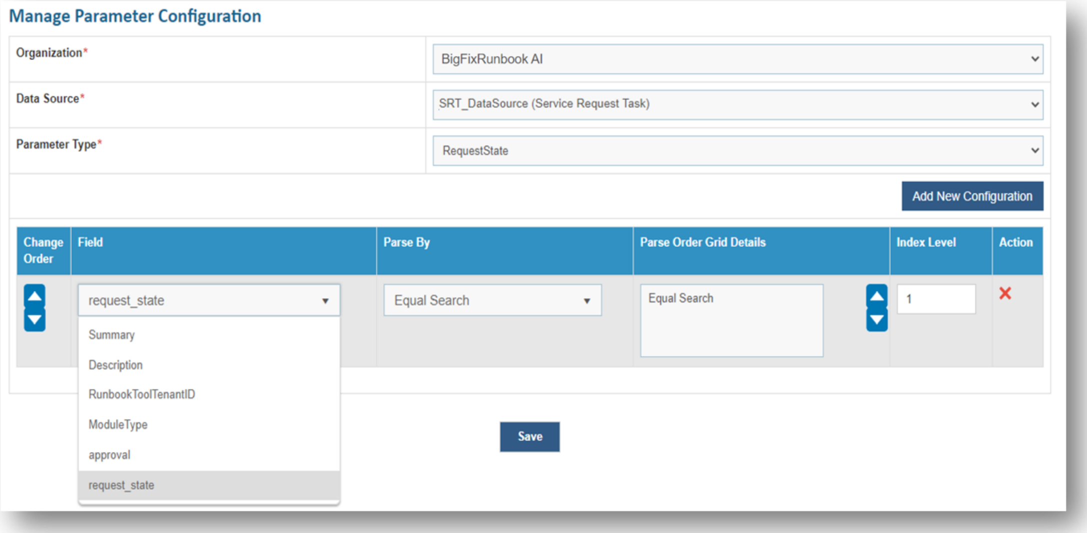
Organization*	-Select-
Data Source*	-Select-
Parameter Type*	-Select-

Change Order	Field	Parse By	Parse Order Grid Details	Index Level	Action

Figure 107 – Map fields of Service Request and Service Request Item to Service Request Task (cont.)

- Selection Organization.
- Select relevant ‘Service Request Task’ as the **Data Source.**

- Select the newly created parameter **RequestState** from **Parameter Type** dropdown**.**
- From the **Field** dropdown, select ‘request_state’, the parameter that has been mapped via **Manage Columns.**



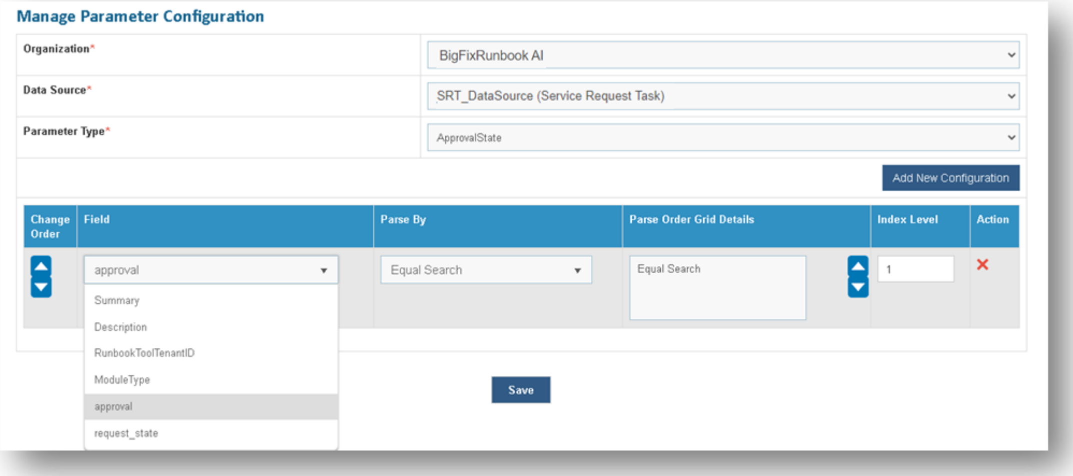
Manage Parameter Configuration

Organization*	BigFixRunbook AI
Data Source*	SRT_DataSource (Service Request Task)
Parameter Type*	RequestState

Change Order	Field	Parse By	Parse Order Grid Details	Index Level	Action
	request_state	Equal Search	Equal Search	1	✕

Figure 108 – Map fields of Service Request and Service Request Item to Service Request Task (cont.)

- Click **Save**.
- Selection Organization.
- Select relevant ‘Service Request Task’ as the **Data Source.**
- Select the newly created parameter i.e. ‘ApprovalState’ from **Parameter Type** dropdown**.**
- From the **Field** dropdown, select ‘approval’, the parameter that has been mapped via **Manage Columns.**
- Click **Save**.



Manage Parameter Configuration

Organization*	BigFixRunbook AI
Data Source*	SRT_DataSource (Service Request Task)
Parameter Type*	ApprovalState

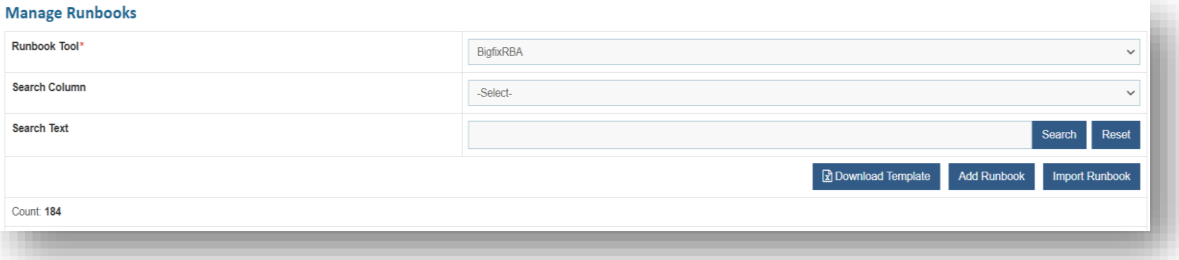
Add New Configuration

Change Order	Field	Parse By	Parse Order Grid Details	Index Level	Action
	approval	Equal Search	Equal Search	1	×

Figure 109 – Map fields of Service Request and Service Request Item to Service Request Task (cont.)

– To verify whether this parameter is successfully parsed or not, perform the following steps -

- On the main menu bar, click **Runbooks**.
- Click Manage Runbooks.
- Select the **Runbook Tool** mapped with the organization.



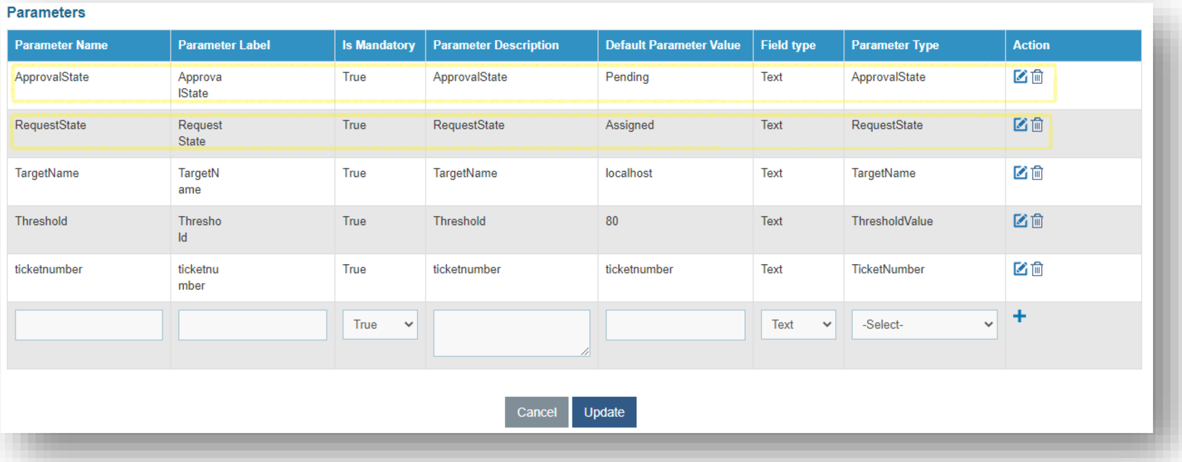
Manage Runbooks

Runbook Tool*	BigfixRBA
Search Column	-Select-
Search Text	

Count: 184

Figure 110 – Map fields of Service Request and Service Request Item to Service Request Task (cont.)

- The parameter, **RequestState** and **ApprovalState**, which were created in earlier steps, have to be added as the parameters to the existing runbook. You can also create a new runbook with **RequestState** and **ApprovalState** as the parameters.
- Click the **Edit** icon to edit the runbook.
- In the Parameters section, add two new parameters with relevant **Parameter Name**, **Parameter Label**, **Parameter Description**, **Default Parameter Value**. Ensure that Parameter Type is selected as **RequestState** and **ApprovalState** respectively.



Parameters

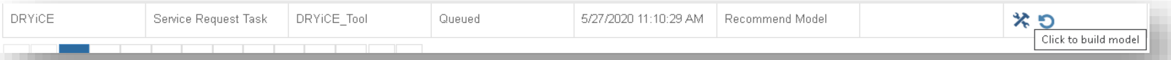
Parameter Name	Parameter Label	Is Mandatory	Parameter Description	Default Parameter Value	Field type	Parameter Type	Action
ApprovalState	ApprovalState	True	ApprovalState	Pending	Text	ApprovalState	✎ 🗑
RequestState	Request State	True	RequestState	Assigned	Text	RequestState	✎ 🗑
TargetName	TargetName	True	TargetName	localhost	Text	TargetName	✎ 🗑
Threshold	Threshold	True	Threshold	80	Text	ThresholdValue	✎ 🗑
ticketnumber	ticketnumber	True	ticketnumber	ticketnumber	Text	TicketNumber	✎ 🗑
		True			Text	-Select-	+

Figure 111 – Map fields of Service Request and Service Request Item to Service Request Task (cont.)

- Add the parameters and click **Update**.
- Ensure that the runbook in which the parameters are added is mapped with the organization.

– Next step is to build the Recommendation model and to do that perform the following steps:

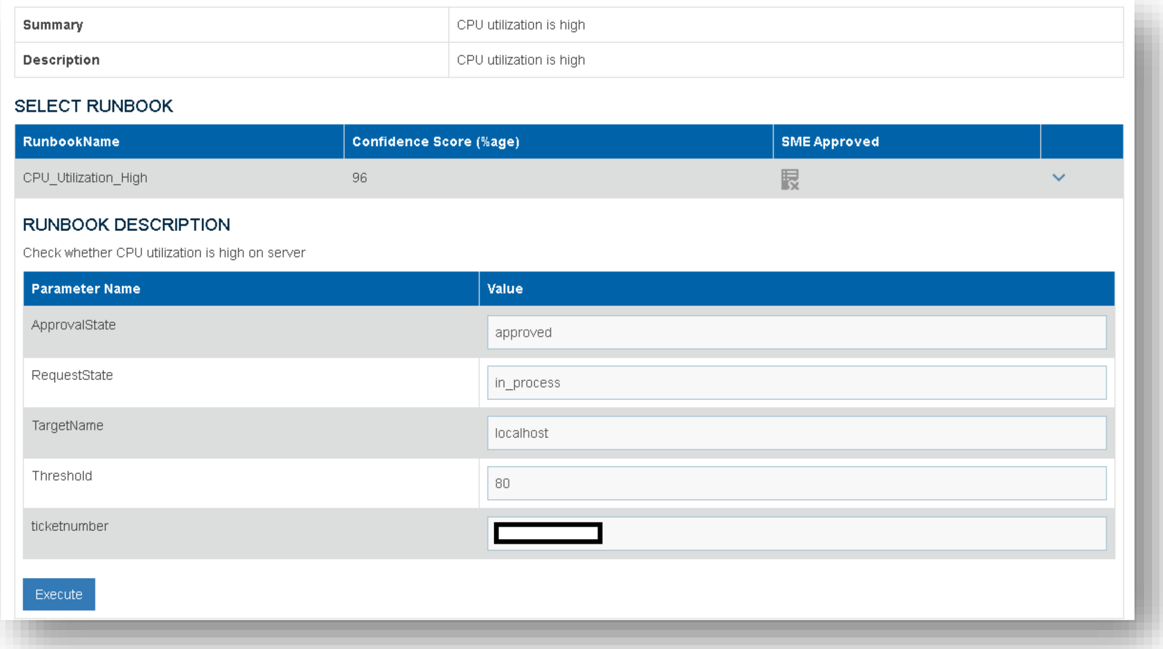
- On the main menu bar, click **Actions tabs → Runbooks**.
- Click Build Models.
- ReBuild / Re-build the model for the **Organization** under **Service Request Task** module for the mapped runbook tool.



DRYiCE	Service Request Task	DRYiCE_Tool	Queued	5/27/2020 11:10:29 AM	Recommend Model	

Figure 112 – Map fields of Service Request and Service Request Item to Service Request Task (cont.)

- Run the entire flow and see if the runbook recommended for the ticket in which the parameters were added have the parameter **RequestState** and **ApprovalState** with their expected values.



Summary	CPU utilization is high
Description	CPU utilization is high

SELECT RUNBOOK

RunbookName	Confidence Score (%age)	SME Approved	
CPU_Utilization_High	96		

RUNBOOK DESCRIPTION

Check whether CPU utilization is high on server

Parameter Name	Value
ApprovalState	approved
RequestState	in_process
TargetName	localhost
Threshold	80
ticketnumber	

Figure 113 – Map fields of Service Request and Service Request Item to Service Request Task (cont.)

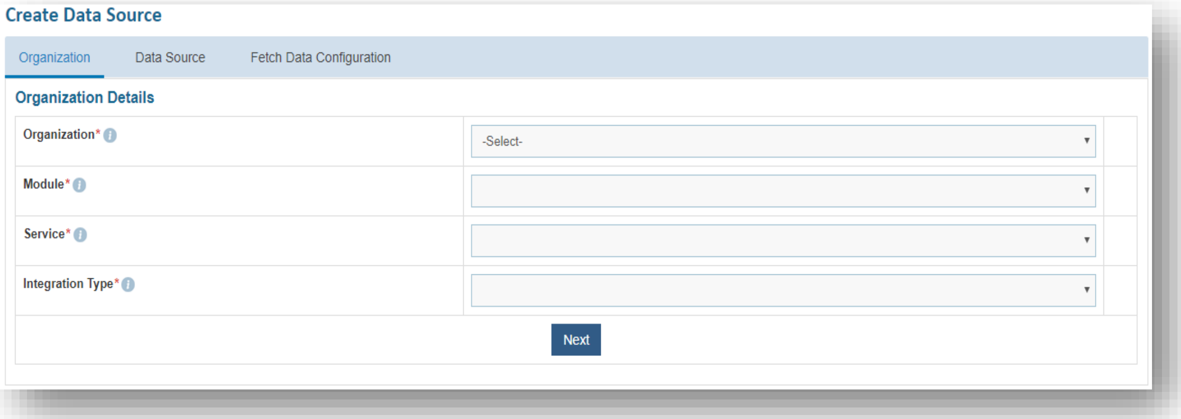
4.2.3 Change Request Management

To fetch information about Change Requests, usually, creation of a data source for Change Request Task should suffice. However, there could be scenarios where some additional fields / values are required from Change Request for processing the tickets – recommending the relevant runbooks and parsing the tickets to extract relevant parameters, for which separate data source for Change Request has to be created. Here, we will cover the procedure for creating both kinds of data sources.

4.2.3.1 Create Data Source for Change Request

To create a data source for Change Request, perform the following steps:

- On the main menu bar, click **Actions Tab → Manage Data Sources**.
- The **Create Data Source** page appears with the following tabs:
  - Organization
  - Data Source
  - Fetch Data Configuration


Create Data Source

Organization Details

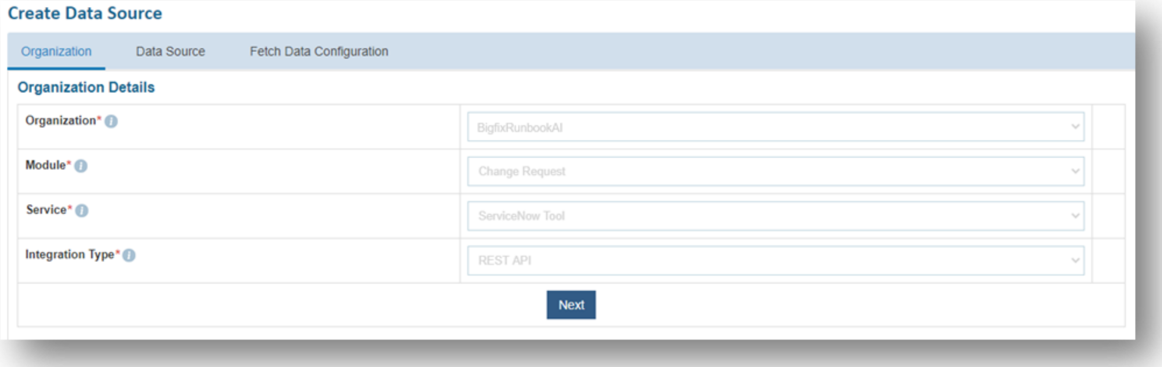
Organization*	-Select-
Module*	
Service*	
Integration Type*	

Figure 114 - Create Data Source – Change Request

Release Rules Configuration is only applicable for the following **Module** types- **Incident Management, Change Request Task and Service Request Task.** This tab will not be activated for other module types.

- On the **Organization** tab,
  - Select the **Organization Name** from the dropdown.

- Select the **Module** as **Change Request** since we are using this data source for using its field value for the change requests.
- Select the **Service** as **Service Now Tool** as we are configuring the data source for ServiceNow
- Select the **Integration Type** as **REST**, since we will be integrating through REST APIs.
- Click **Next**.



Create Data Source

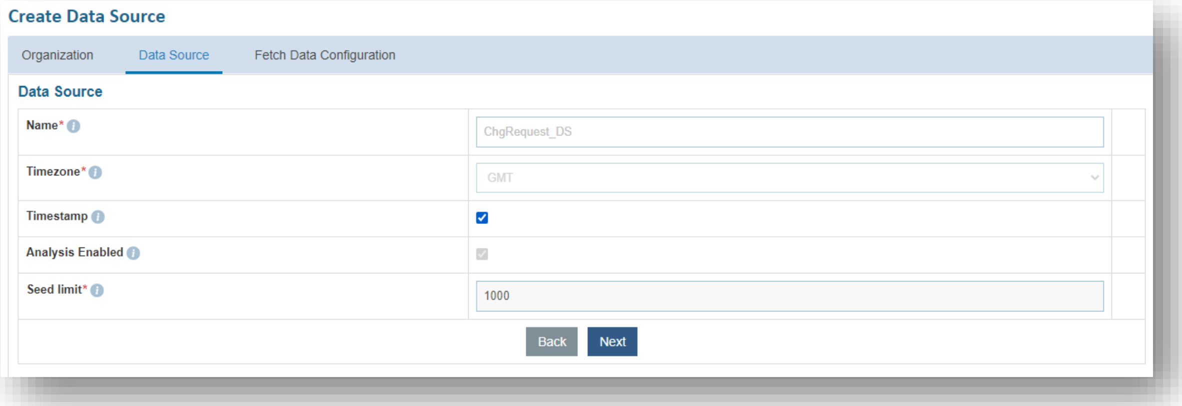
Organization Details

Organization*	BigfixRunbookAI
Module*	Change Request
Service*	ServiceNow Tool
Integration Type*	REST API

Figure 115 - Create Data Source – Change Request (cont.)

– On the **Data Source** tab:

- Type the new data source in the **Name** field.
- Select the **Timezone** to specify the time zone of the selected data source.
- Select **Timestamp** to view the present data with date and time.
- Select **Analysis Enabled** if user wants to analyze the data retrieved from the data source.
- Click **Next**.



Create Data Source

Data Source

Name*	ChgRequest_DS
Timezone*	GMT
Timestamp	☑
Analysis Enabled	☑
Seed limit*	1000

Figure 116 - Create Data Source – Change Request (cont.)

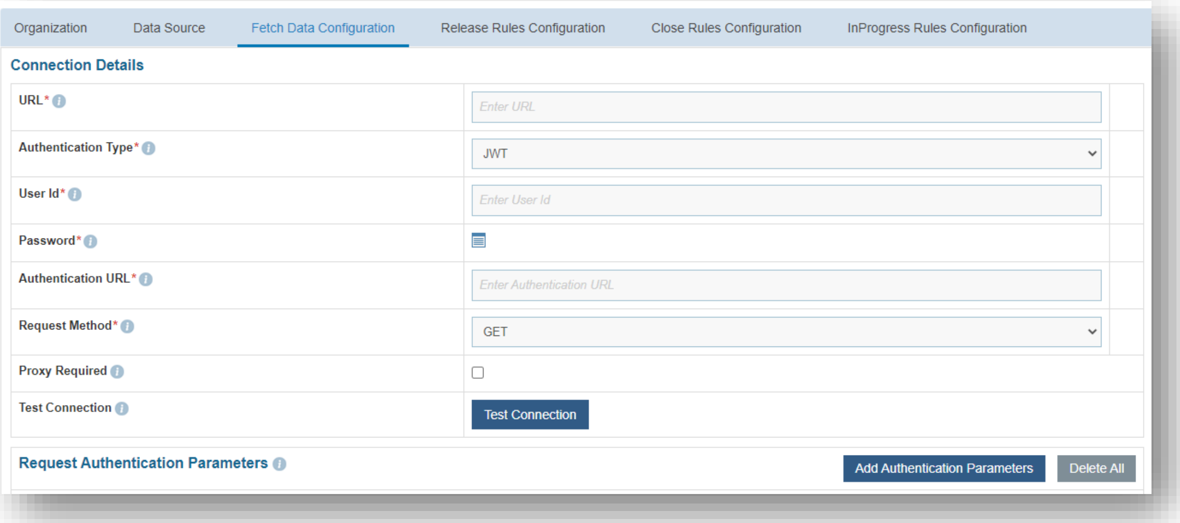
- On the **Fetch Data Configuration** tab, type in the details as per the environment.
- In the **Connection Details** section, enter the following details:
  - **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.
  - **Sample URL** - https://URL.service-now.com/api/now/v1/table/ change_request?sysparm_fields=#Columns#&sysparm_query=active=true^ sys_updated_on >=#StartDate#^ sys_updated_on <=#EndDate#^ORDERBYsys_updated_on
  - **Authentication Type** – Select one of the Authentication Types from Basic / Windows, JWT, OAuth 2.0

    Selection of **Basic / Windows** requires you to enter -

    - User Id
    - Password

    Selection of **JWT / OAuth 2.0** requires you to enter -

    - User Id
    - Password
    - Authentication URL
  - **Request Body –** Select the request method as GET, POST or PUT as per the configured URL.
  - **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
  - Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

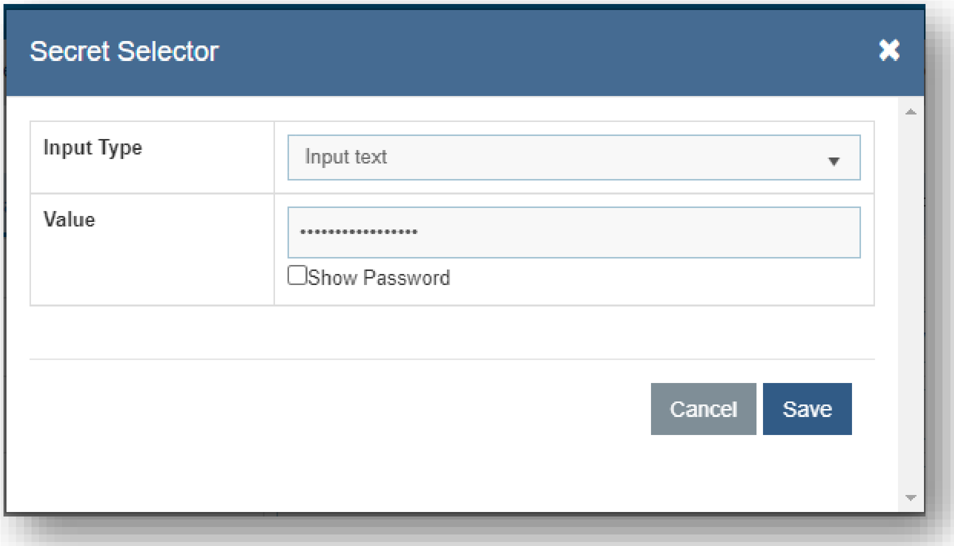
Connection Details

URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	GET
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters
Add Authentication Parameters
Delete All

Figure 117 – Create Data Source – Change Request (Connection Details)

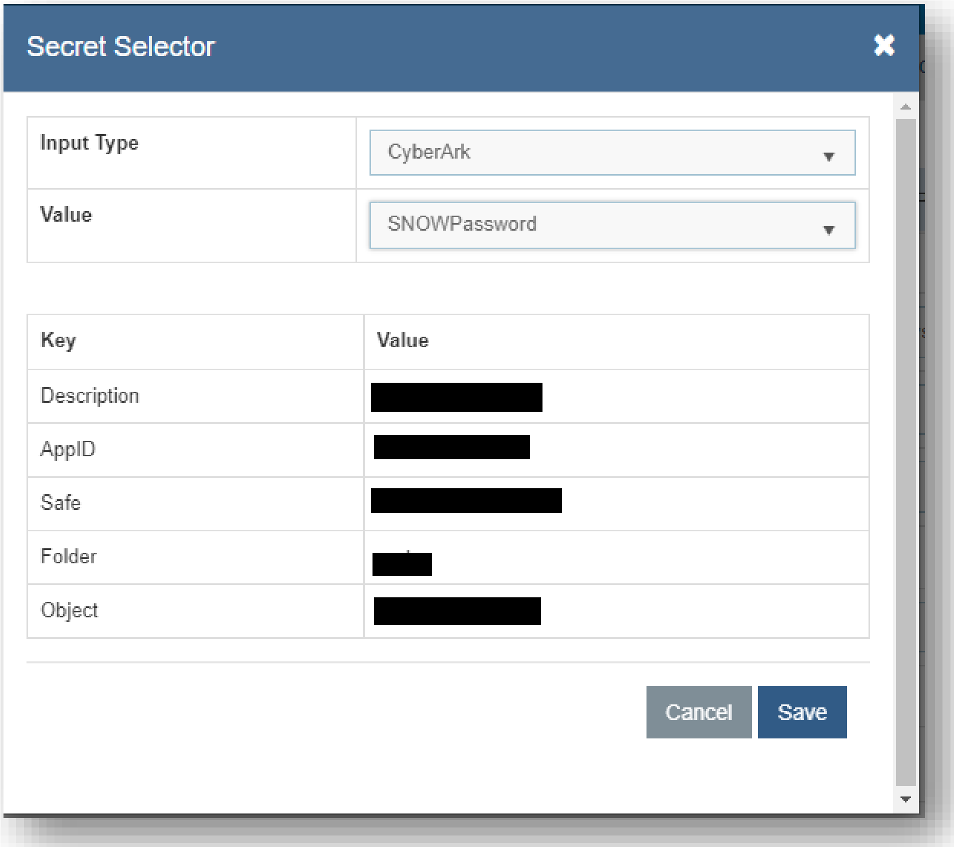
- **Password** – For password, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Secret Selector

Input Type	Input text
Value	••••••••••••••••• □Show Password

Figure 118 – Password in plaintext



Secret Selector

Input Type	CyberArk
Value	SNOWPassword

Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

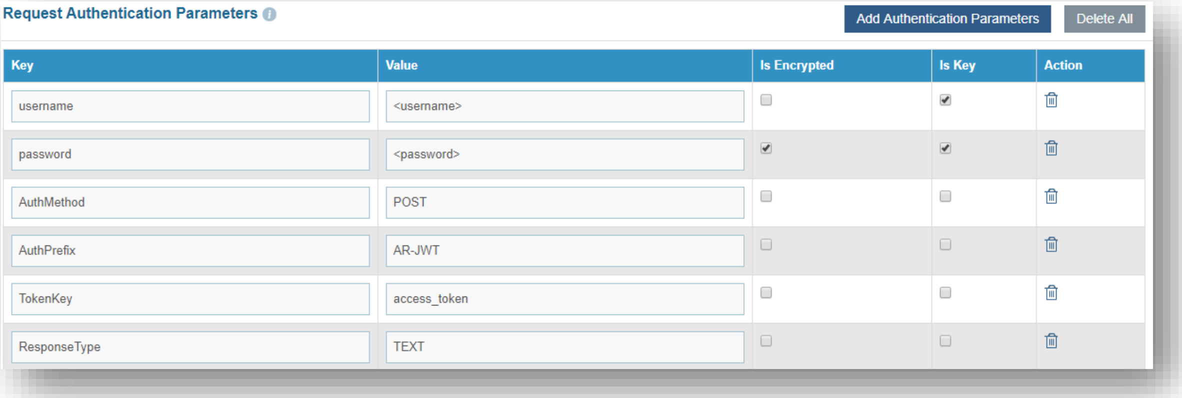
Figure 119 – Password from Key Vault (CyberArk)

- **Request Authentication Parameters** – If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.
- Based on the **Authentication Type**, add the parameters mentioned in the below table.

Table 21– Sample Authentication Parameters– Change Request

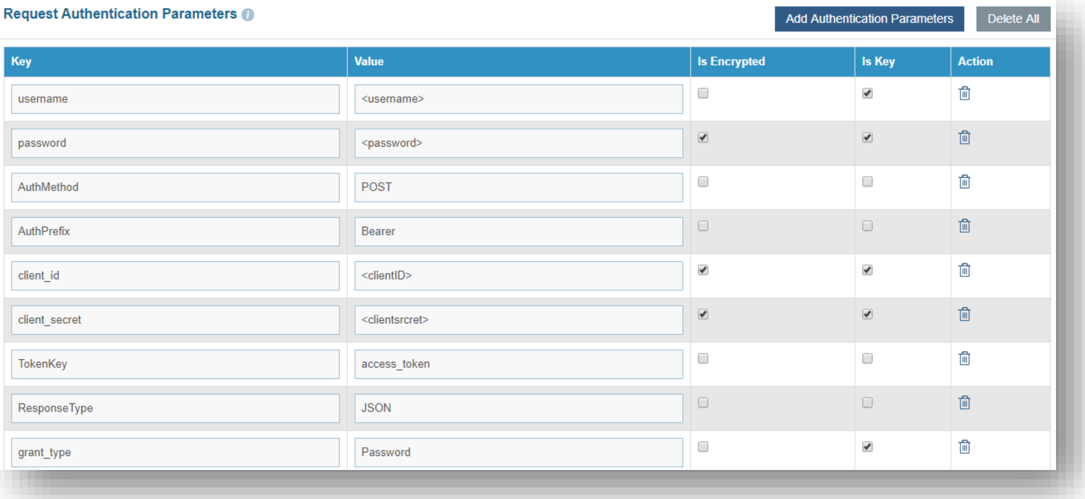
Authentication Type	Key	Value	Is Encrypted?	Is Key?
JWT	username	<username>	NO	YES
JWT	password	<password>	YES	YES
JWT	AuthMethod	POST	NO	NO
JWT	AuthPrefix	AR-JWT	NO	NO
JWT	TokenKey	access_token	NO	NO
JWT	ResponseType	TEXT	NO	NO
OAuth2.0	username	<username	NO	YES
OAuth2.0	password	<password>	YES	YES
OAuth2.0	AuthMethod	POST	NO	NO

Authentication Type	Key	Value	Is Encrypted?	Is Key?
OAuth2.0	AuthPrefix	Bearer	NO	NO
OAuth2.0	client_id	<clientID>	YES	YES
OAuth2.0	client_secret	<clientsrcret>	YES	YES
OAuth2.0	TokenKey	access_token	NO	NO
OAuth2.0	ResponseType	JSON	NO	NO
OAuth2.0	grant_type	Password	NO	YES


Request Authentication Parameters
Add Authentication Parameters
Delete All

Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	AR-JWT	☐	☐	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	TEXT	☐	☐	🗑

Figure 120 – Create Data Source – Change Request (Request Authentication Parameters for JWT)

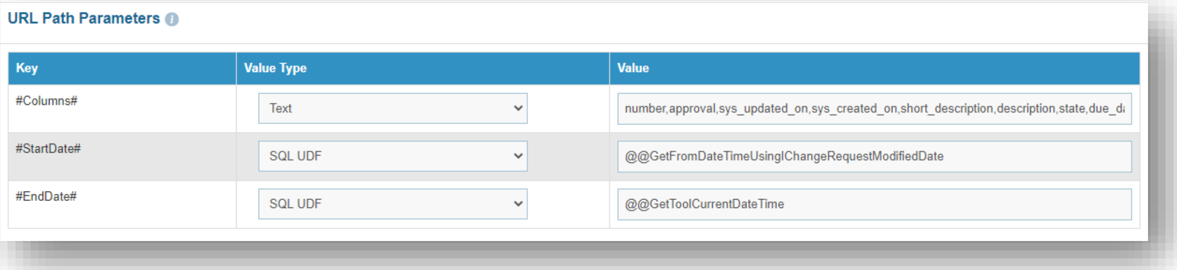

Request Authentication Parameters
Add Authentication Parameters
Delete All

Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	Bearer	☐	☐	🗑
client_id	<clientID>	☑	☑	🗑
client_secret	<clientsrcret>	☑	☑	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	JSON	☐	☐	🗑
grant_type	Password	☐	☑	🗑

Figure 121 – Change Request (Request Authentication Parameters for OAuth2.0)

- **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #Columns#

ValueType: Text

Value:

number, approval,sys_updated_on,sys_created_on,short_description,
description,state,due_date,
change_request,sys_id,assignment_group,priority


Note – These columns are mandatory. User can add more columns if
more data is required to be fetched from ITSM tool.


Key: #StartDate#

ValueType: SQL UDF

VALUE: @@GetFromDateTimeUsingIChangeRequestModifiedDate


Key: #EndDate#

ValueType: SQL UDF

VALUE: @@GetToolCurrentDateTime
```



URL Path Parameters

Key	Value Type	Value
#Columns#	Text	number,approval,sys_updated_on,sys_created_on,short_description,description,state,due_d
#StartDate#	SQL UDF	@@GetFromDateTimeUsingIChangeRequestModifiedDate
#EndDate#	SQL UDF	@@GetToolCurrentDateTime

Figure 122 – URL Parameters (Change Request)

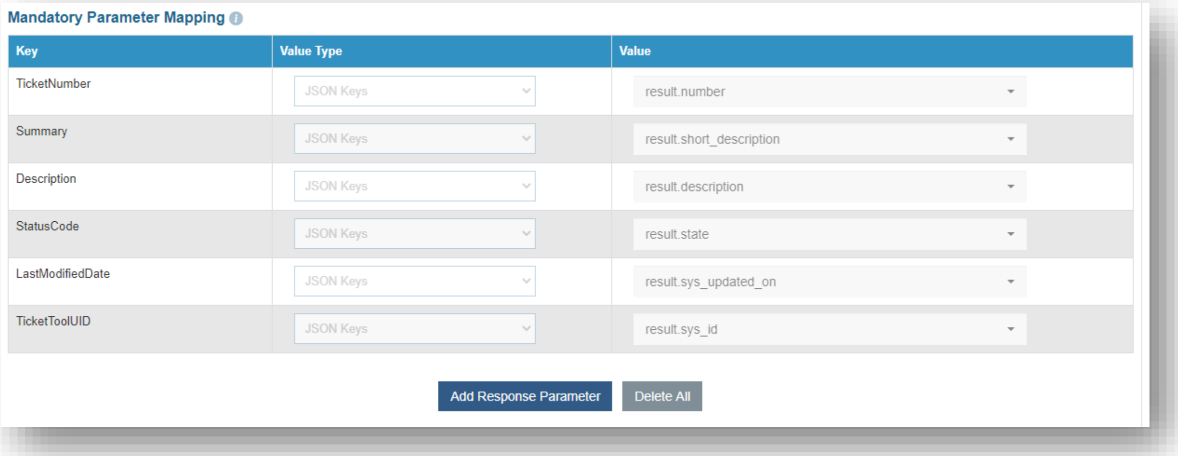
- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body -** Enter the request body in JSON format as per the configured URL, if applicable.
- **Response Body –** In this section, please enter the output of URL query for one of the incidents in JSON format. A sample response is mentioned below.

```
Response Body –

{ "result": {"sys_updated_on": "2018-03-18 13:59:04","number":
"CHG556563","approval":"approved","priority":"4","sys_created_on":
"2018-03-18 13:59:02","state": "1", "short_description":
"Implementation Task", "description": "Please initiate the
Implementation process.","sys_id":
"d612a2a34ff85b40b2627d918110c7ef","expected_start": "2018-03-19
13:58:31",

"change_request": {"link": "https://hclgbpdev.service-
now.com/api/now/v1/table/change_request/c6c12e634ff85b40b2627d9181
10c724","value": "c6c12e634ff85b40b2627d918110c724" },

"assignment_group":{

   "link": "https://dryicegbpdevdemo.service-
now.com/api/now/v1/table/sys_user_group/73be6572db1bdf00ce29b6bffe
96193d",

   "value": "73be6572db1bdf00ce29b6bffe96193d"

}

 }}
```

- After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section**.**
- **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below:

Table 22– Sample Mandatory Mapping Parameters– Change Request

Key	Value Type	Value
TicketNumber	JSON.Keys	result.number
Summary	JSON.Keys	result.short_description
Description	JSON.Keys	result.description
StatusCode	JSON.Keys	result.state
LastModifiedDate	JSON.Keys	result.sys_updated_on
TicketToolUID	JSON.Keys	result.sys_id



Mandatory Parameter Mapping

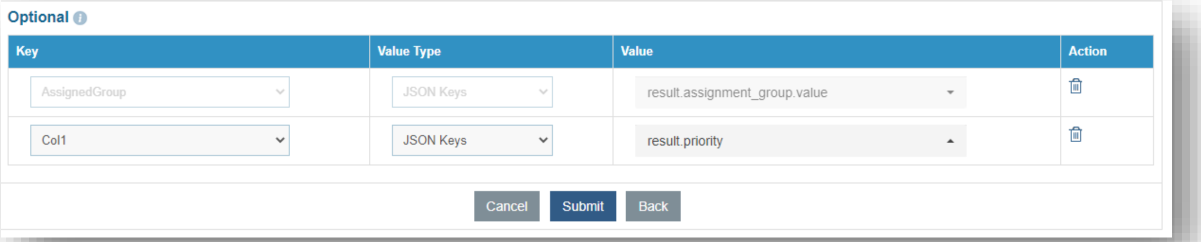
Key	Value Type	Value
TicketNumber	JSON Keys	result.number
Summary	JSON Keys	result.short_description
Description	JSON Keys	result.description
StatusCode	JSON Keys	result.state
LastModifiedDate	JSON Keys	result.sys_updated_on
TicketToolUID	JSON Keys	result.sys_id

Figure 123 – Mandatory Parameter Mapping (Change Request)

– If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 23 – Sample Optional Mapping Parameters– Change Request

Key	Value Type	Value
AssignedGroup	JSON.Keys	result.assignment_group.value
Col1	JSON.Keys	result.sys_id



Optional

Key	Value Type	Value	Action
AssignedGroup	JSON Keys	result.assignment_group.value	🗑
Col1	JSON Keys	result.priority	🗑

Figure 124 – Optional Parameter Mapping (Change Request)

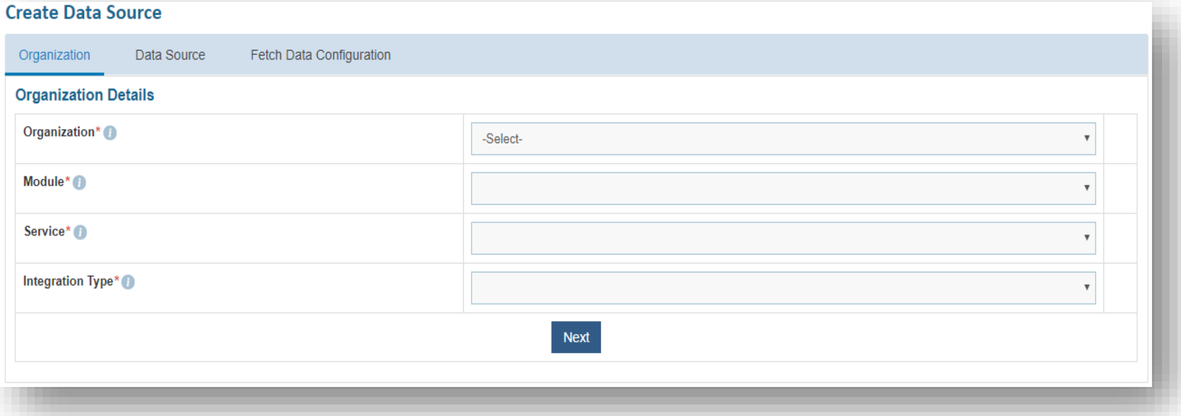
– Click **Submit** to add the data source.

4.2.3.2 Create Data Source for Change Request Task

To create a data source for Change Request Task Management, perform the following steps:

– On the main menu bar, click **Actions Tab → Manage Data Sources**.

- The **Create Data Source** page appears with the following tabs:
  - Organization
  - Data Source
  - Fetch Data Configuration
  - Release Rules Configuration


Create Data Source

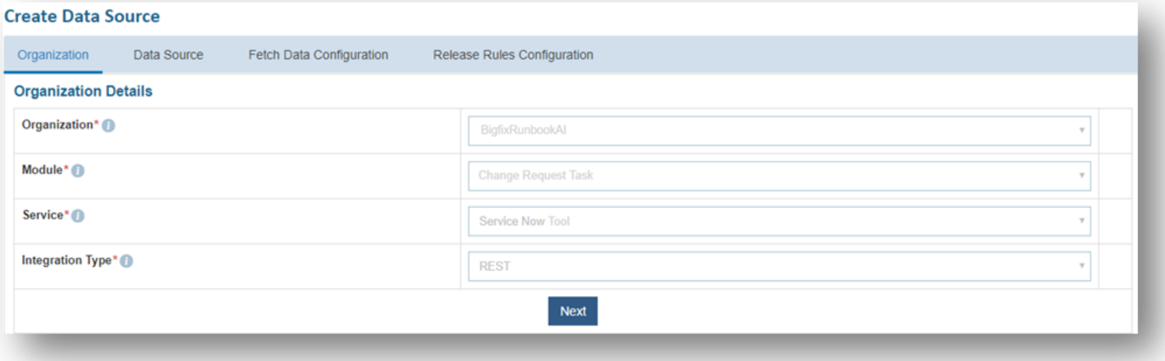
Organization Details

Organization*	-Select-
Module*	
Service*	
Integration Type*	

Figure 125 - Create Data Source – Change Request Task

Release Rules Configuration is only applicable for the following **Module** types- **Incident Management, Change Request Task and Service Request Task.** This tab will not be activated for other module types.

- On the **Organization** tab,
  - Select the **Organization Name** from the dropdown.
  - In the **Module** field, select 'Change Request Task', since we are configuring this data source for pulling the change requests.
  - In the **Service** field, select 'Service Now Tool' as we are configuring the data source for ServiceNow.
  - In the **Integration Type** field, select **REST**, since we will be integrating through REST APIs.
  - Click **Next**.



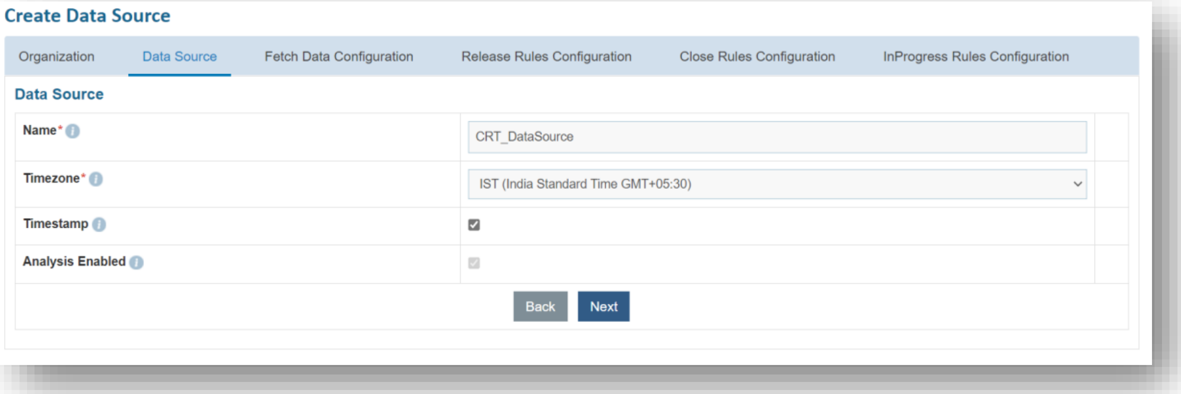
Organization Details

Organization*	BigfixRunbookAI
Module*	Change Request Task
Service*	Service Now Tool
Integration Type*	REST

Figure 126 - Create Data Source – Change Request Task (cont.)

– On the **Data Source** tab,

- Type the new data source in the **Name** field.
- Select the **Timezone** to specify the time zone of the selected data source.
- Select **Timestamp** to view the present data with date and time.
- Select **Analysis Enabled?** if user wants to analyze the data retrieved from the data source.
- Click Next.


Create Data Source
Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration
Back
Next

Data Source

Name*	CRT_DataSource
Timezone*	IST (India Standard Time GMT+05:30)
Timestamp	☑
Analysis Enabled	☑

Figure 127 - Create Data Source – Change Request Task (cont.)

– On the **Fetch Data Configuration** tab, type in the details as per the environment.

– In the **Connection Details** section, enter the following details:

- **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.

- **Sample URL** - https://<URL>.service-now.com/api/now/v1/table/change_task?sysparm_fields=#Columns#&sysparm_query=active=true^ sys_updated_on >=#StartDate#^ sys_updated_on <=#EndDate#^ORDERBYsys_updated_on
- **Authentication Type** – Select one of the Authentication Types from Basic / Windows, JWT, OAuth 2.0

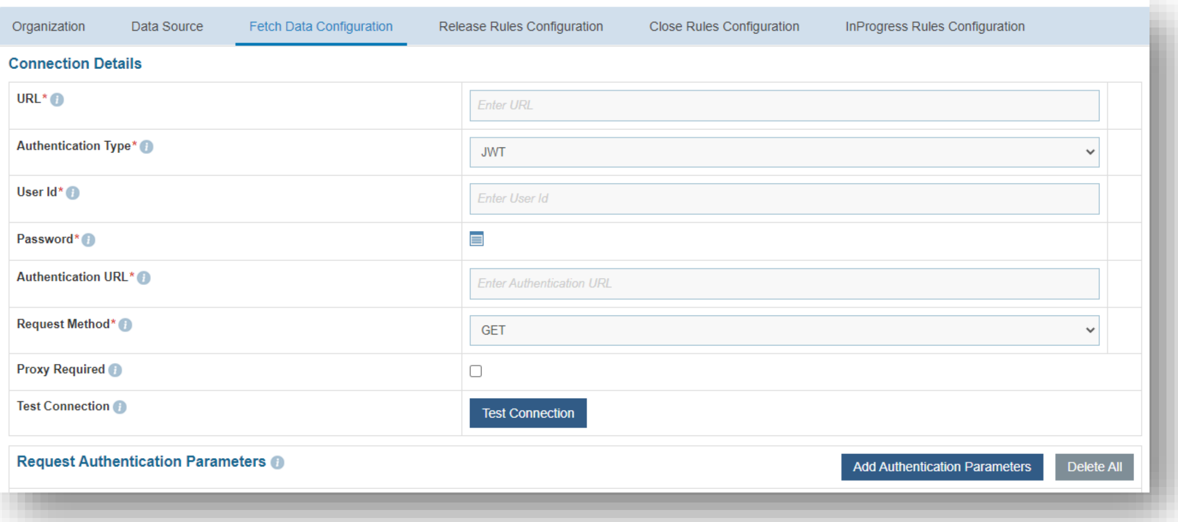
  Selection of **Basic / Windows** requires you to enter -

  - User Id
  - Password

  Selection of **JWT / OAuth 2.0** requires you to enter -

  - User Id
  - Password
  - Authentication URL

- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.


Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration
Test Connection
Add Authentication Parameters
Delete All

Connection Details

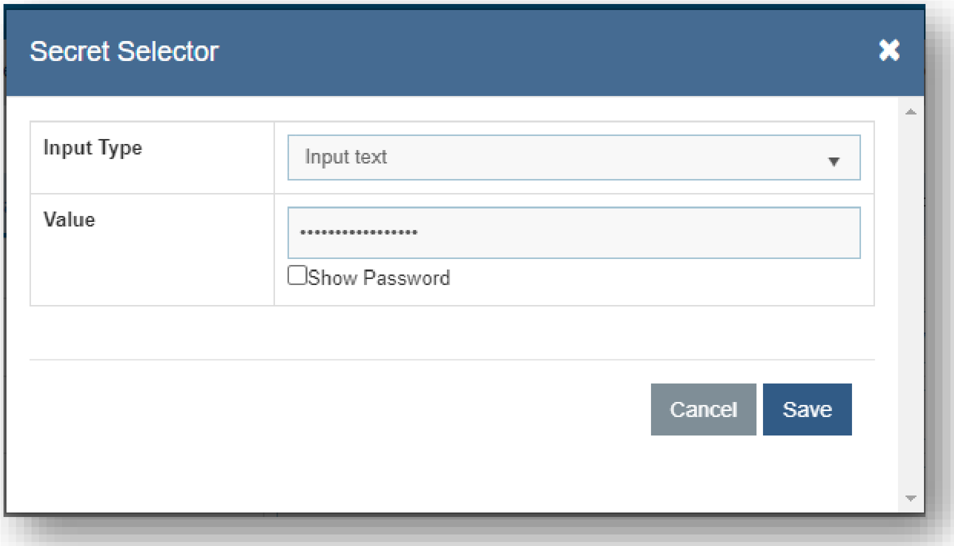
URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	GET
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

Figure 128 – Create Data Source – Change Request Task (Connection Details)

- **Password** – For password, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in

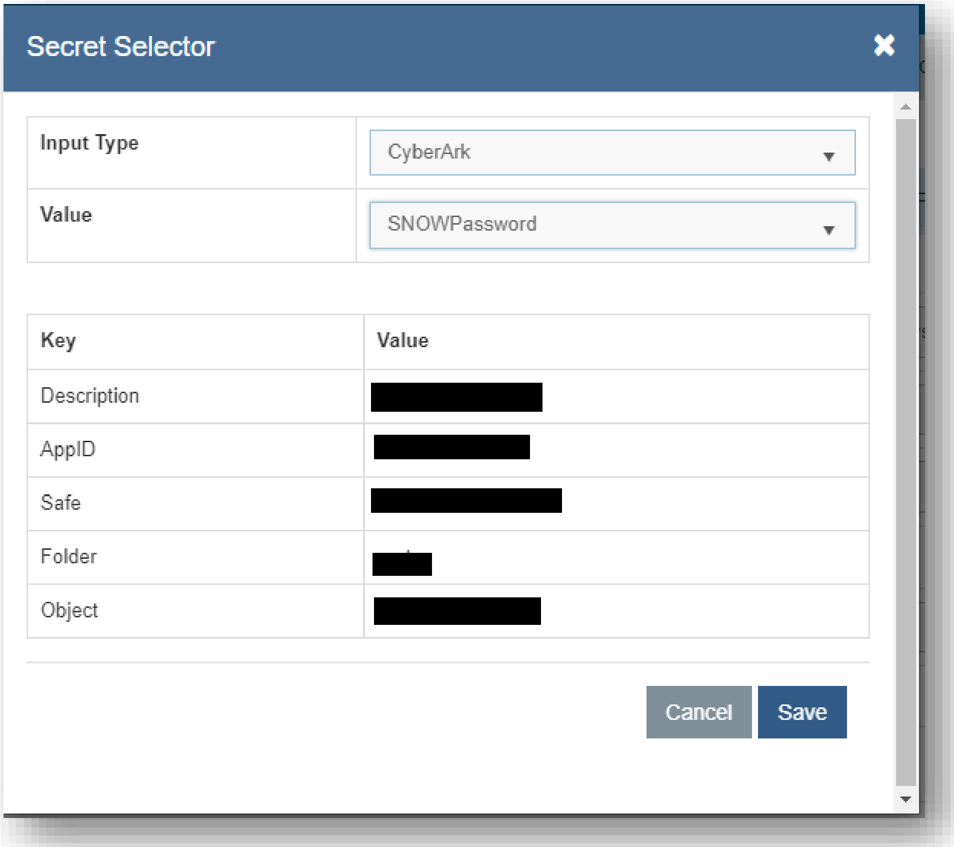
any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Secret Selector

Input Type	Input text
Value	•••••••••••••••••• □Show Password

Figure 129 – Password in plaintext



Secret Selector

Input Type	CyberArk
Value	SNOWPassword

Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

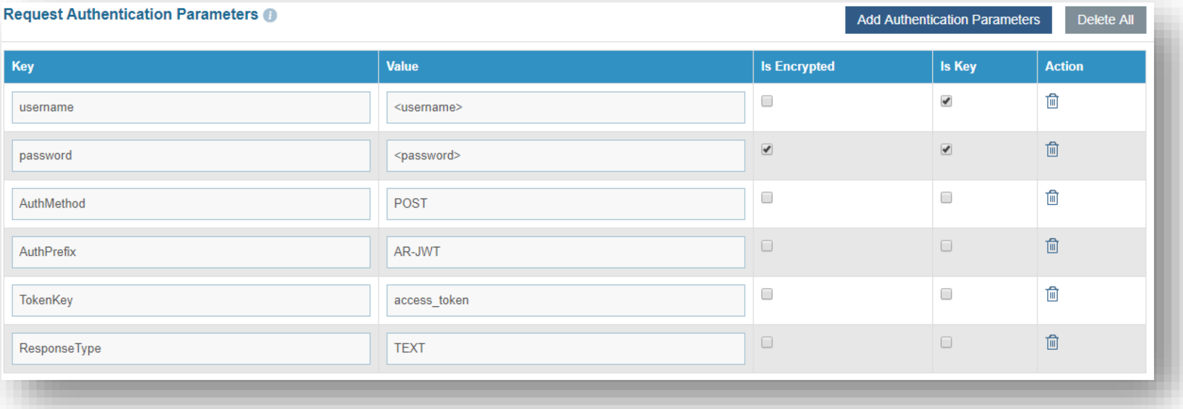
Figure 130 – Password from Key Vault (CyberArk)

- **Request Authentication Parameters** – If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.
- Based on the **Authentication Type**, add the parameters mentioned in the below table.

Table 24– Sample Authentication Parameters– Change Request Task

Authentication Type	Key	Value	Is Encrypted?	Is Key?
JWT	username	<username>	NO	YES
JWT	password	<password>	YES	YES
JWT	AuthMethod	POST	NO	NO
JWT	AuthPrefix	AR-JWT	NO	NO
JWT	TokenKey	access_token	NO	NO
JWT	ResponseType	TEXT	NO	NO
OAuth2.0	username	<username	NO	YES
OAuth2.0	password	<password>	YES	YES
OAuth2.0	AuthMethod	POST	NO	NO
OAuth2.0	AuthPrefix	Bearer	NO	NO

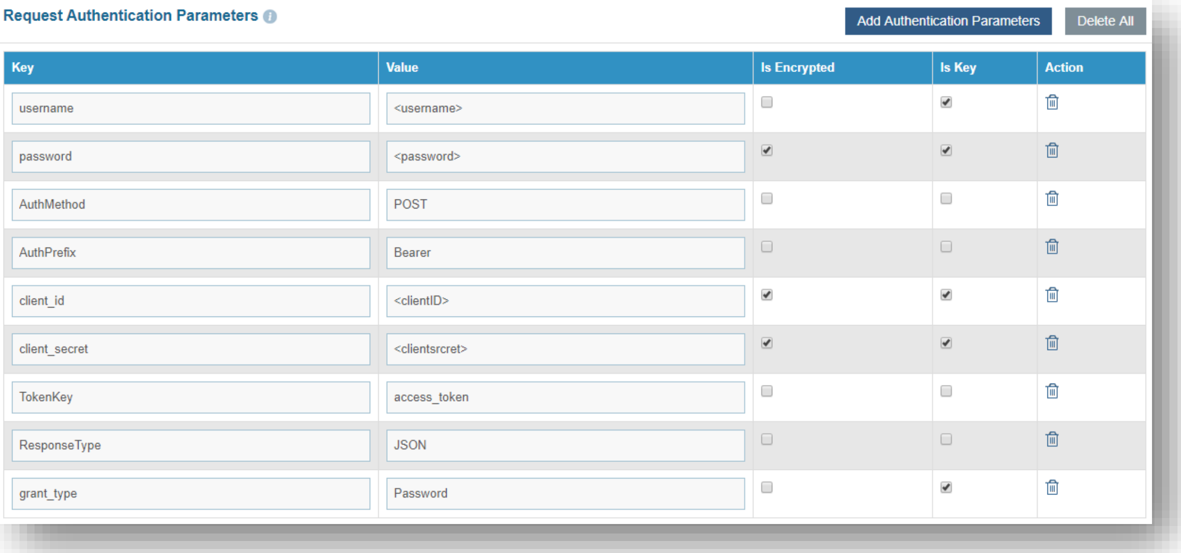
Authentication Type	Key	Value	Is Encrypted?	Is Key?
OAuth2.0	client_id	<clientID>	YES	YES
OAuth2.0	client_secret	<clientsrcret>	YES	YES
OAuth2.0	TokenKey	access_token	NO	NO
OAuth2.0	ResponseType	JSON	NO	NO
OAuth2.0	grant_type	Password	NO	YES



Request Authentication Parameters

Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	AR-JWT	☐	☐	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	TEXT	☐	☐	🗑

Figure 131 – Request Authentication Parameters for JWT



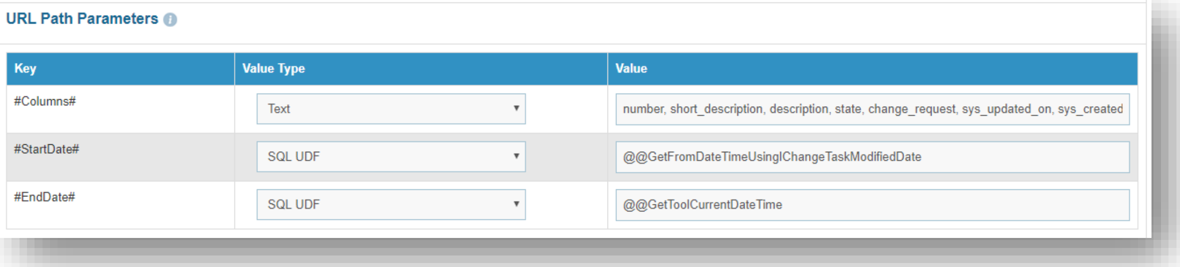
Request Authentication Parameters

Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	Bearer	☐	☐	🗑
client_id	<clientID>	☑	☑	🗑
client_secret	<clientsrcret>	☑	☑	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	JSON	☐	☐	🗑
grant_type	Password	☐	☑	🗑

Figure 132 – Change Request Task (Request Authentication Parameters for OAuth2.0)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #Columns#

ValueType: Text

Value:

number, short_description, description, state, change_request,
sys_updated_on, sys_created_on


Note – These columns are mandatory. User can add more columns if
more data is required to be fetched from ITSM tool.


Key: #StartDate#

ValueType: SQL UDF

VALUE: @@GetFromDateTimeUsingIChangeTaskModifiedDate


Key: #EndDate#

ValueType: SQL UDF

VALUE: @@GetToolCurrentDateTime
```



URL Path Parameters

Key	Value Type	Value
#Columns#	Text	number, short_description, description, state, change_request, sys_updated_on, sys_created
#StartDate#	SQL UDF	@@GetFromDateTimeUsingIChangeTaskModifiedDate
#EndDate#	SQL UDF	@@GetToolCurrentDateTime

Figure 133 – URL Parameters (Change Request Task)

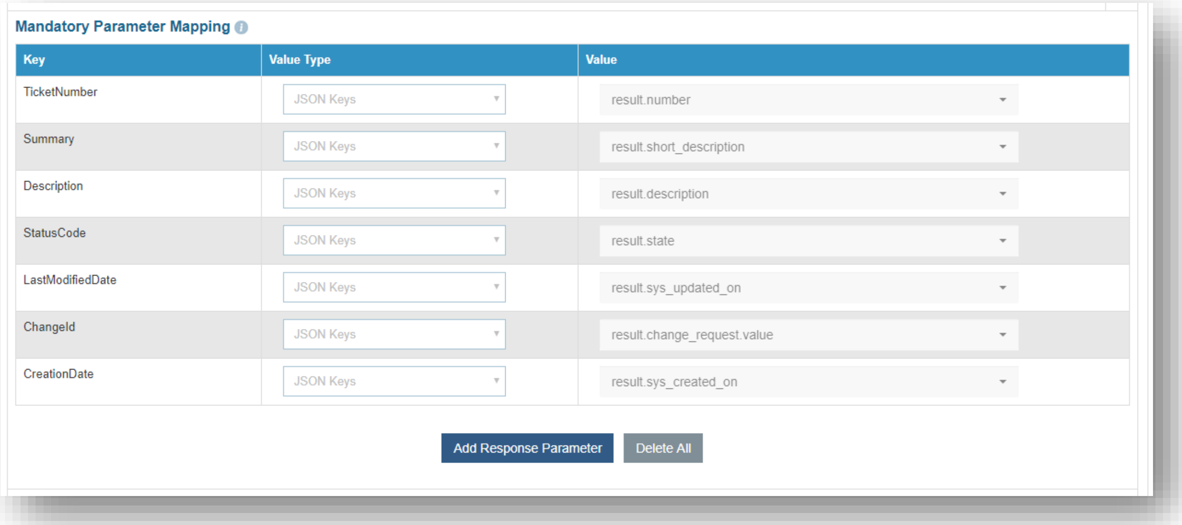
- **Request Header Parameters –** Please enter the request header parameters as required.
- **Response Body –** In this section, please enter the output of URL query for one of the incidents in JSON format. A sample response is mentioned below:

```
Response Body –

{
   "result": {"sys_updated_on": "2018-03-18 13:59:04","number":
"CTASK0039760","sys_created_on":

2018-03-18 13:59:02","state": "1", "short_description":
"Implementation Task", "description": "Please initiate the
Implementation process.","sys_id":
"d612a2a34ff85b40b2627d918110c7ef”,

"change_request": {"link": "https://hclgbpdev.service-
now.com/api/now/v1/table/change_request/c6c12e634ff85b40b2627d9181
10c724","value": "c6c12e634ff85b40b2627d918110c724" } }

}
```

- After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section**.**
- **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below.

Table 25– Sample Mandatory Mapping Parameters– Change Request Task

Key	Value Type	Value
TicketNumber	JSON.Keys	result.number
Summary	JSON.Keys	result.short_description
Description	JSON.Keys	result.description
StatusCode	JSON.Keys	result.state
LastModifiedDate	JSON.Keys	result.sys_updated_on
ChangeId	JSON.Keys	result.change_request.value
CreationDate	JSON.Keys	result.sys_created_on



Mandatory Parameter Mapping

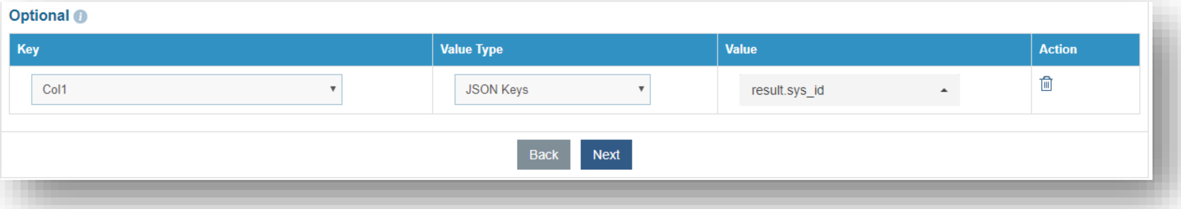
Key	Value Type	Value
TicketNumber	JSON Keys	result.number
Summary	JSON Keys	result.short_description
Description	JSON Keys	result.description
StatusCode	JSON Keys	result.state
LastModifiedDate	JSON Keys	result.sys_updated_on
ChangeId	JSON Keys	result.change_request.value
CreationDate	JSON Keys	result.sys_created_on

Figure 134 – Mandatory Parameter Mapping (Change Request Task)

– If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 26 – Sample Optional Mapping Parameters– Change Request Task

Key	Value Type	Value
Col1	JSON.Keys	result.sys_id



Optional

Key	Value Type	Value	Action
Col1	JSON Keys	result.sys_id	🗑

Figure 135 – Optional Parameter Mapping (Change Request Task)

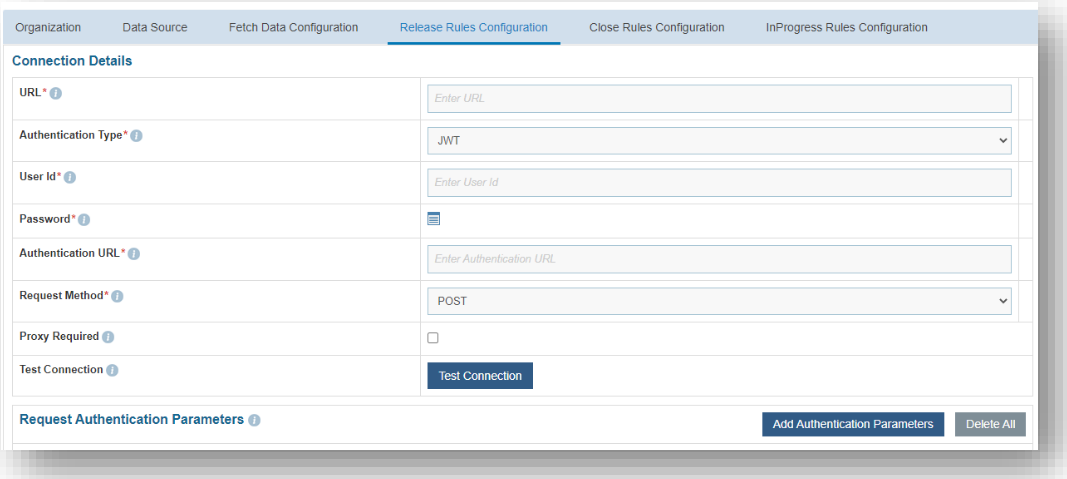
– Click Next to proceed to Release Rules Configuration.

– On **Release Rules Configuration** tab, type in the details as per the requirement.

– In the **Connection Details** section, enter the following details:

- **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.

- **Sample URL** - https://<url>.service-now.com/api/now/table/change_task/#incident#
- **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously.
- **Request Method** – Select Request Method as PUT from the drop-down.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.


Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration
Test Connection
Add Authentication Parameters
Delete All

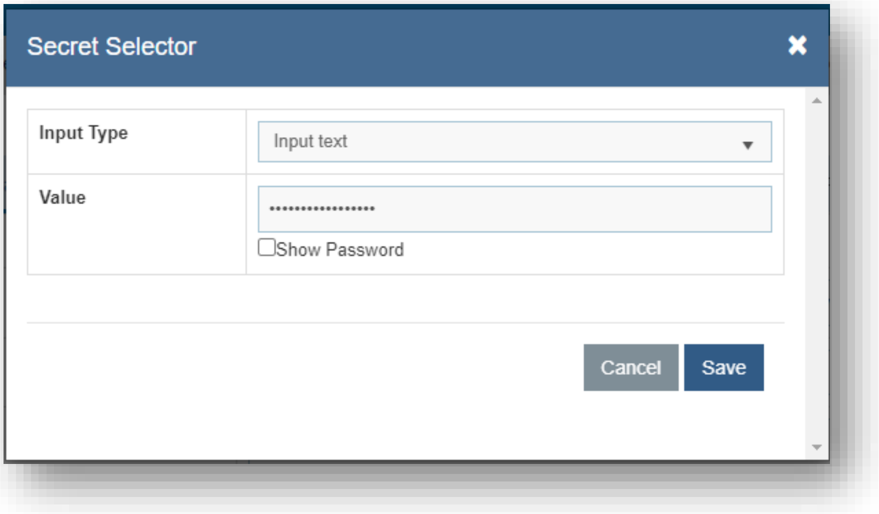
Connection Details

URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	▤
Authentication URL*	Enter Authentication URL
Request Method*	POST
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

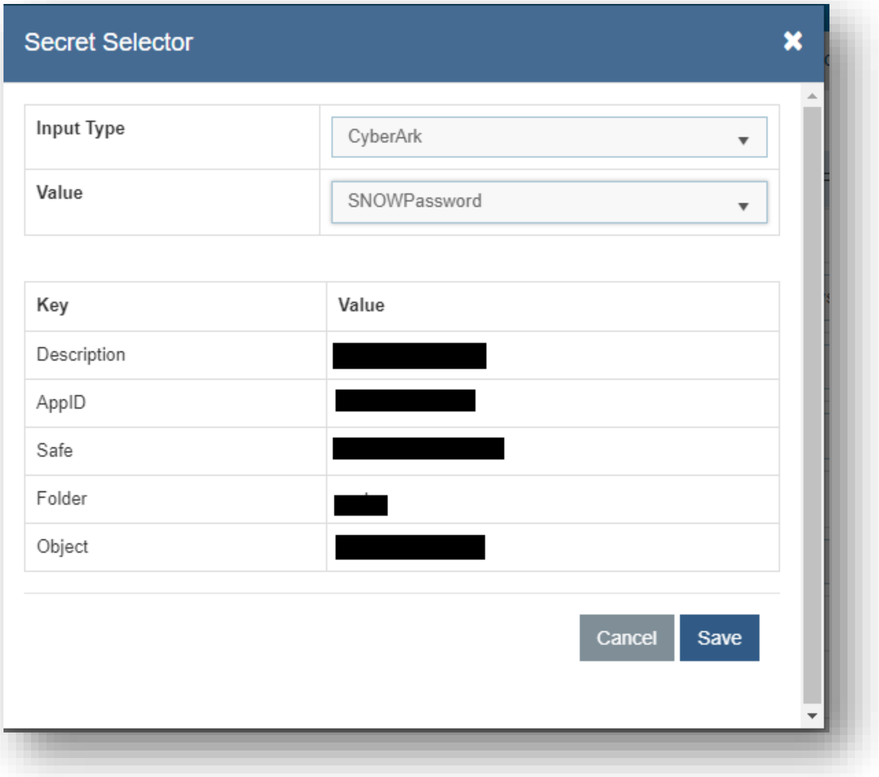
Figure 136 – Release Rules Configuration – Change Request Task (Connection Details)

- **Password** – For password, click on icon next to it. If the password is available in plaintext then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Input Type	Input text
Value	••••••••••••••••• □Show Password

Figure 137 – Password in plaintext



Input Type	CyberArk
Value	SNOWPassword

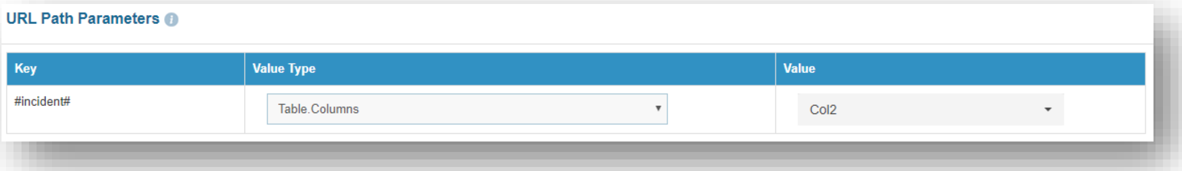
Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 138 – Password from Key Vault (CyberArk)

- **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #incident#
```

```
ValueType: Table Columns

Value:

Select from dropdown that mapped to sys_id from previous screen
"Col2"
```

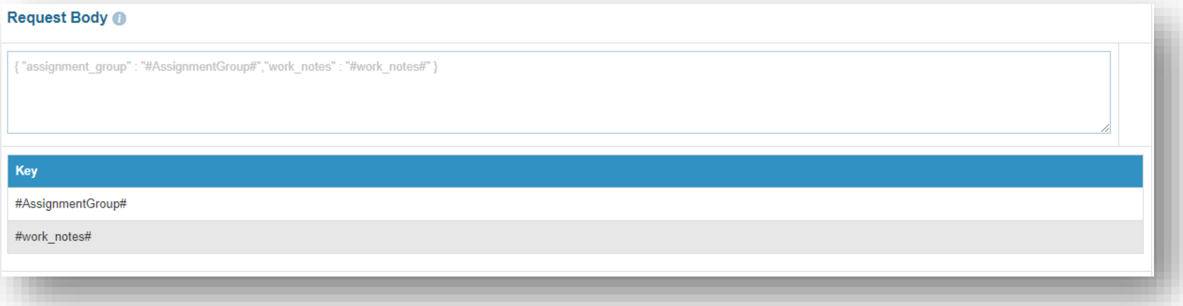

URL Path Parameters

Key	Value Type	Value
#incident#	Table.Columns	Col2

Figure 139 – Release Rules Configuration – Change Request Task (URL Path Parameters)

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body –** In this section, please enter the request body in JSON format. A sample request is mentioned below:

```
Request Body –

{ "assignment_group" : "#AssignmentGroup#","work_notes" :
"#work_notes#" }
```


Request Body

{ "assignment_group" : "#AssignmentGroup#","work_notes" : "#work_notes#" }

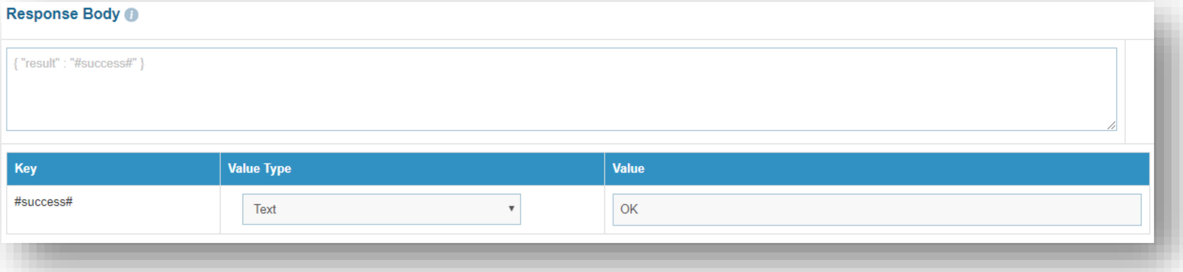
Key
#AssignmentGroup#
#work_notes#

Figure 140 – Release Rules Configuration – Change Request Task (Request Body)

- **Response Body –** In this section, please enter the response body in JSON format. A sample response is mentioned below.

```
Response Body –
```

```
{ "result" : "#success#" }
```



Response Body

```
{ "result" : "#success#" }
```

Key	Value Type	Value
#success#	Text	OK

Figure 141 – Release Rules Configuration – Change Request Task (Response Body)

– **Response Key Value** mapping can be done as per the below table.

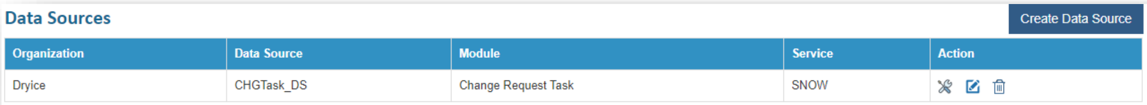
Table 27– Sample Response Key Value Mapping Parameters– Change Request Task

#success#	Text	OK

– Click **Submit** to add the data source.

– In order to bring the tickets within BigFix Runbook AI scope, a specific queue needs to be configured in the ITSM tool and same has to be configured in BigFix Runbook AI. This is achieved through **Manage Entry Criteria**. Please perform the below steps:

- Go to Actions Tab and click Manage Data Sources.
- On the **Data Sources** tab, click ✂ next to the data source user wants to manage. **Manage Entry Criteria** screen appears.

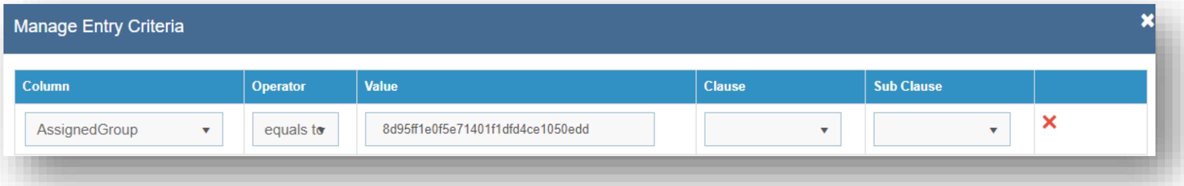


Data Sources

Organization	Data Source	Module	Service	Action
Dryice	CHGTask_DS	Change Request Task	SNOW	✂ ✎ 🗑

Figure 142 – Manage Entry Criteria (Change Request Task)

- Select 'AssignedGroup' for the **Column** field and 'equals to' for the **Operator** field.
- Enter the sys_id of the assignment group in ServiceNow in the **Value** field.
- **Clause** and **Sub-Clause** fields can also be added based on requirement.



Manage Entry Criteria

Column	Operator	Value	Clause	Sub Clause	
AssignedGroup	equals to	8d95ff1e0f5e71401f1dfd4ce1050edd			✕

Figure 143 – Manage Entry Criteria – Change Request Task (cont.)

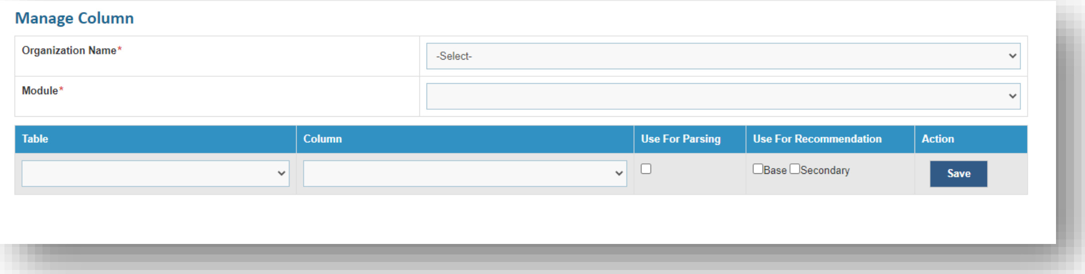
- Click **Save**.

4.2.3.3 Configuration of additional parameters for Recommendation and Parsing

To use the field values of Change Request for the purpose of Recommendation and Parsing by BigFix Runbook AI services, they need to be mapped to Change Request Task.

To do so, perform the following steps -

– On the main menu bar, click Advance Configuration → Parameter → Manage Column.



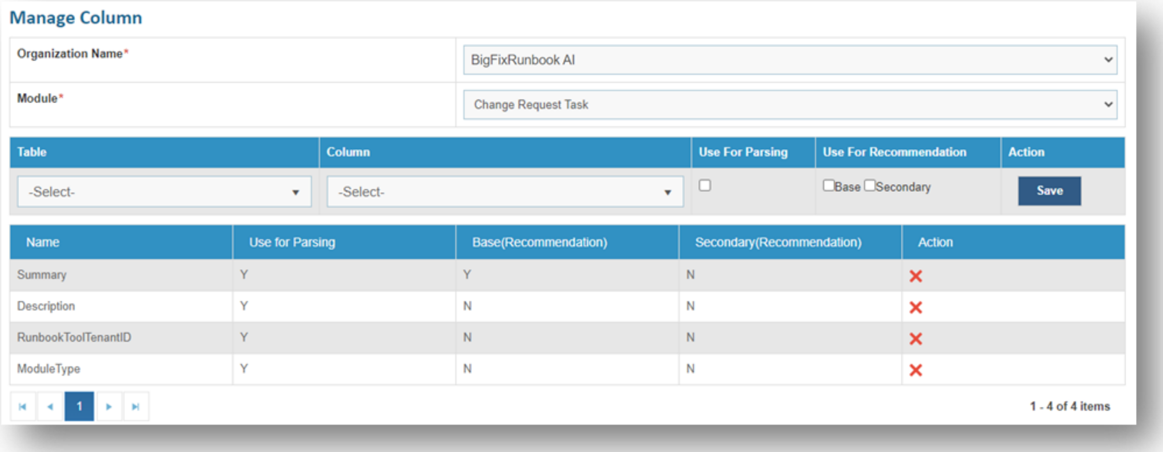
Manage Column

Organization Name*	-Select-
Module*	

Table	Column	Use For Parsing	Use For Recommendation	Action
		□	□Base □Secondary	Save

Figure 144 – Map fields of Change Request to Change Request Task

– Select **Organization Name** from dropdown. Select 'Change Request Task' as the **Module**.



Manage Column

Organization Name*	BigFixRunbook AI
Module*	Change Request Task

Table	Column	Use For Parsing	Use For Recommendation	Action
-Select-	-Select-	□	□Base □Secondary	Save

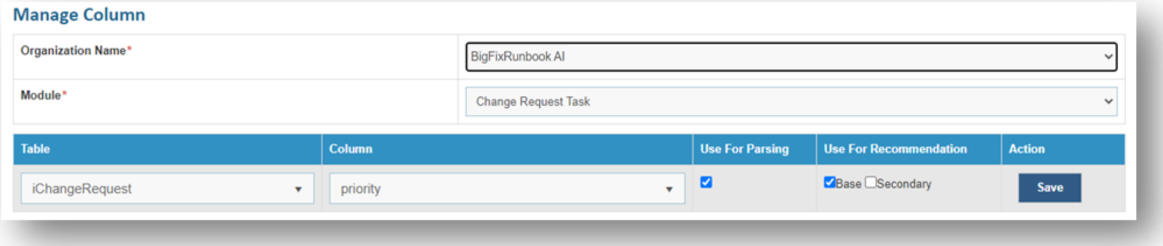
Name	Use for Parsing	Base(Recommendation)	Secondary(Recommendation)	Action
Summary	Y	Y	N	✕
Description	Y	N	N	✕
RunbookToolTenantID	Y	N	N	✕
ModuleType	Y	N	N	✕

1 - 4 of 4 items

Figure 145 – Map fields of Change Request to Change Request Task (cont.)

Note - Summary, Description, RunbookToolTenantID, ModuleType are the default entries.

- Select ‘iChangeRequest’ in **Table** dropdown.
- Select the column of Change Request which has to be mapped to Change Request in the **Column** dropdown. In this case, we are selecting ‘priority’.
- Check the fields **Use For Parsing** and select ‘Base’ for **Use For Recommendation** field.



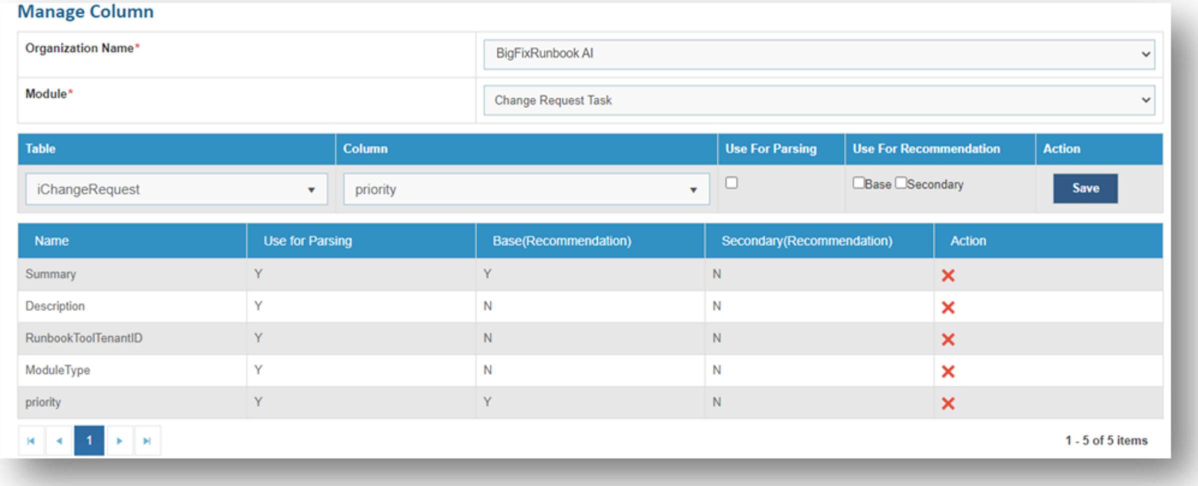
Manage Column

Organization Name*	BigFixRunbook AI
Module*	Change Request Task

Table	Column	Use For Parsing	Use For Recommendation	Action
iChangeRequest	priority	☑	☑Base □Secondary	Save

Figure 146 – Map fields of Change Request to Change Request Task (cont.)

- Click **Save.** The page lists one additional entry i.e. ‘priority’, as depicted below.



Manage Column

Organization Name*	BigFixRunbook AI
Module*	Change Request Task

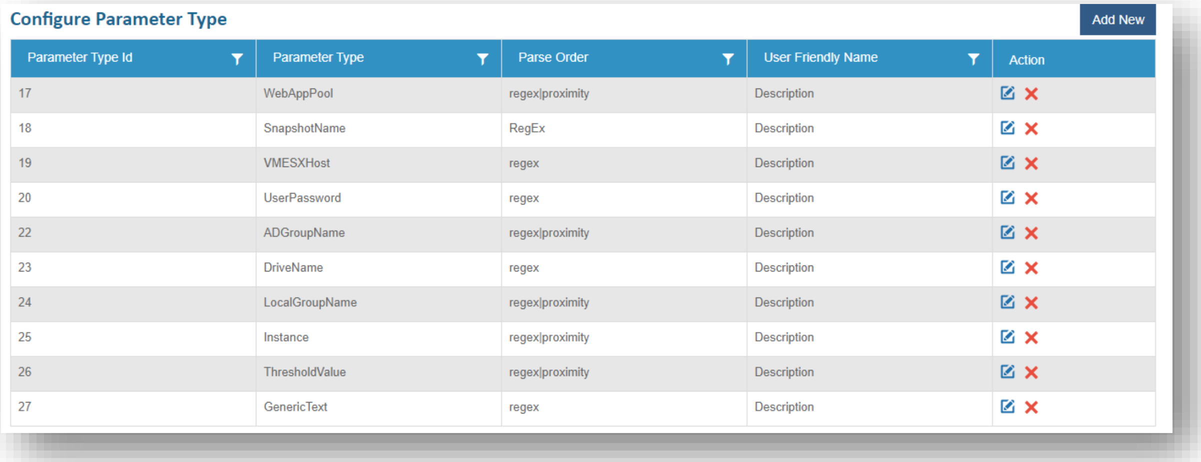
Table	Column	Use For Parsing	Use For Recommendation	Action
iChangeRequest	priority	□	□Base □Secondary	Save

Name	Use for Parsing	Base(Recommendation)	Secondary(Recommendation)	Action
Summary	Y	Y	N	✕
Description	Y	N	N	✕
RunbookToolTenantID	Y	N	N	✕
ModuleType	Y	N	N	✕
priority	Y	Y	N	✕

1 - 5 of 5 items

Figure 147 – Map fields of Change Request to Change Request Task (cont.)

- For Recommendation, above steps are sufficient. But for Parsing, additional steps are required to be performed.
- On the main menu bar, click **Environment.**
- Click **Configure Parameter Type**. By default, there are several entries already defined.



Configure Parameter Type

Parameter Type Id	Parameter Type	Parse Order	User Friendly Name	Action
17	WebAppPool	regex|proximity	Description	✎ ✕
18	SnapshotName	RegEx	Description	✎ ✕
19	VMESXHost	regex	Description	✎ ✕
20	UserPassword	regex	Description	✎ ✕
22	ADGroupName	regex|proximity	Description	✎ ✕
23	DriveName	regex	Description	✎ ✕
24	LocalGroupName	regex|proximity	Description	✎ ✕
25	Instance	regex|proximity	Description	✎ ✕
26	ThresholdValue	regex|proximity	Description	✎ ✕
27	GenericText	regex	Description	✎ ✕

Figure 148 – Map fields of Change Request to Change Request Task (cont.)

- Click **Add New**.

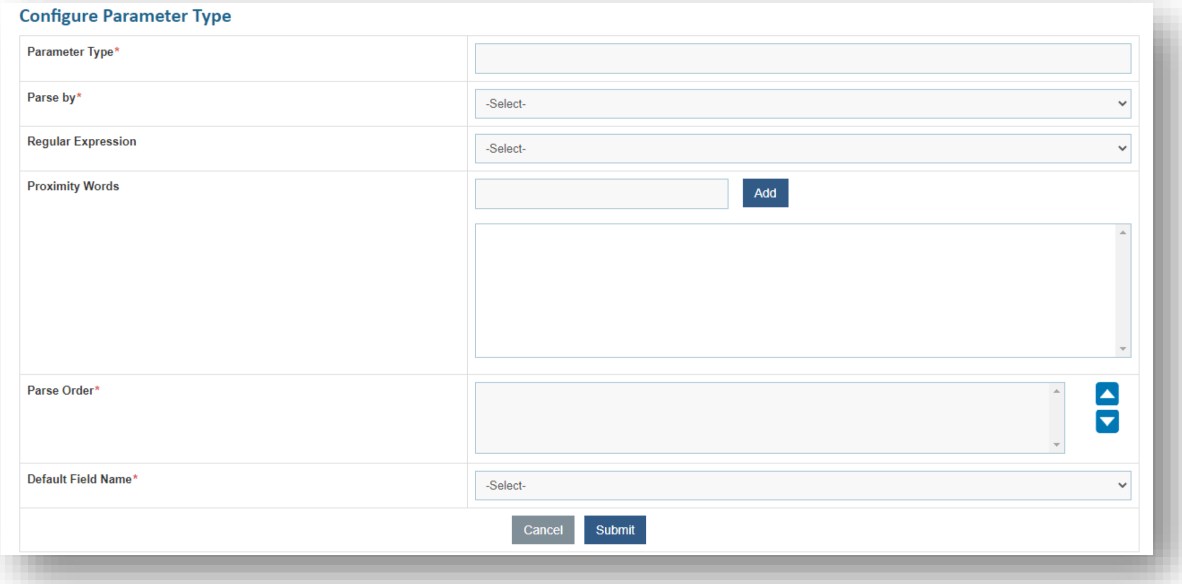
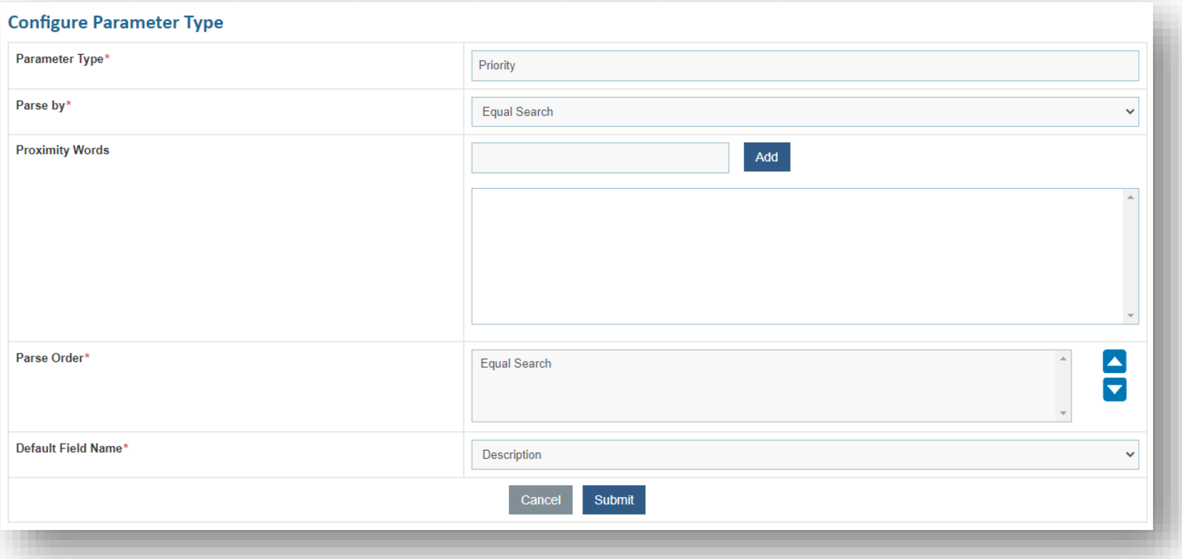


Figure 149 – Map fields of Change Request to Change Request Task (cont.)

- Type **Parameter Type**, for e.g. Priority
- Select 'Equal Search' as **Parse By**.
- Select 'Description' as **Default Field Name**.
- Click **Submit**.

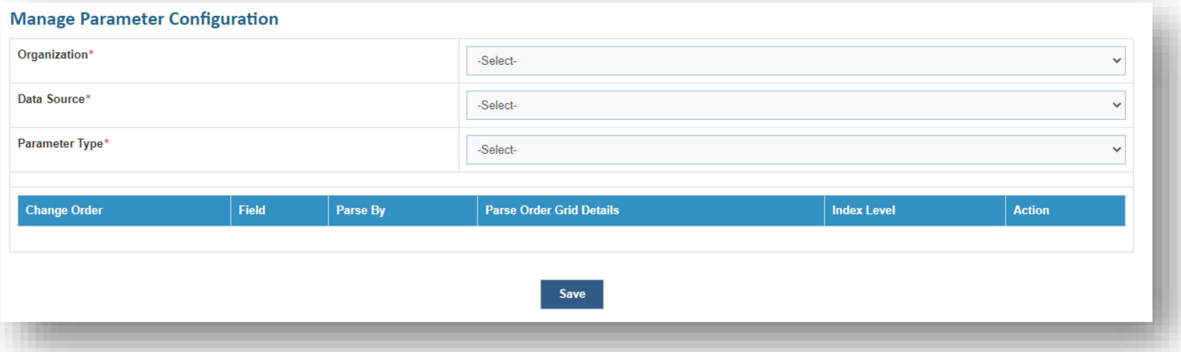


Configure Parameter Type

Parameter Type*	Priority
Parse by*	Equal Search
Proximity Words	
Parse Order*	Equal Search
Default Field Name*	Description

Figure 150 – Map fields of Change Request to Change Request Task (cont.)

- Next step is to map this **Parameter Type** ‘Category’, to the one that was created via **Manage Columns** in earlier step by the name **priority**. To do that, perform the following steps:
- On the main menu bar, click **Organization.**
- Click Manage Parameter Configuration.



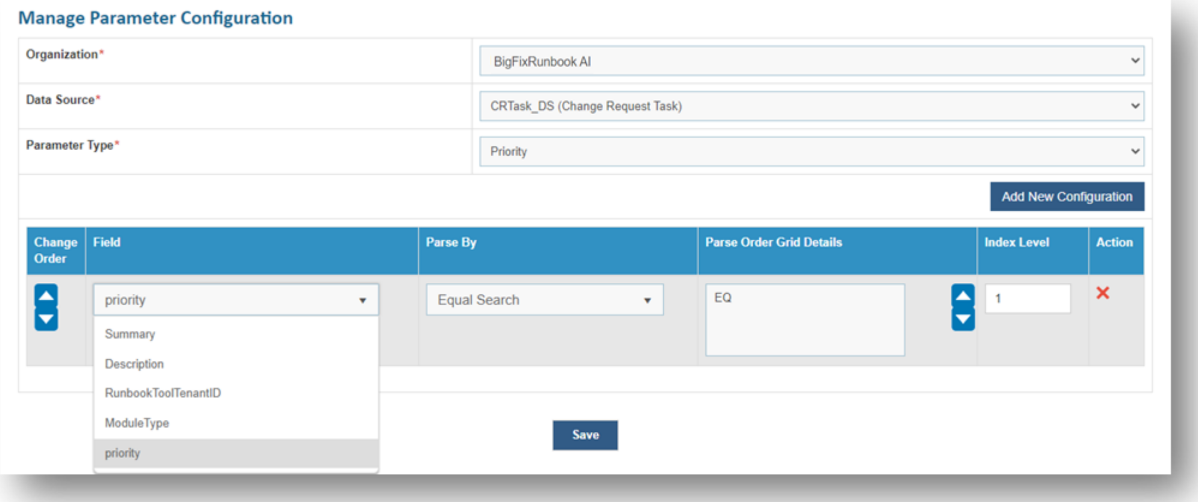
Manage Parameter Configuration

Organization*	-Select-
Data Source*	-Select-
Parameter Type*	-Select-

Change Order	Field	Parse By	Parse Order Grid Details	Index Level	Action

Figure 151 – Map fields of Change Request to Change Request Task (cont.)

- Selection **Organization.** Select ‘Change Request Task’ as the **Data Source.**
- Select the newly created parameter ‘Priority’ from **Parameter Type** dropdown**.**
- From the **Field** dropdown, select ‘priority’**,** the parameter that has been mapped via **Manage Columns.**



Manage Parameter Configuration

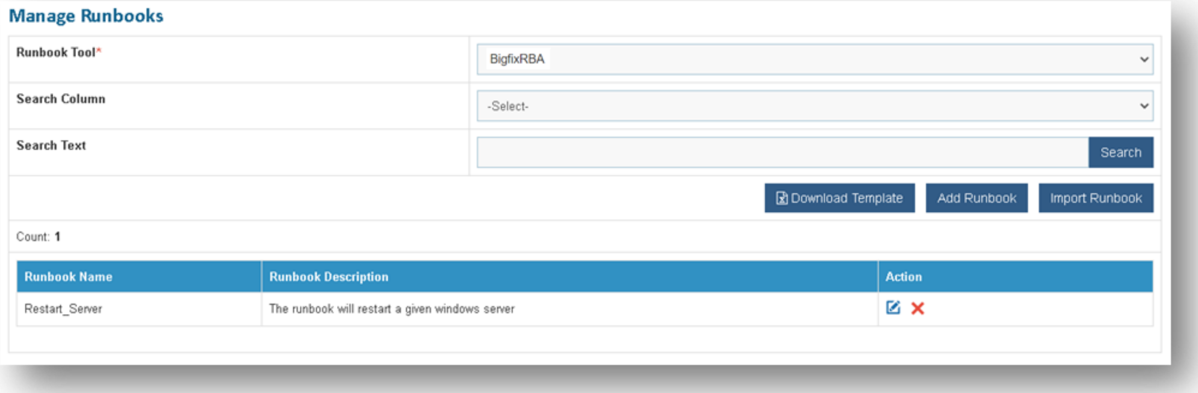
Organization*	BigFixRunbook AI
Data Source*	CRTask_DS (Change Request Task)
Parameter Type*	Priority

Change Order	Field	Parse By	Parse Order Grid Details	Index Level	Action
	priority	Equal Search	EQ	1	✕

Figure 152 – Map fields of Change Request to Change Request Task (cont.)

- Click **Save**.
- To verify whether this parameter is successfully parsed or not, perform the following steps:

- On the main menu bar, click **Runbooks**.
- Click Manage Runbooks.
- Select the **Runbook Tool** mapped with the organization.



Manage Runbooks

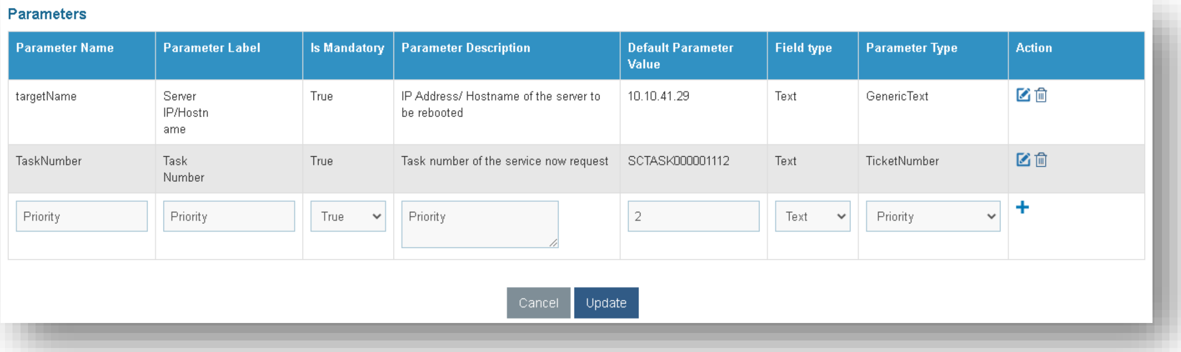
Runbook Tool*	BigfixRBA
Search Column	-Select-
Search Text	Search

Count: 1

Runbook Name	Runbook Description	Action
Restart_Server	The runbook will restart a given windows server	✎ ✕

Figure 153 – Map fields of Change Request to Change Request Task (cont.)

- The parameter, **Priority**, which was created in earlier steps, has to be added as one of the parameters to the existing runbook. You can also create a new runbook with **Priority** as one of the parameters.
- Click the **Edit** icon to edit the runbook.
- In the Parameters section, add a new parameter with any relevant **Parameter Name**, **Parameter Label**, **Parameter Description**, **Default Parameter Value**. Ensure that Parameter Type is selected as **Priority.**



Parameters

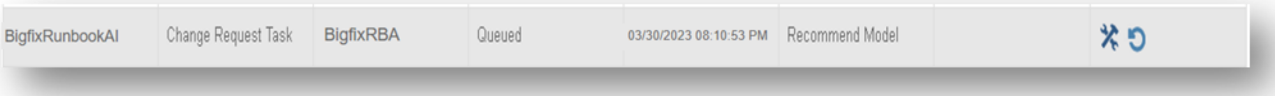
Parameter Name	Parameter Label	Is Mandatory	Parameter Description	Default Parameter Value	Field type	Parameter Type	Action
targetName	Server IP/Hostn ame	True	IP Address/ Hostname of the server to be rebooted	10.10.41.29	Text	GenericText	✎ 🗑
TaskNumber	Task Number	True	Task number of the service now request	SCTASK000001112	Text	TicketNumber	✎ 🗑
Priority	Priority	True	Priority	2	Text	Priority	+

Figure 154 – Map fields of Change Request to Change Request Task (cont.)

- Add the parameter and click **Update**.
- Ensure that the runbook in which the parameter is added is mapped with the organization.

– Next step is to build the Recommendation model and to do that perform the following steps:

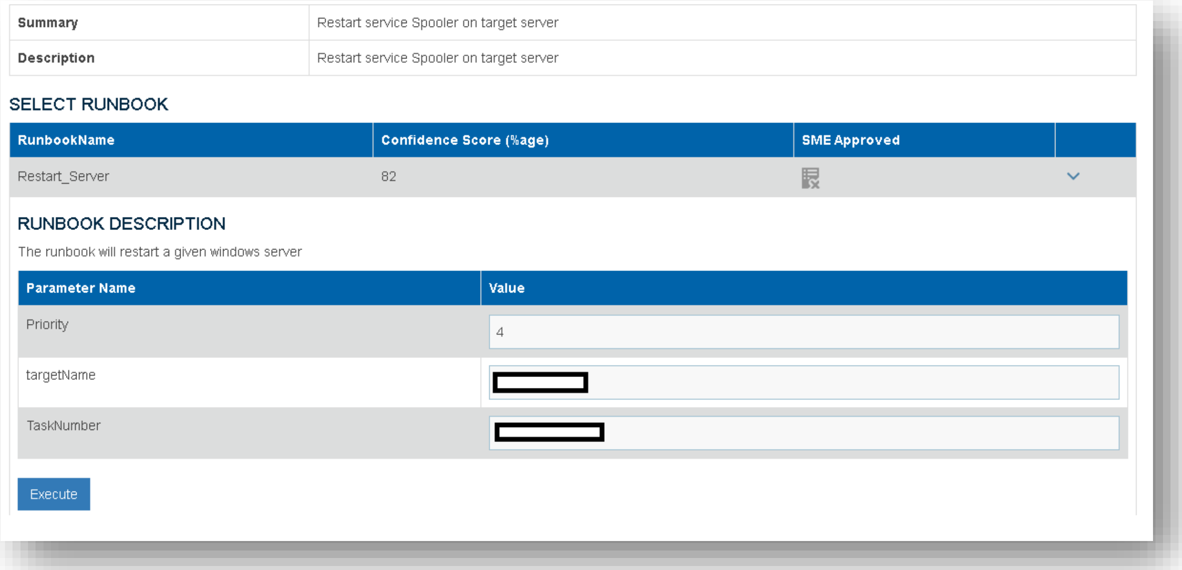
- On the main menu bar, click **Actions Tab → Runbooks.**
- Click Build Model.
- ReBuild / Re-build the model for the Organization under Change Request Task module for the mapped runbook tool.



BigfixRunbookAI	Change Request Task	BigfixRBA	Queued	03/30/2023 08:10:53 PM	Recommend Model		

Figure 155 – Map fields of Change Request to Change Request Task (cont.)

- Run the entire flow and see if the runbook recommended for the ticket in which the parameter was added has the parameter **Priority** with its expected value.



Summary	Restart service Spooler on target server
Description	Restart service Spooler on target server

SELECT RUNBOOK

RunbookName	Confidence Score (%age)	SME Approved	
Restart_Server	82		

RUNBOOK DESCRIPTION

The runbook will restart a given windows server

Parameter Name	Value
Priority	4
targetName	[redacted]
TaskNumber	[redacted]

Figure 156 – Map fields of Change Request to Change Request Task (cont.)

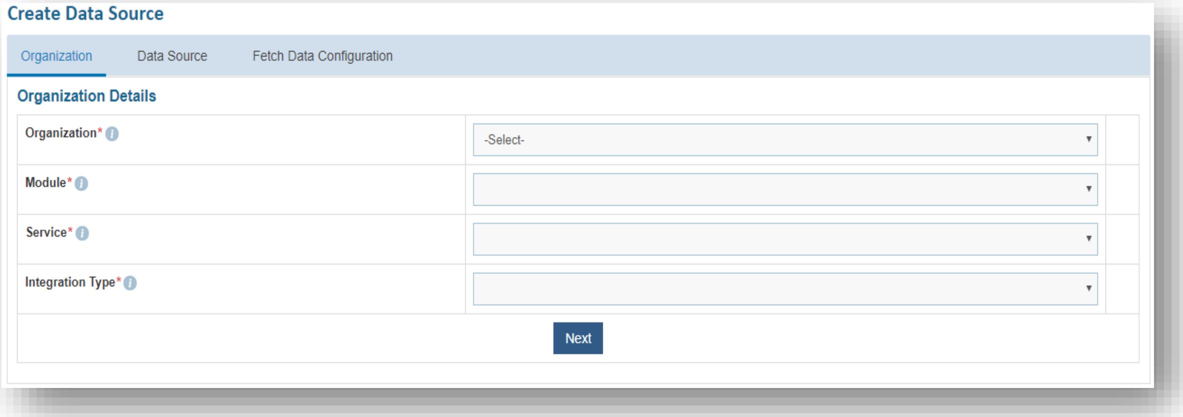
4.3 Integration with BMC Remedy

4.3.1 Incident Management

To create a data source for Incident Management, perform the following steps:

- On the main menu bar, click **Action tab → Manage Data Sources**.
- The **Create Data Source** page appears with the following tabs:
  - Organization

- Data Source
- Fetch Data Configuration
- Release Rules Configuration



Create Data Source

Organization Details

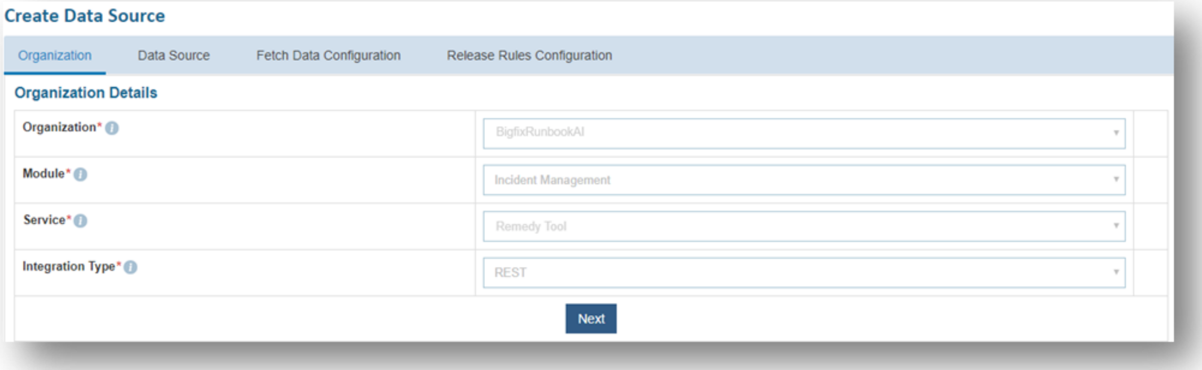
Organization*	-Select-
Module*	
Service*	
Integration Type*	

Figure 157 - Create Data Source

Release Rules Configuration is only applicable for the following **Module** types- **Incident Management, Change Request Task and Service Request Task.** This tab will not be activated for other module types.

– On the **Organization** tab:

- Select the **Organization Name** from the dropdown.
- Select the **Module** as **Incident Management** since we are configuring this data source for pulling the incident tickets.
- Select the **Service** as **Remedy Tool** as we are configuring the data source for BMC Remedy
- Select the **Integration Type** as **REST**, since we will be integrating through REST APIs.
- Click **Next**.



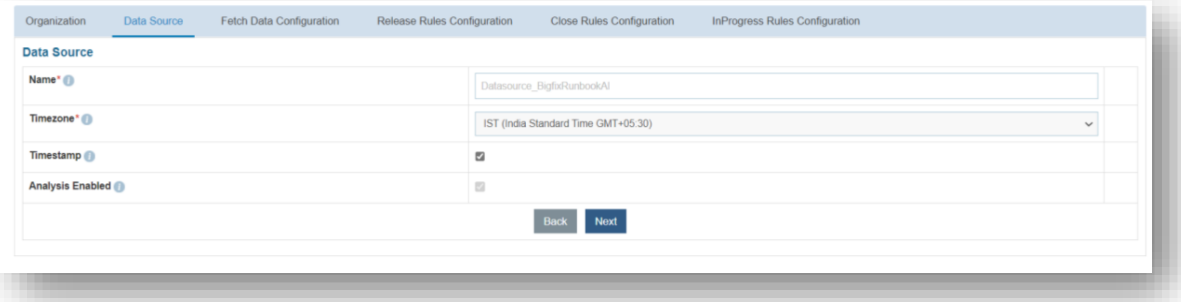
Organization Details

Organization*	BigfixRunbookAI
Module*	Incident Management
Service*	Remedy Tool
Integration Type*	REST

Figure 158 - Create Data Source (cont.)

– On the **Data Source** tab,

- Type the new data source in the **Name** field.
- Select the **Timezone** to specify the time zone of the selected data source.
- Select **Timestamp** to view the present data with date and time.
- Select **Analysis Enabled?** if user wants to analyze the data retrieved from the data source.
- Click Next.


Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration
Back
Next

Data Source

Name*	Datasource_BigfixRunbookAI
Timezone*	IST (India Standard Time GMT+05:30)
Timestamp	☑
Analysis Enabled	☑

Figure 159 - Create Data Source (cont.)

– On the **Fetch Data Configuration** tab, type in the details as per the environment.

– In the **Connection Details** section, enter the following details:

- **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.
- **Sample URL** - *http://URL/api/arsys/v1/entry/HPD:Help%20Desk/?q='Assigned Group'="#Group#" AND 'Last Modified Date'>"#StartDate#" AND 'Last Modified Date'<"#EndDAte#"&fields=values(#Columns#)*

- **Authentication Type** – Select one of the Authentication Types from Basic / Windows, JWT, OAuth 2.0

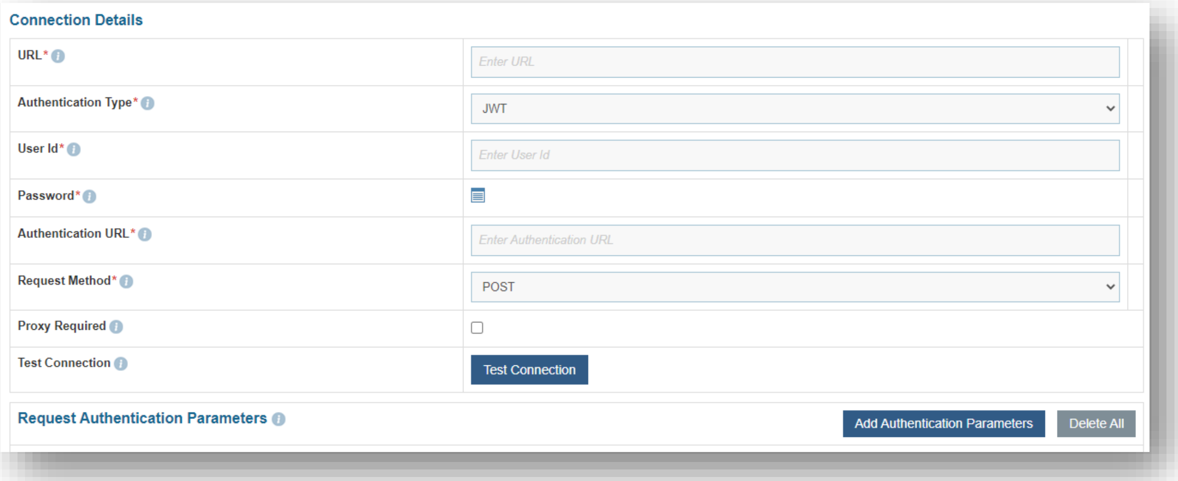
  Selection of **Basic / Windows** requires you to enter -

  - User Id
  - Password

  Selection of **JWT / OAuth 2.0** requires you to enter -

  - User Id
  - Password
  - Authentication URL

- Here, we will be using **JWT** as the **Authentication Type**.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



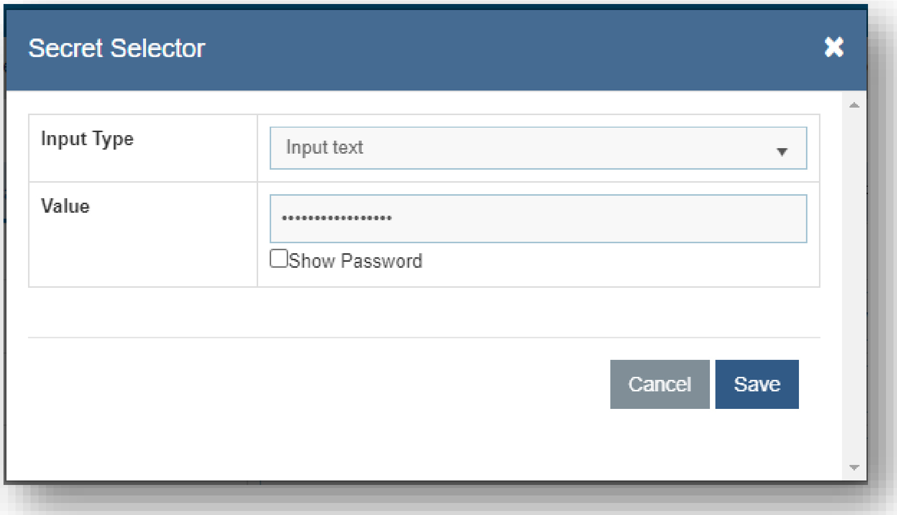
Connection Details

URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	▤
Authentication URL*	Enter Authentication URL
Request Method*	POST
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters
Add Authentication Parameters
Delete All

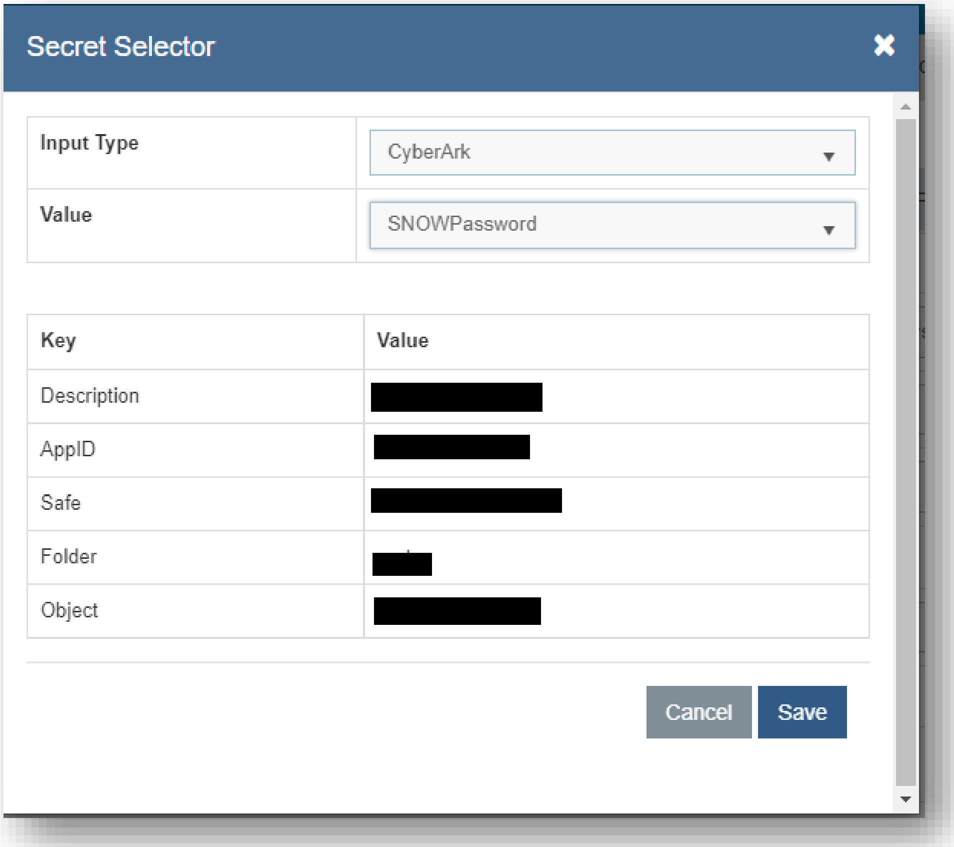
Figure 160 – Create Data Source (Connection Details)

- **Password** – For password, click on icon next to it. If the password is available in plaintext then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Input Type	Input text
Value	•••••••••••••••• □Show Password

Figure 161 – Password in plaintext



Input Type	CyberArk
Value	SNOWPassword

Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

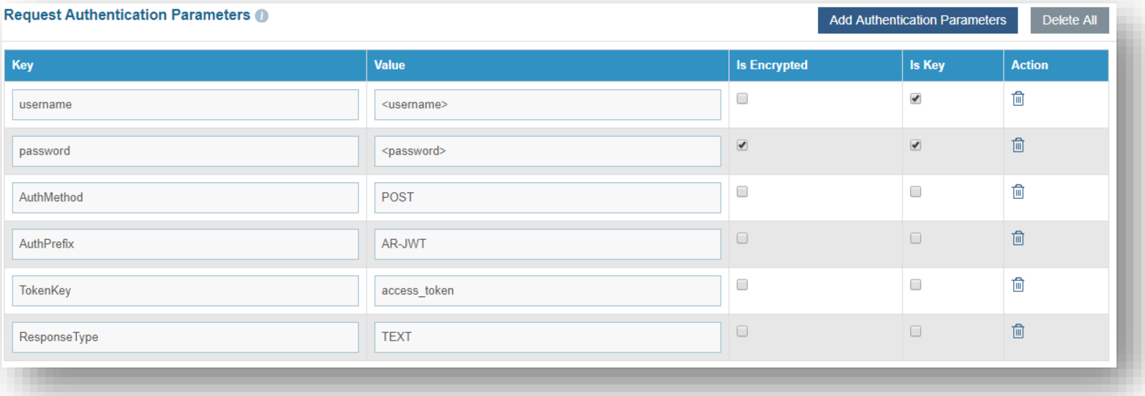
Figure 162 – Password from Key Vault (CyberArk)

- **Request Authentication Parameters** – If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.

- Based on the **Authentication Type**, add the parameters mentioned in the below table.

Table 28– Sample Authentication Parameters

Authentication Type	Key	Value	Is Encrypted?	Is Key?
JWT	username	<username>	NO	YES
JWT	password	<password>	YES	YES
JWT	AuthMethod	POST	NO	NO
JWT	AuthPrefix	AR-JWT	NO	NO
JWT	TokenKey	access_token	NO	NO
JWT	ResponseType	TEXT	NO	NO


Request Authentication Parameters
Add Authentication Parameters
Delete All

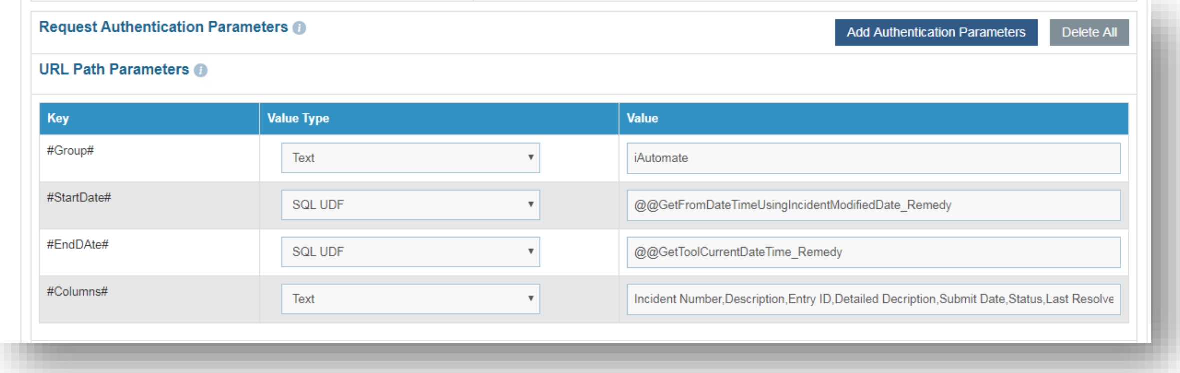
Key	Value	Is Encrypted	Is Key	Action
username	<username>	□	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	□	□	🗑
AuthPrefix	AR-JWT	□	□	🗑
TokenKey	access_token	□	□	🗑
ResponseType	TEXT	□	□	🗑

Figure 163 – Create Data Source (Request Authentication Parameters for JWT)

- **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #Columns#

ValueType: Text

Value:

Incident Number,Description,Entry ID,Detailed Decription,Submit
Date,Status,Last Resolved Date,Assigned Group, Last Modified
Date,Parent_SAP_ID,Fraud Alert No.


Note – These columns are mandatory. User can add more columns if
more data is required to be fetched from ITSM tool.
```

```
Key: #StartDate#
ValueType: SQL UDF
VALUE: @@GetFromDateTimeUsingIncidentModifiedDate_Remedy

Key: #EndDate#
ValueType: SQL UDF
VALUE: @@GetToolCurrentDateTime_Remedy
```


Request Authentication Parameters
Add Authentication Parameters
Delete All
URL Path Parameters

Key	Value Type	Value
#Group#	Text	iAutomate
#StartDate#	SQL UDF	@@GetFromDateTimeUsingIncidentModifiedDate_Remedy
#EndDAte#	SQL UDF	@@GetToolCurrentDateTime_Remedy
#Columns#	Text	Incident Number,Description,Entry ID,Detailed Decription,Submit Date,Status,Last Resolve

Figure 164 – URL Path Parameters (BMC Remedy – Incident Management)

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Response Body –** In this section, please enter the output of URL query for one of the incidents in JSON format. A sample response is mentioned below:

```
Response Body -
{
    "entries": [
        {
            "values": {
                "Incident Number": "INC000000695805",
                "Description": "Test ticket please ignore",
                "Entry ID": "INC000000454748",
                "Detailed Decription": "Test ticket please 
ignore",
                "Submit Date": "2018-12-06T16:43:52.000+0000",
                "Status": "Assigned",
                "Last Resolved Date": "dummy",
                "Assigned Group": "NOC",
                "Last Modified Date": "2018-12-
06T16:43:52.000+0000"
, "Fraud Alert No.": "67570898119"
, "Parent_SAP_ID": "102614"
            },
            "_links": {
                "self": [
                    {
                        "href": 
"http://remlex12:8008/api/arsys/v1/entry/HPD:Help%20Desk/INC000000
454748"
                    }
                ]
            }
        }
    ],
    "_links": {
        "next": [
            {
                "href": 
"http://remlex12:8008/api/arsys/v1/entry/HPD:Help%20Desk/?q=%27Ass
igned%20Group%27=%22NOC%22%20AND%20%27Last%20Modified%20Date%27%3E
```

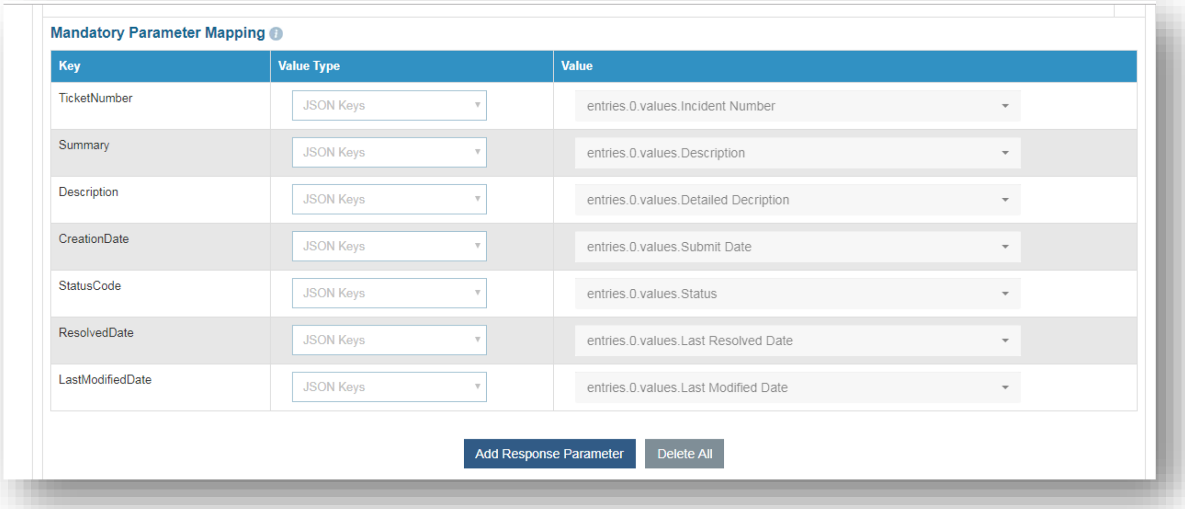
```
%222018-11-
01T15:48:00%22%20AND%20%27Last%20Modified%20Date%27%3C%222018-12-
07T15:48:00%22&offset=1&limit=1&fields=values(Incident%20Number,De
scription,Entry%20ID,Detailed%20Decription,Submit%20Date,Status,La
st%20Resolved%20Date,Assigned%20Group,%20Last%20Modified%20Date)"
            }
        ],
        "self": [
            {
                "href":
"http://remlex12:8008/api/arsys/v1/entry/HPD:Help%20Desk/?q=%27Ass
igned%20Group%27=%22NOC%22%20AND%20%27Last%20Modified%20Date%27%3E
%222018-11-
01T15:48:00%22%20AND%20%27Last%20Modified%20Date%27%3C%222018-12-
07T15:48:00%22&fields=values(Incident%20Number,Description,Entry%2
0ID,Detailed%20Decription,Submit%20Date,Status,Last%20Resolved%20D
ate,Assigned%20Group,%20Last%20Modified%20Date)&limit=1"
            }
        ]
    }
}
```

- After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section**.**
- **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below.

Table 29– Sample Mandatory Mapping Parameters

Key	Value Type	Value
TicketNumber	JSON.Keys	entries.0.values.Incident Number
Summary	JSON.Keys	entries.0.values.Description

Description	JSON.Keys	entries.0.values.Detailed Description
CreationDate	JSON.Keys	entries.0.values.Submit Date
StatusCode	JSON.Keys	entries.0.values.Status
ResolvedDate	JSON.Keys	entries.0.values.Last Resolved Date
LastModifiedDate	JSON.Keys	entries.0.values.Last Modified Date



Mandatory Parameter Mapping

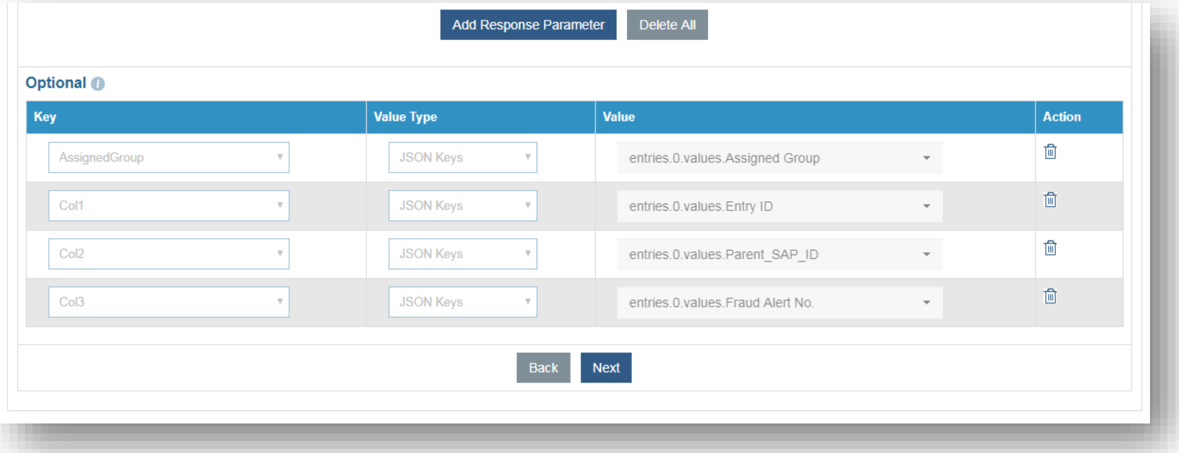
Key	Value Type	Value
TicketNumber	JSON Keys	entries.0.values.Incident Number
Summary	JSON Keys	entries.0.values.Description
Description	JSON Keys	entries.0.values.Detailed Decription
CreationDate	JSON Keys	entries.0.values.Submit Date
StatusCode	JSON Keys	entries.0.values.Status
ResolvedDate	JSON Keys	entries.0.values.Last Resolved Date
LastModifiedDate	JSON Keys	entries.0.values.Last Modified Date

Figure 165 – Mandatory Parameter Mapping

– If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 30– Sample Optional Mapping Parameters

Key	Value Type	Value
AssignedGroup	JSON.Keys	entries.0.values.Assigned Group
Col1	JSON.Keys	entries.0.values.Entry ID
Col2	JSON.Keys	entries.0.values.Parent_SAP_ID
Col3	JSON.Keys	entries.0.values.Fraud Alert No.

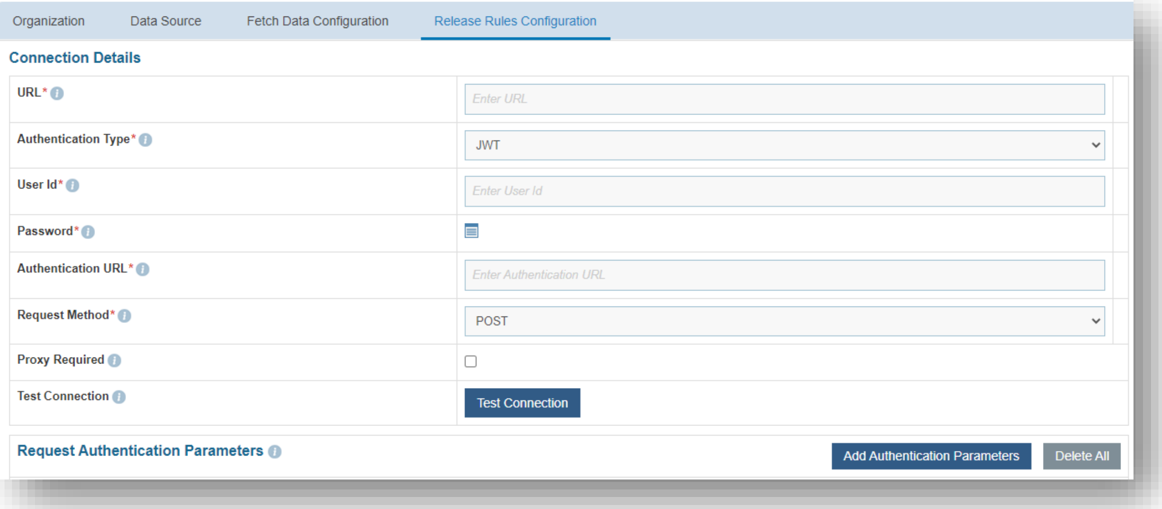


Optional

Key	Value Type	Value	Action
AssignedGroup	JSON Keys	entries.0.values.Assigned Group	🗑
Col1	JSON Keys	entries.0.values.Entry ID	🗑
Col2	JSON Keys	entries.0.values.Parent_SAP_ID	🗑
Col3	JSON Keys	entries.0.values.Fraud Alert No.	🗑

Figure 166 – Optional Parameter Mapping

- Click Next to proceed to Release Rules Configuration.
- On **Release Rules Configuration** tab, type in the details as per the requirement.
- In the **Connection Details** section, enter the following details:
  - **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.
  - **Sample URL** - http://URL/api/arsys/v1/entry/HPD:IncidentInterface/#TicketID#|#TicketID1#
  - **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously.
  - Request Method – Select Request Method as PUT from the drop-down.
  - **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
  - Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



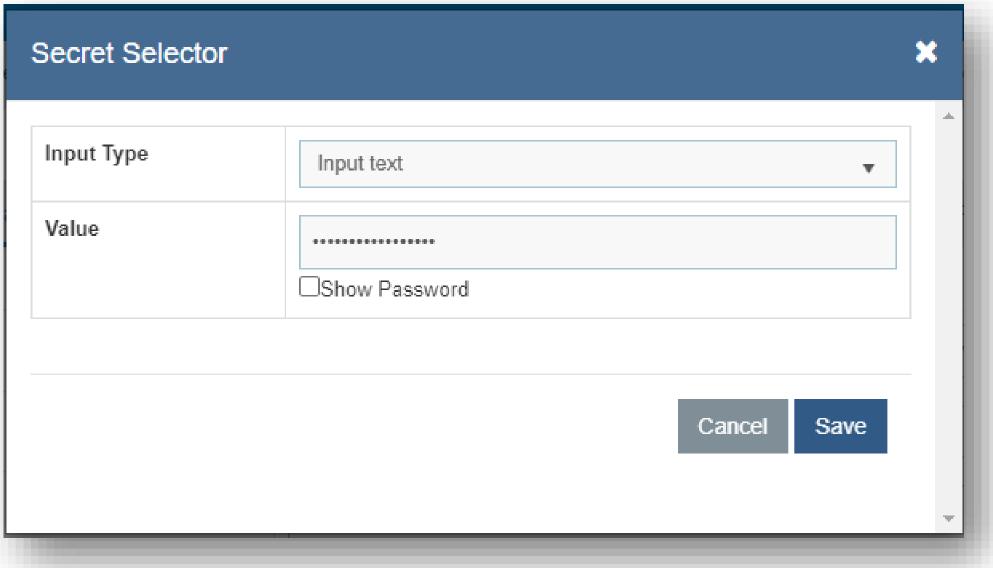
Connection Details

URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	POST
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

Figure 167 – Create Data Source (Connection Details)

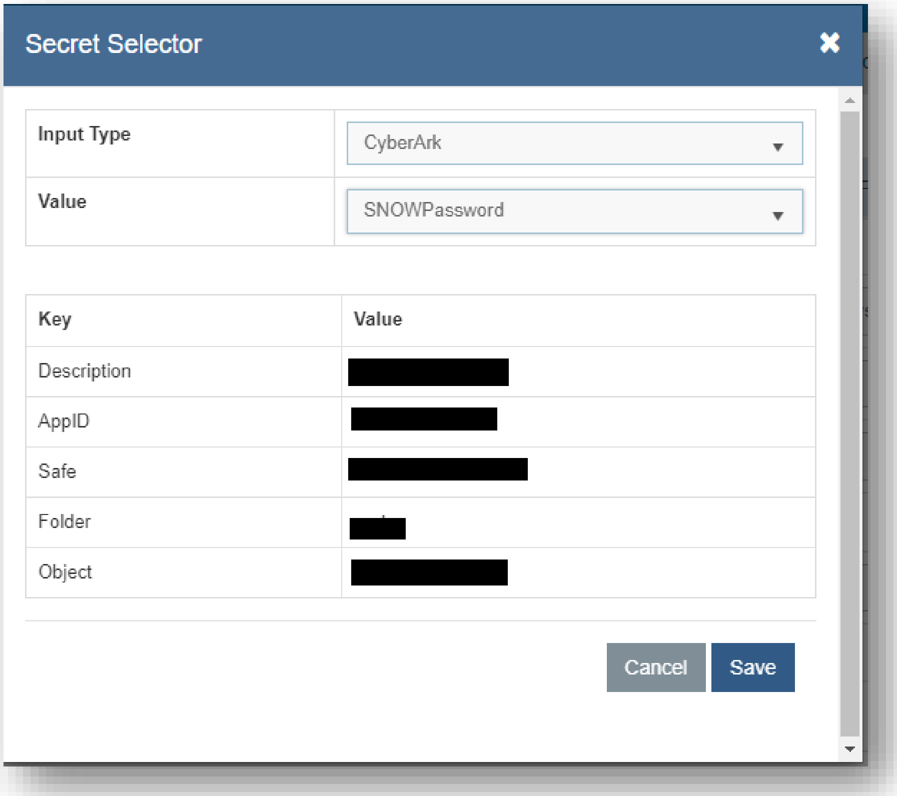
For password, click on icon next to it. If the password is available in plaintext then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Secret Selector

Input Type	Input text
Value	•••••••••••••••••• □ Show Password

Figure 168 – Password in plaintext



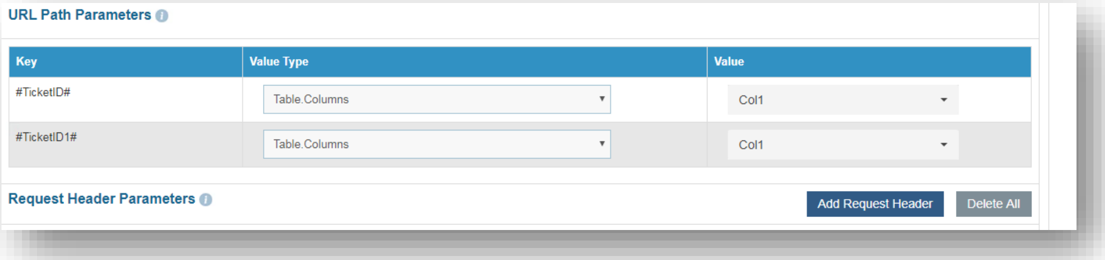
Input Type	CyberArk
Value	SNOWPassword

Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 169 – Password from Key Vault (CyberArk)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs.

```
Key: #TicketID#

ValueType: Table Columns

Value:

Select from dropdown that mapped to sys_id from previous screen
“Col1”


Key: #TicketID1#

ValueType: Table Columns

Value:

Select from dropdown that mapped to sys_id from previous screen
“Col1”
```



URL Path Parameters

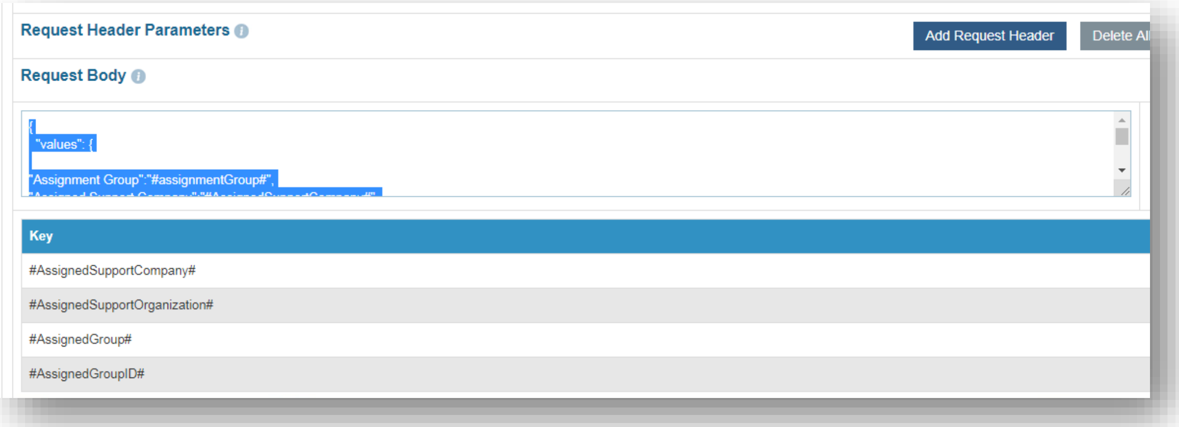
Key	Value Type	Value
#TicketID#	Table.Columns	Col1
#TicketID1#	Table.Columns	Col1

Request Header Parameters

Figure 170 – Release Rules Configuration (URL Path Parameters)

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body –** In this section, please enter the request body in JSON format. A sample request is mentioned below.

```
Request Body –
{
  "values": {

"Assignment Group":"#assignmentGroup#",
"Assigned Support Company":"#AssignedSupportCompany#",
"Assigned Support Organization":"#AssignedSupportOrganization#",
"Assigned Group":"#AssignedGroup#",
"Assigned Group ID":"#AssignedGroupID#",
"WorkInfo Submitter":"#z1D_WorkInfoSubmitter#",
"WorkLog Details":"#z1D_WorklogDetails#",
"z1D Details":"#z1D_Details#",
"z1D View Access":"#z1D_Activity_Type#",
"z1D Secure Access":"#z1D_View_Access#",
"z1D Secure Logs":"#z1D_Secure_Logs#"}
}
```



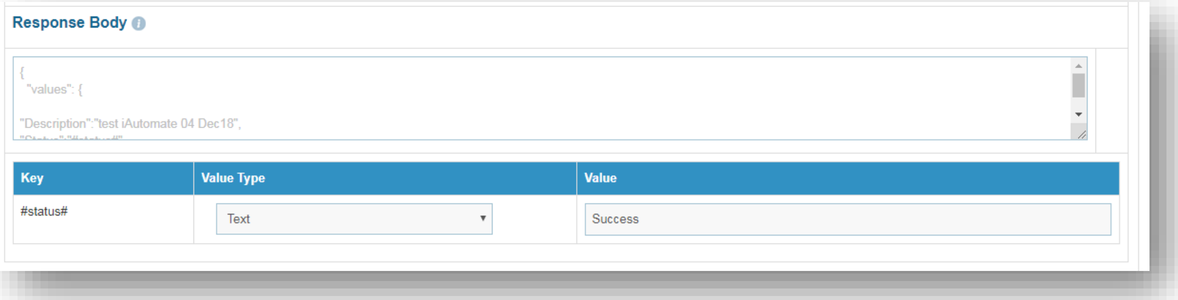
```
{
  "values": {

"Assignment Group":"#assignmentGroup#",
```

Key
#AssignedSupportCompany#
#AssignedSupportOrganization#
#AssignedGroup#
#AssignedGroupID#

Figure 171 – Release Rules Configuration (Request Body)

- **Response Body –** In this section, please enter the response body in JSON format. A sample response is mentioned below.

```
Response Body –
{
  "values": {


"Description":"test BigFix Runbook AI 04 Dec18",
"Status":"#status#"
}}
```



```
{
  "values": {

"Description":"test iAutomate 04 Dec18",
```

Key	Value Type	Value
#status#	Text ▾	Success

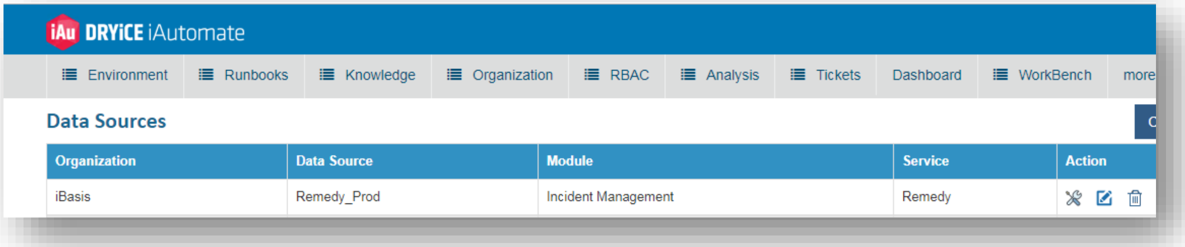
Figure 172 – Release Rules Configuration (Response Body)

- **Response Key Value** mapping can be done as per the below table.

Table 31– Sample Response Key Value Mapping

#success#	Text	Success

- Click **Submit** to add the data source.
- In order to bring the tickets within BigFix Runbook AI scope, a specific queue needs to be configured in the ITSM tool and same has to be configured in BigFix Runbook AI. This is achieved through **Manage the Entry Criteria**. Please perform the below steps –
  - Go to Action tab and click manage Data Sources.
  - On the **Data Sources** tab, click 🛠 next to the data source user wants to manage. **Manage Entry Criteria** screen appears.

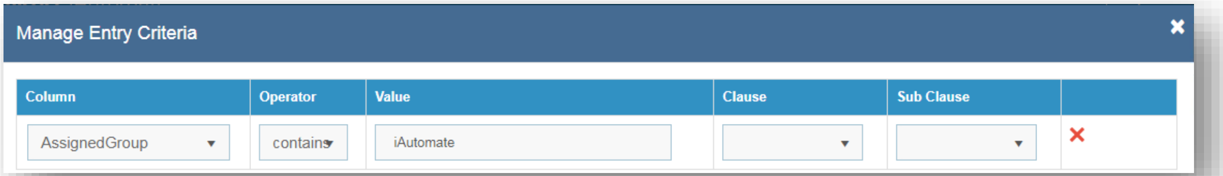


Data Sources

Organization	Data Source	Module	Service	Action
iBasis	Remedy_Prod	Incident Management	Remedy	🛠 ✎ 🗑

Figure 173 – Manage Entry Criteria

  - Select ‘AssignedGroup’ for the **Column** field and ‘equals to’ for the **Operator** field.
  - Enter the sys_id of the assignment group in ServiceNow in the **Value** field.
  - **Clause** and **Sub-Clause** fields can also be added based on requirement.



Manage Entry Criteria

Column	Operator	Value	Clause	Sub Clause	
AssignedGroup	contains	iAutomate			✕

Figure 174 – Manage Entry Criteria (cont.)

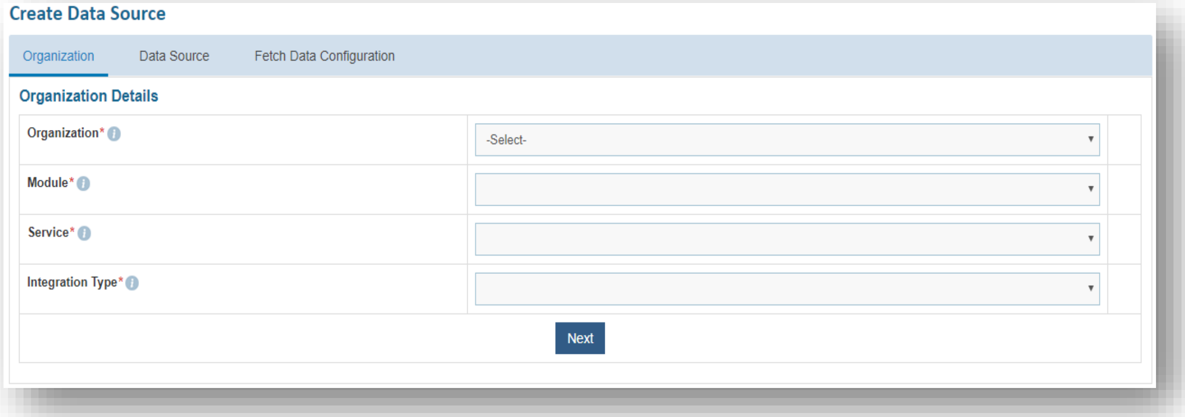
  - Click **Save**.

4.4 Integration with Cherwell ITSM

4.4.1 Incident Management

To create a data source for Incident Management, perform the following steps:

- On the main menu bar, click **Action tab → Manage Data Sources**.
- The **Create Data Source** page appears with the following tabs:
  - Organization
  - Data Source
  - Fetch Data Configuration
  - Release Rules Configuration



Create Data Source

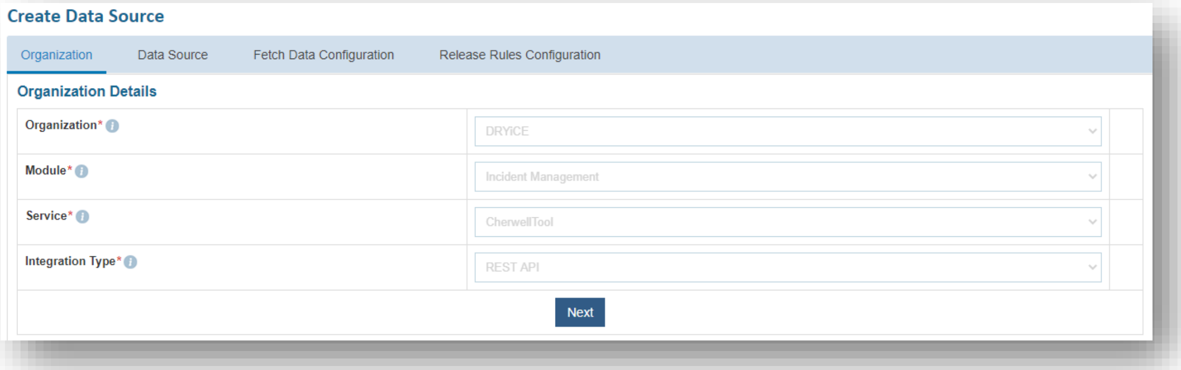
Organization Details

Organization*	-Select-
Module*	
Service*	
Integration Type*	

Figure 175 – Create Data Source

Release Rules Configuration is only applicable for the following **Module** types- **Incident Management, Change Request Task and Service Request Task.** This tab will not be activated for other module types.

- On the **Organization** tab,
  - Select the **Organization Name** from the dropdown.
  - Select the **Module** as **Incident Management,** since we are configuring this data source for pulling the incident tickets.
  - Select the **Service** as **Cherwell Tool** as we are configuring the data source for Cherwell
  - Select the **Integration Type** as **REST**, since we will be integrating through REST APIs.
  - Click **Next**.



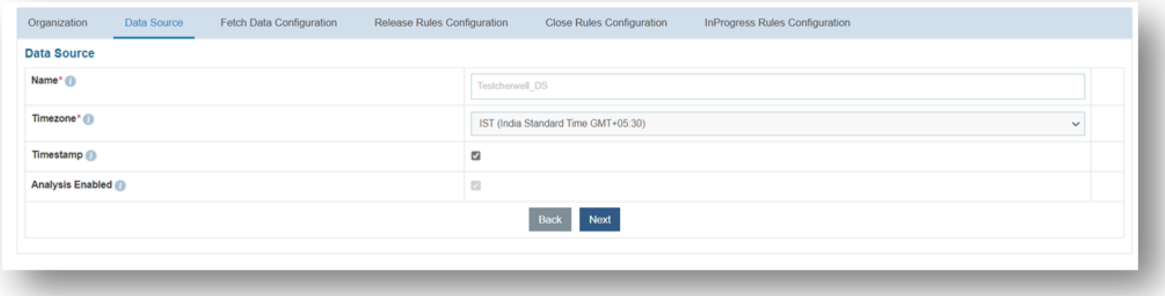
Create Data Source

Organization Details

Organization*	DRYICE
Module*	Incident Management
Service*	CherwellTool
Integration Type*	REST API

Figure 176 – Create Data Source (cont.)

- On the **Data Source** tab,
  - Type the new data source in the **Name** field.
  - Select the **Timezone** to specify the time zone of the selected data source.
  - Select **Timestamp** to view the present data with date and time.
  - Select **Analysis Enabled?** if user wants to analyze the data retrieved from the data source.
  - Click **Next**.


Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration
Back
Next

Data Source

Name*	Testcherwell_DS
Timezone*	IST (India Standard Time GMT+05:30)
Timestamp	☑
Analysis Enabled	☑

Figure 177 – Create Data Source (cont.)

- On the **Fetch Data Configuration** tab, type in the details as per the environment.
- In the **Connection Details** section, enter the following details:
  - **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.
  - **Sample URL** - http://<iAutomate_API_URL>/iAutomateAPI/Request/GetIncidentTicketData/<Org_ID>?start_date>=#Start_Date#&end_date<=#End_Date#&

- Here, < iAutomate_API_URL > is the API URL of BigFix Runbook AI where Push APIs are present and <Org_ID> is the OrgID for the organization for which you are creating the data source. It is available in Organization Master in Database.
- **Authentication Type** – Select one of the Authentication Types from Basic / Windows, JWT, OAuth 2.0

The user details that are entered here should be an API User

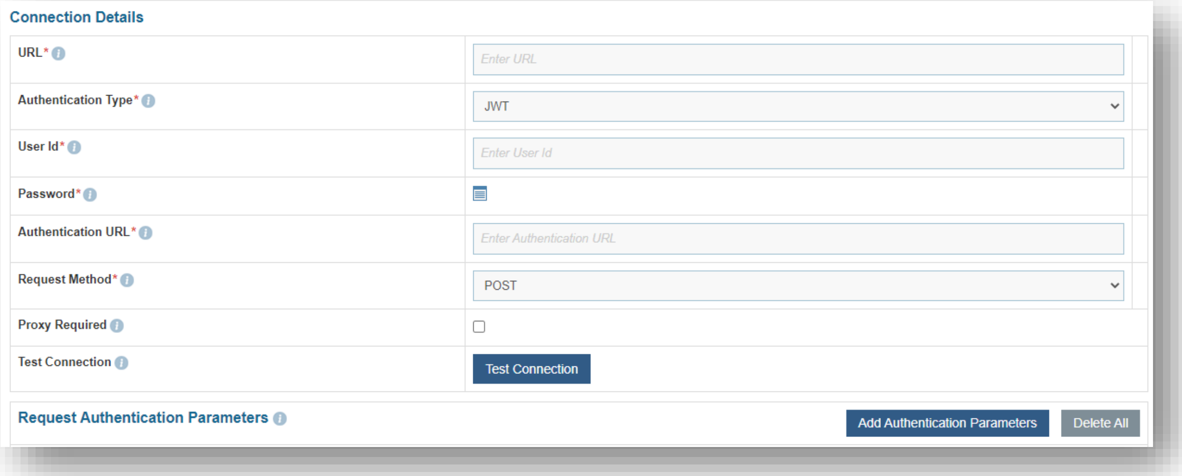
Selection of **Basic / Windows** requires you to enter -

- User Id
- Password.

Selection of **JWT / OAuth 2.0** requires you to enter -

- User Id
- Password
- Authentication URL

- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



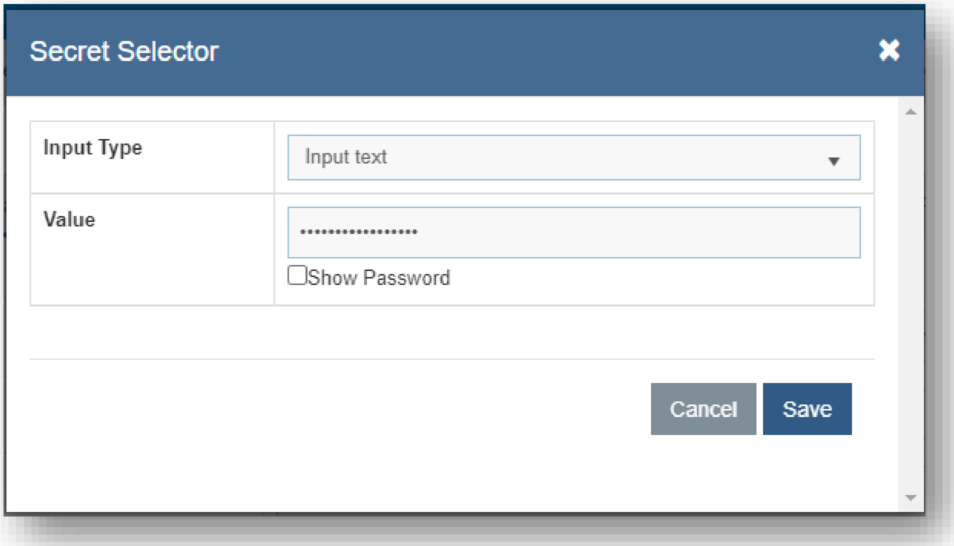
Connection Details

URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	POST
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

Figure 178 – Create Data Source (Connection Details)

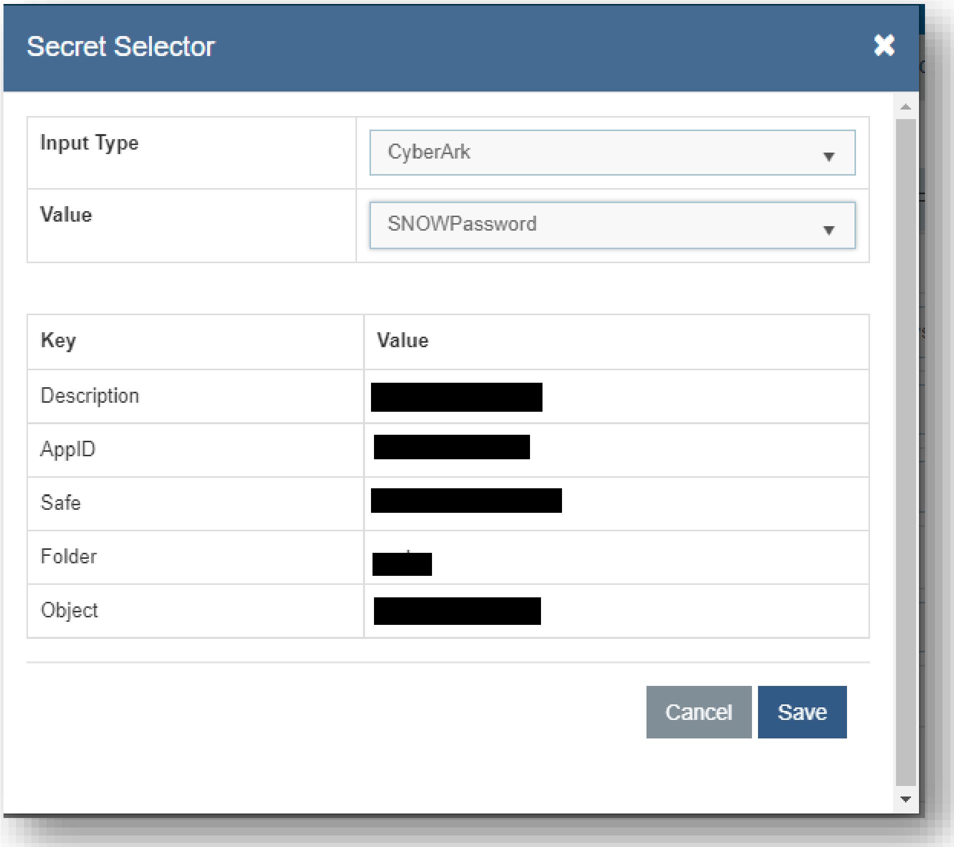
- **Password** – For password, click on icon next to it. If the password is available in plaintext then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Secret Selector

Input Type	Input text ▾
Value	••••••••••••••••• ☐ Show Password

Figure 179 – Password in plaintext



Secret Selector

Input Type	CyberArk ▾
Value	SNOWPassword ▾

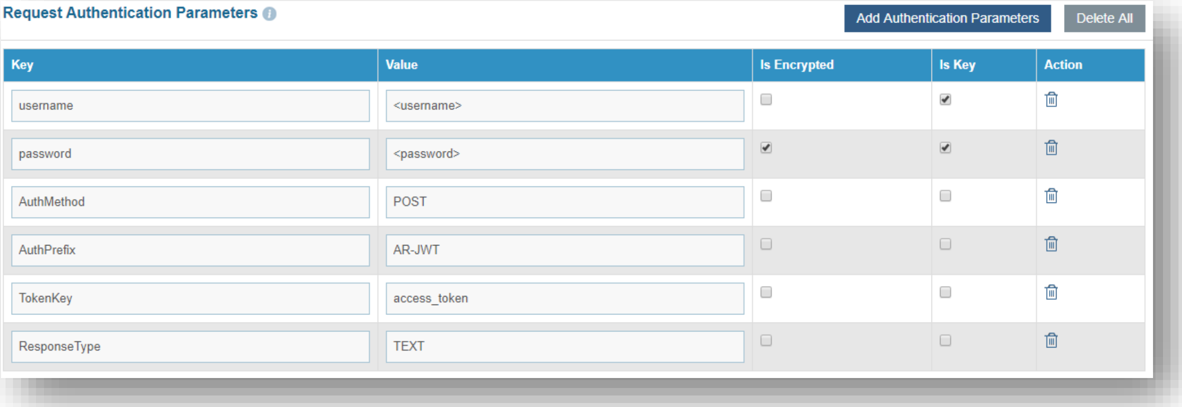
Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 180 – Password from Key Vault (CyberArk)

- **Request Authentication Parameters** – If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.
- Based on the **Authentication Type**, add the parameters mentioned in the below table.

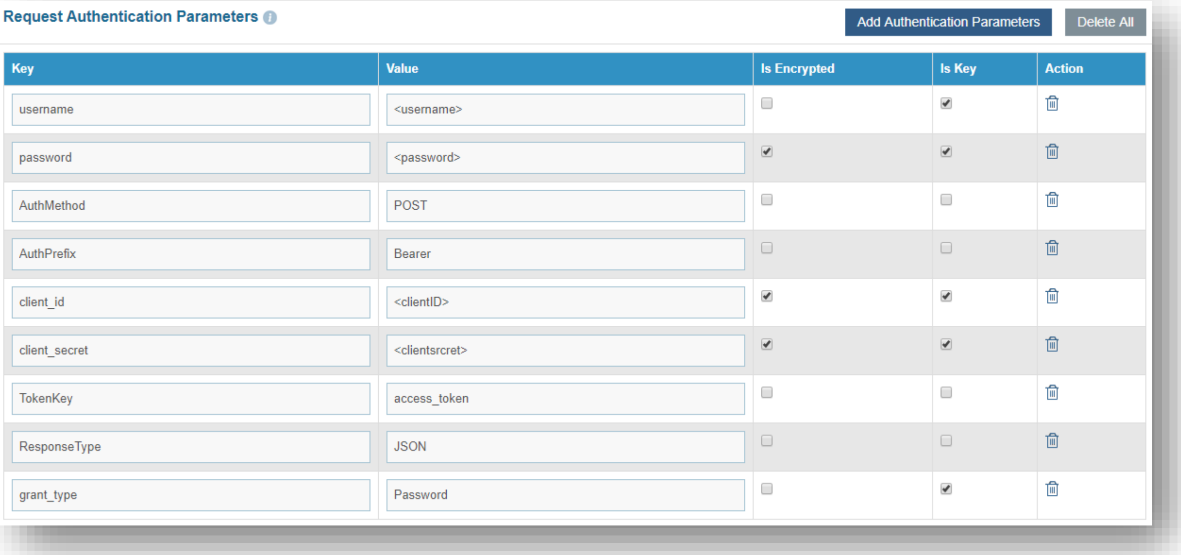
Table 32– Sample Authentication Parameters

Authentication Type	Key	Value	Is Encrypted?	Is Key?
JWT	username	<username>	NO	YES
JWT	password	<password>	YES	YES
JWT	AuthMethod	POST	NO	NO
JWT	AuthPrefix	AR-JWT	NO	NO
JWT	TokenKey	access_token	NO	NO
JWT	ResponseType	TEXT	NO	NO
OAuth2.0	username	<username	NO	YES
OAuth2.0	password	<password>	YES	YES
OAuth2.0	AuthMethod	POST	NO	NO
OAuth2.0	AuthPrefix	Bearer	NO	NO
OAuth2.0	client_id	<clientID>	YES	YES
OAuth2.0	client_secret	<clientsecret>	YES	YES
OAuth2.0	TokenKey	access_token	NO	NO
OAuth2.0	ResponseType	JSON	NO	NO
OAuth2.0	grant_type	Password	NO	YES


Request Authentication Parameters
Add Authentication Parameters
Delete All

Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	AR-JWT	☐	☐	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	TEXT	☐	☐	🗑

Figure 181 – Create Data Source (Request Authentication Parameters for JWT)



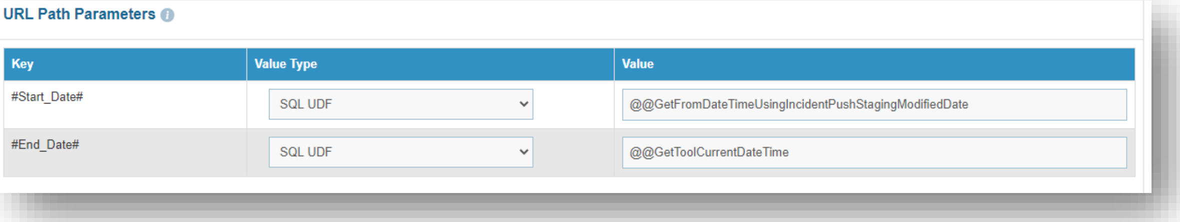
Request Authentication Parameters

Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	Bearer	☐	☐	🗑
client_id	<clientID>	☑	☑	🗑
client_secret	<clientsrcret>	☑	☑	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	JSON	☐	☐	🗑
grant_type	Password	☐	☑	🗑

Figure 182 – Create Data Source (Request Authentication Parameters for OAuth2.0)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #StartDate#

ValueType: SQL UDF

VALUE: @@GetFromDateTimeUsingIncidentPushStagingModifiedDate


Key: #EndDate#

ValueType: SQL UDF

VALUE: @@GetToolCurrentDateTime
```



URL Path Parameters

Key	Value Type	Value
#Start_Date#	SQL UDF	@@GetFromDateTimeUsingIncidentPushStagingModifiedDate
#End_Date#	SQL UDF	@@GetToolCurrentDateTime

Figure 183– URL Path Parameters

– **Request Header Parameters –** Please enter the request header parameters as required.

- **Response Body –** In this section, please enter the output of URL query for one of the incidents in JSON format. A sample response is mentioned below.

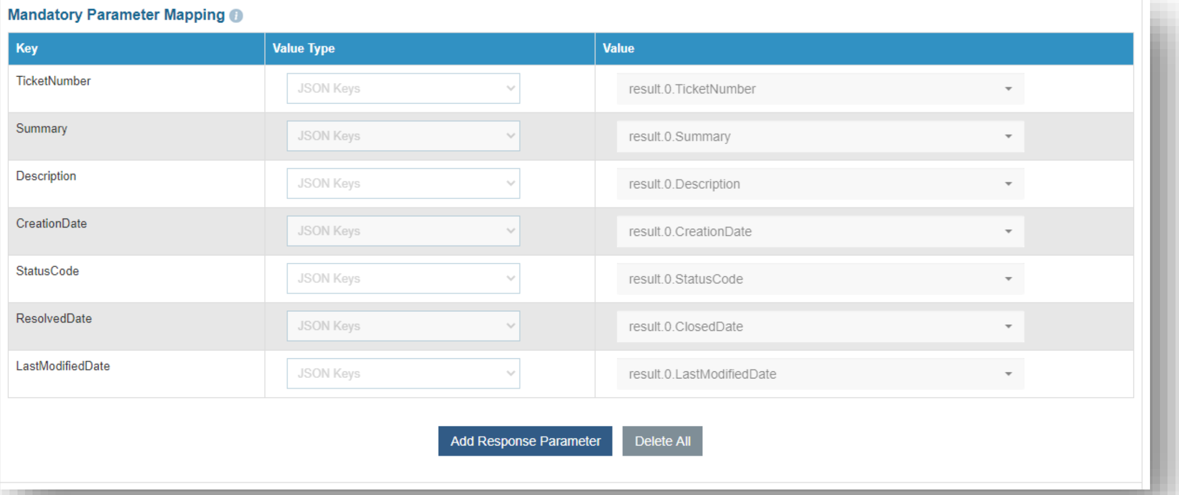
```
Response Body –

{"result": [{

            "TicketNumber": "INC0303860",

            "Summary": "testing",

            "Description": "testing data",

            "AssignedGroup": "02cc6a39376e4f00c72b2b2943990e69",

            "StatusCode": "1",

            "CreationDate": "2020-05-06 12:06:05.000",

            "LastModifiedDate": "2020-05-06 12:06:05.000",

            "ClosedDate": "2020-05-06 12:26:05.000",

            "sys_id": "2b535ab3dbc988506d7550d3dc96190e",

            "Col1": "",

            "Col2": "A",

            "Col3": "A",

            "Col4": "A",

            "Col5": "A"

        }]

}
```

- After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section**.**
- **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below:

Table 33– Sample Mandatory Parameter Mapping

Key	Value Type	Value
TicketNumber	JSON.Keys	result.0.TicketNumber
Summary	JSON.Keys	result.0.Summary

Description	JSON.Keys	result.0.Description
CreationDate	JSON.Keys	result.0.CreationDate
StatusCode	JSON.Keys	result.0.StatusCode
ResolvedDate	JSON.Keys	result.0.ClosedDate
LastModifiedDate	JSON.Keys	result.0.LastModifiedDate



Mandatory Parameter Mapping

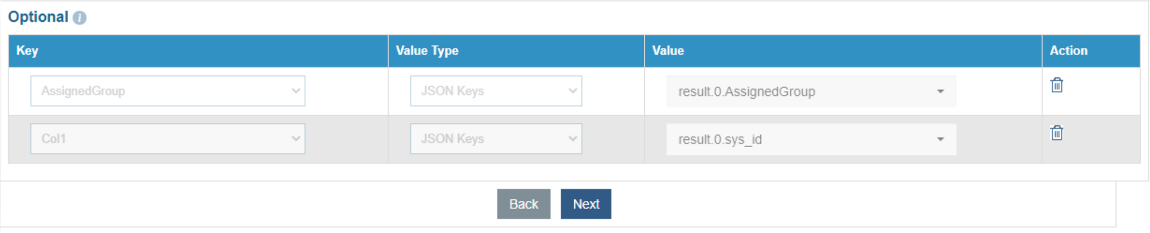
Key	Value Type	Value
TicketNumber	JSON Keys	result.0.TicketNumber
Summary	JSON Keys	result.0.Summary
Description	JSON Keys	result.0.Description
CreationDate	JSON Keys	result.0.CreationDate
StatusCode	JSON Keys	result.0.StatusCode
ResolvedDate	JSON Keys	result.0.ClosedDate
LastModifiedDate	JSON Keys	result.0.LastModifiedDate

Figure 184 – Mandatory Parameter Mapping

- If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 34– Sample Optional Parameters

Key	Value Type	Value
AssignedGroup	JSON.Keys	result.0.AssignedGroup
Col1	JSON.Keys	result.0.sys_id



Optional

Key	Value Type	Value	Action
AssignedGroup	JSON Keys	result.0.AssignedGroup	🗑
Col1	JSON Keys	result.0.sys_id	🗑

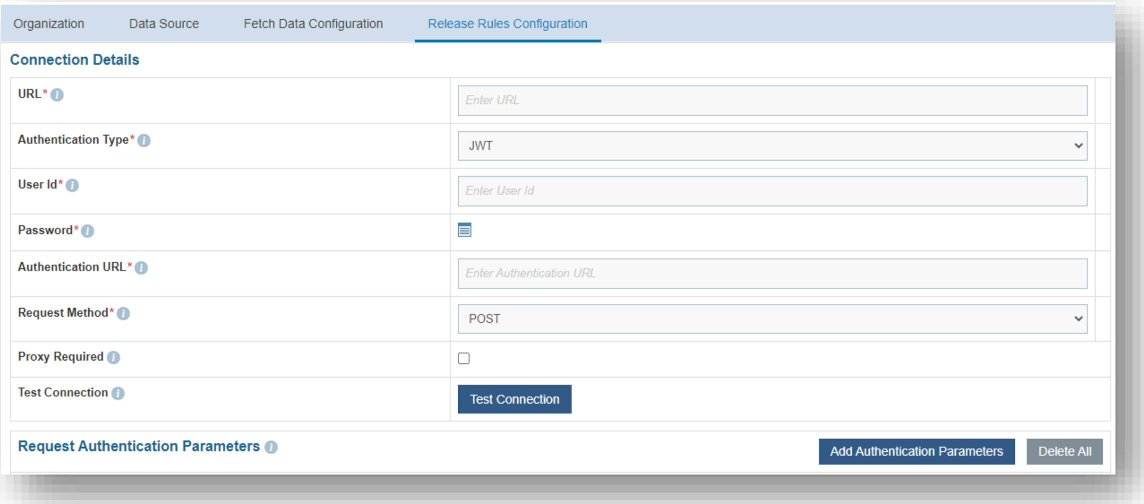
Figure 185 – Optional Parameter Mapping

- Click Next to proceed to Release Rules Configuration.

– On **Release Rules Configuration** tab, type in the details as per the requirement.

– In the **Connection Details** section, enter the following details:

- **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.
- **Sample URL** - https://<url>.cherwellondemand.com/CherwellAPI/api/V1/savebusinessobjectbatch
- **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously. For e.g., **JWT**.
- **Request Method** – Select Request Method as POST from the drop-down.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



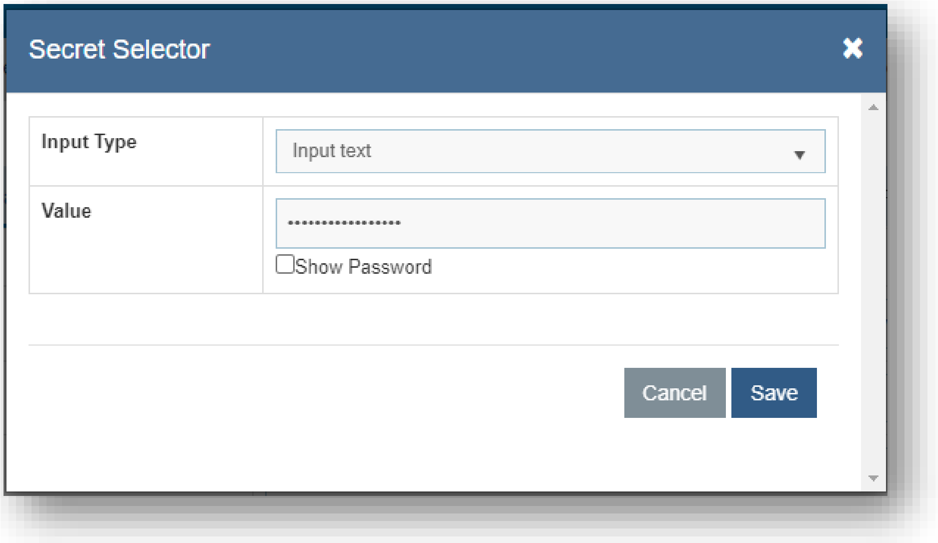
Connection Details

URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	POST
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

Figure 186 – Release Rules Configuration (Connection Details)

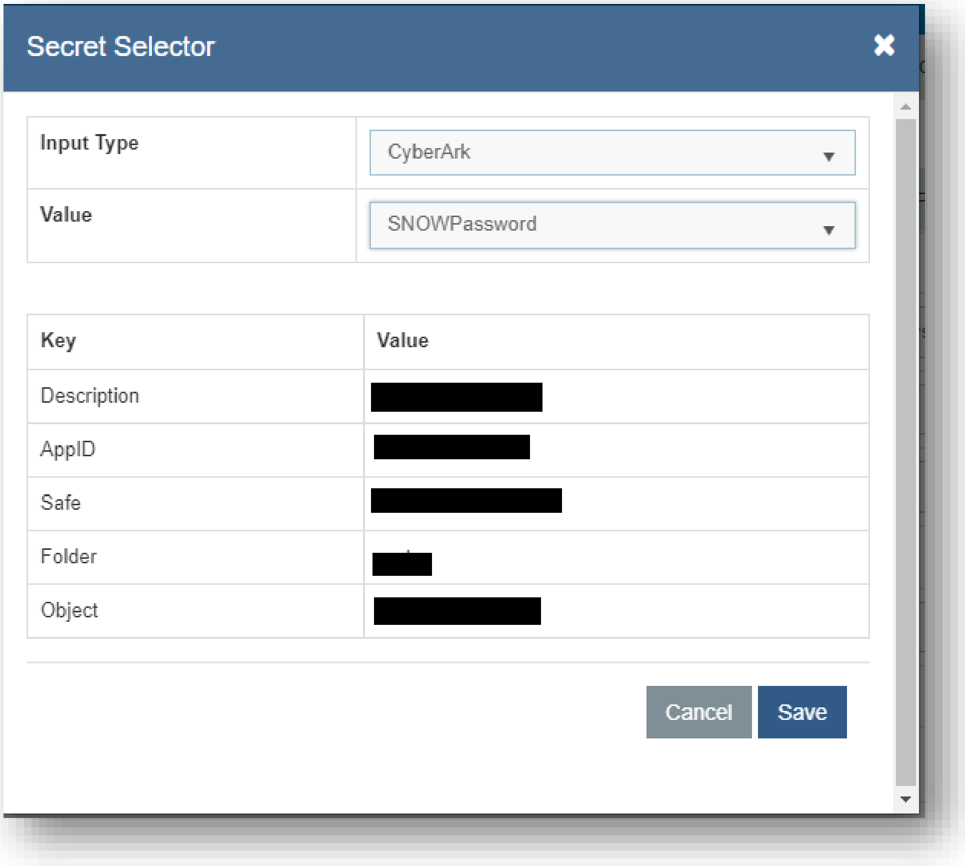
- **Password** – For password, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Secret Selector

Input Type	Input text
Value	•••••••••••••••••• □ Show Password

Figure 187 – Password in plaintext



Secret Selector

Input Type	CyberArk
Value	SNOWPassword

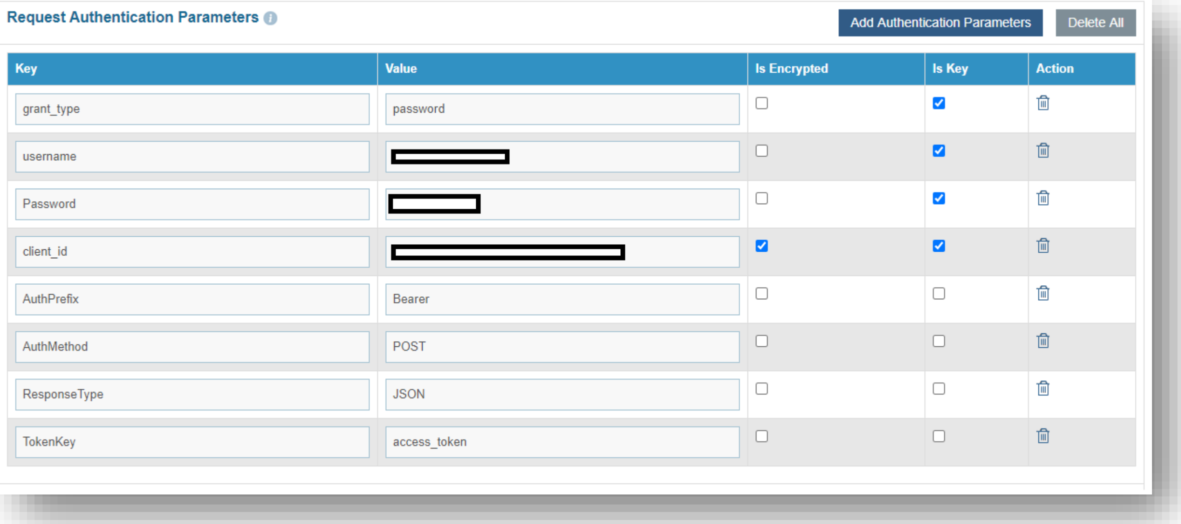
Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 188 – Password from Key Vault (CyberArk)

- **Request Authentication Parameters -** If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.
- Based on the **Authentication Type**, add the parameters mentioned in the below table

Table 35– Sample Authentication Parameters

Key	Value	Is Encrypted?	Is Key?
grant_type	password	N	Y
username	<username>	N	Y
Password	<password>	Y	Y
client_id	<client_id>	N	Y
AuthPrefix	Bearer	N	N
AuthMethod	POST	N	N
ResponseType	JSON	N	N
TokenKey	access_token	N	N



Request Authentication Parameters

Key	Value	Is Encrypted	Is Key	Action
grant_type	password	☐	☑	🗑
username	[redacted]	☐	☑	🗑
Password	[redacted]	☐	☑	🗑
client_id	[redacted]	☑	☑	🗑
AuthPrefix	Bearer	☐	☐	🗑
AuthMethod	POST	☐	☐	🗑
ResponseType	JSON	☐	☐	🗑
TokenKey	access_token	☐	☐	🗑

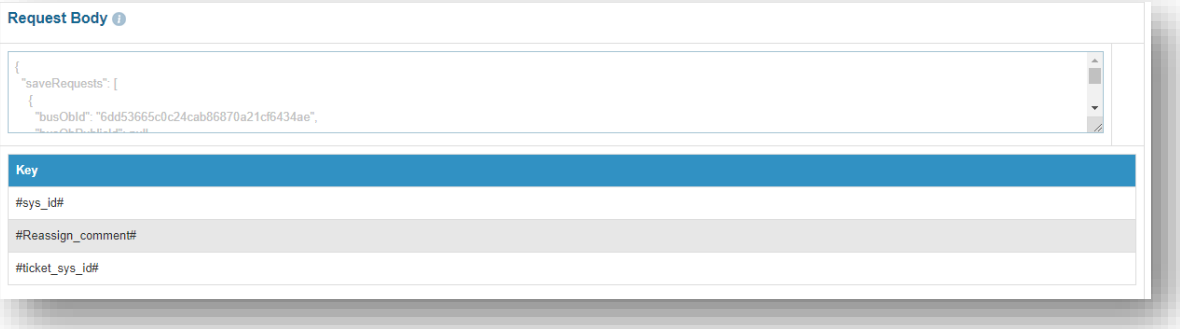
Figure 189 – Create Data Source (Request Authentication Parameters)

- **Request Body -** In this section, please enter the request body in JSON format. A sample request is mentioned below:

```
Request Body –
{
  "saveRequests": [
    {
```

```
"busObId": "6dd53665c0c24cab86870a21cf6434ae",
"busObPublicId": null,
"busObRecId": "#sys_id#",
"cacheKey": null,
"cacheScope": "Tenant",
"fields": [

{
"dirty": true,
"displayName": null,
"fieldId": "9339fc404e8d5299b7a7c64de79ab81a1c1ff4306c",
"html": null,
"name": null,
"value": "Service Desk"
},
{
"dirty": true,
"displayName": null,
"fieldId": "9339fc404e4c93350bf5be446fb13d693b0bb7f219",
"html": null,
"name": null,
"value": ""
},
{
"dirty": true,
"displayName": null,
"fieldId": "5eb3234ae1344c64a19819eda437f18d",
       "html": null,

       "name": null,

       "value": "Assigned"

       }


       ],

       "persist": true

     },

    {

       "busObId": "934d8181ba9d3a6a506d7643e1bc71f70fa9b47412",

       "busObPublicId": null,

       "busObRecId": null,

       "cacheKey": null,

       "cacheScope": "Tenant",

       "fields": [

         {

       "dirty": true,

       "displayName": null,

       "fieldId": "9341223bbcef1e2b8dfa6048a2bb4be1e94bad60ac",

       "html": null,

       "name": null,

       "value": "#Reassign_comment#"

     },

    {

       "dirty": true,

       "displayName": null,

       "fieldId": "9341222c4b89e253dd22b64d1fb16d0008bef6971f",
       "html": null,

       "name": null,

       "value": "#ticket_sys_id#"

     }

       ],

       "persist": true

     }

   ],

   "stopOnError": true}
```

–



Request Body

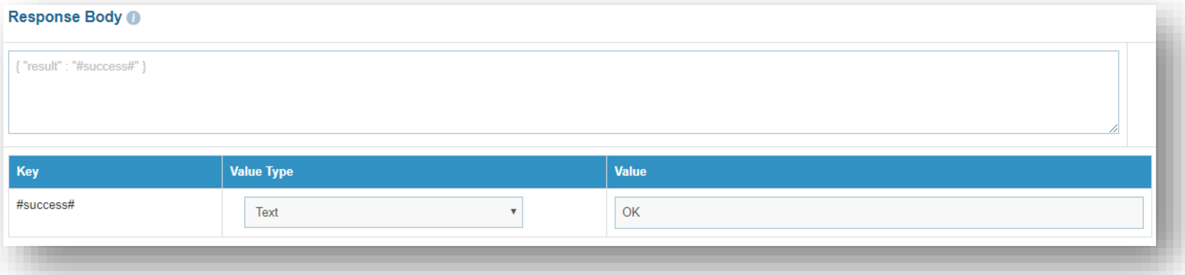
```
{
  "saveRequests": [
    {
      "busObId": "6dd53665c0c24cab86870a21cf6434ae",
```

Key
#sys_id#
#Reassign_comment#
#ticket_sys_id#

Figure 190 – Release Rules Configuration (Request Body)

– **Response Body –** In this section, please enter the response body in JSON format. A sample request is mentioned below:

```
Response Body –

{ "result" : "#success#" }
```



Response Body

{ "result" : "#success#" }

Key	Value Type	Value
#success#	Text	OK

Figure 191 – Release Rules Configuration (Response Body)

– **Response Key Value** mapping can be done as per the below table.

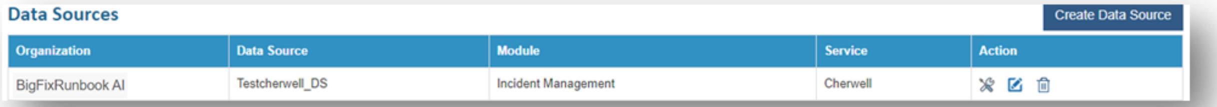
Table 36– Sample Response Key Value Mapping

#success#	Text	OK

– Click **Submit** to add the data source.

– In order to bring the tickets within BigFix Runbook AI scope, a specific queue needs to be configured in the ITSM tool and same has to be configured in BigFix Runbook AI. This is achieved through **Manage the Entry Criteria**. Please perform the below steps:

- Go to Action tab and click Manage Data Sources.
- On the **Data Sources** tab, click ⚒ next to the data source user wants to manage. **Manage Entry Criteria** screen appears.

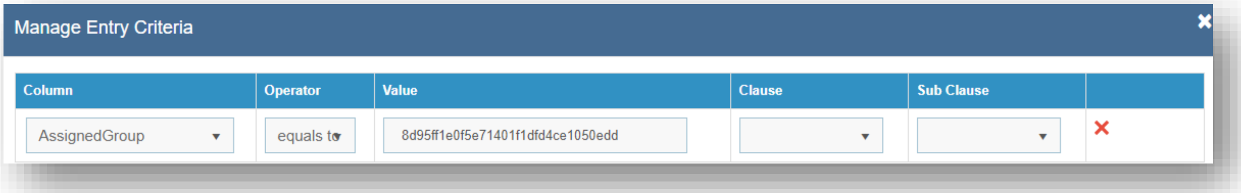


Data Sources

Organization	Data Source	Module	Service	Action
BigFixRunbook AI	Testcherwell_DS	Incident Management	Cherwell	⚒ ✎ 🗑

Figure 192 – Manage Entry Criteria

- Select 'AssignedGroup' for the **Column field and** 'equals to' for the **Operator** field.
- Enter the sys_id of the assignment group in Cherwell in the **Value** field.
- **Clause** and **Sub-Clause** fields can also be added based on requirement.

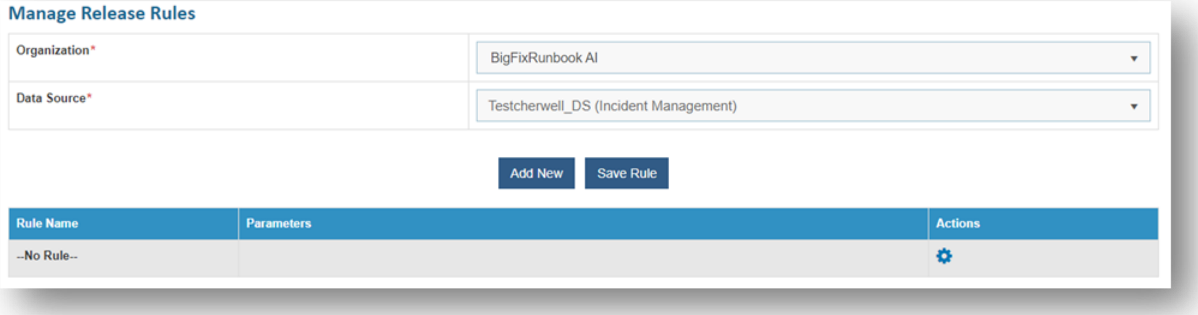


Manage Entry Criteria

Column	Operator	Value	Clause	Sub Clause	
AssignedGroup	equals to	8d95ff1e0f5e71401f1dfd4ce1050edd			✕

Figure 193 – Manage Entry Criteria (cont.)

- Click **Save**.
- To configure the Release rules for the data source created earlier, perform the below steps:
  - Go to Action Tab →Runbooks →Manage Rules.
  - Select the **Organization** and the data source created from **Data Source** dropdown.



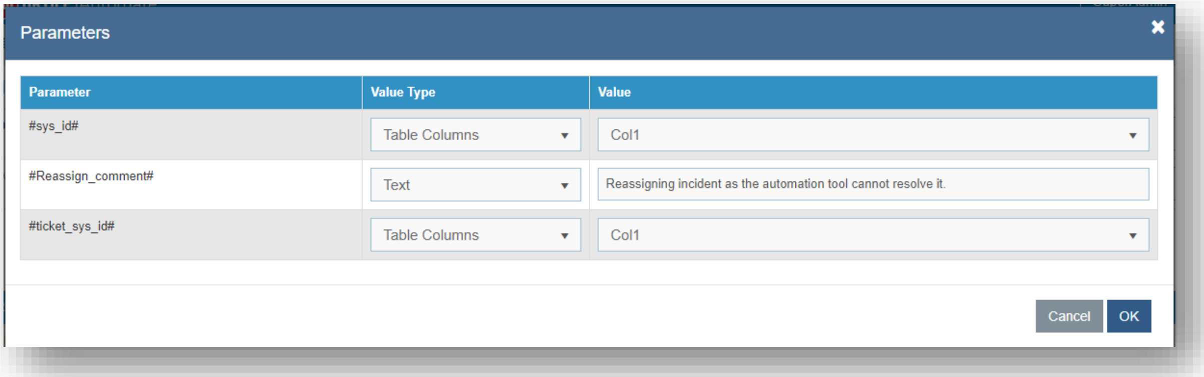
Manage Release Rules

Organization*	BigFixRunbook AI
Data Source*	Testcherwell_DS (Incident Management)

Rule Name	Parameters	Actions
--No Rule--		⚙

Figure 194 – Manage Release Rules

- Click on ⚙ corresponding to **–No Rule—**
- Map the parameters #sys_id# to the column in which sys_id was mapped while performing the mandatory parameter mapping while data source creation.
- Mention the reason for releasing ticket in #reassign_comments#.
- Map #ticket_sys_id# again to the column in which sys_id was mapped while performing the mandatory parameter mapping while data source creation.

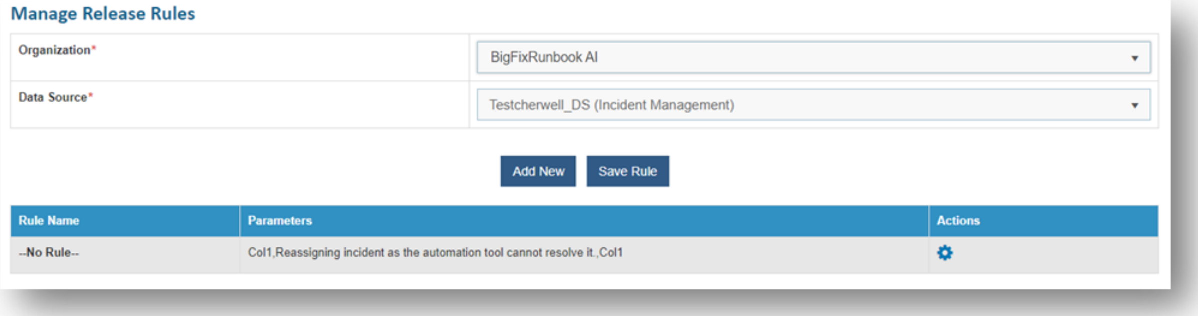


Parameters

Parameter	Value Type	Value
#sys_id#	Table Columns	Col1
#Reassign_comment#	Text	Reassigning incident as the automation tool cannot resolve it.
#ticket_sys_id#	Table Columns	Col1

Figure 195 – Manage Release Rules (cont.)

- Click **OK**.



Manage Release Rules

Organization*	BigFixRunbook AI
Data Source*	Testcherwell_DS (Incident Management)

Rule Name	Parameters	Actions
--No Rule--	Col1,Reassigning incident as the automation tool cannot resolve it.,Col1	⚙

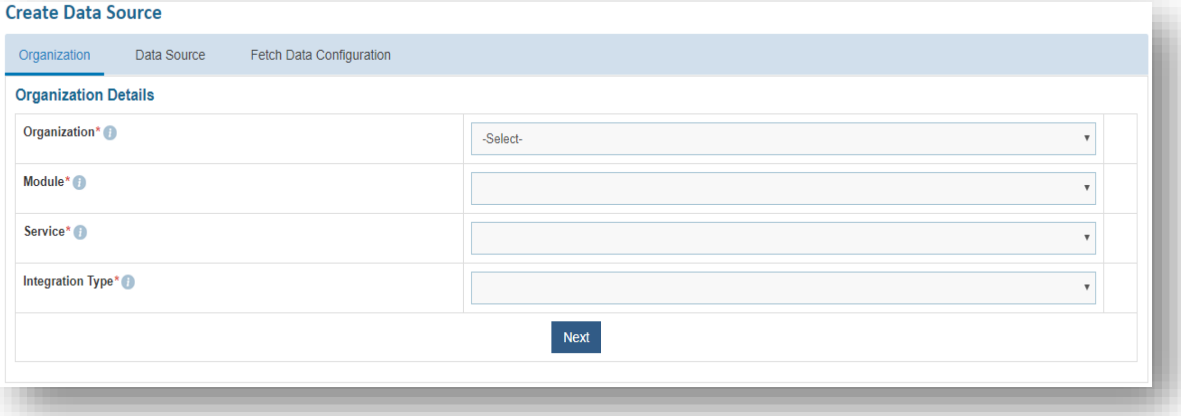
Figure 196 – Manage Release Rules (cont.)

- Click Save Rule.

4.4.2 Service Request Task Management

To create a data source for Service Request Task Management, perform the following steps:

- On the main menu bar, click **Actions tab → Manage Data Sources**.
- The **Create Data Source** page appears with the following tabs:
  - Organization
  - Data Source
  - Fetch Data Configuration
  - Release Rules Configuration



Create Data Source

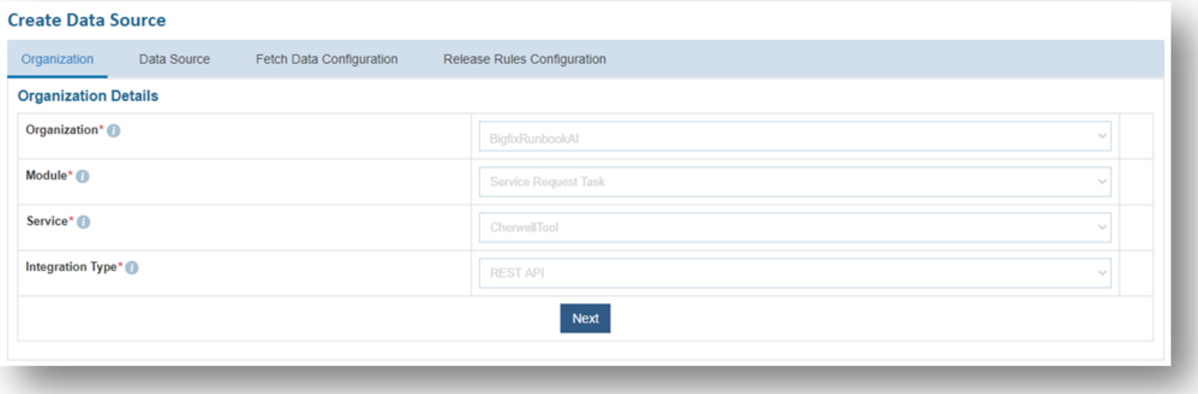
Organization Details

Organization*	-Select-
Module*	
Service*	
Integration Type*	

Figure 197 – Create Data Source

Release Rules Configuration is only applicable for the following **Module** types- **Incident Management, Change Request Task and Service Request Task.** This tab will not be activated for other module types.

- On the **Organization** tab,
  - Select the **Organization Name** from the dropdown.
  - In the **Module** field, select **Service Request Task,** since we are configuring this data source for pulling the service request task tickets.
  - In the **Service** field, select **Cherwell Tool** as we are configuring the data source for Cherwell
  - In the **Integration Type** field, select **REST**, since we will be integrating through REST APIs.
  - Click **Next**.



Create Data Source

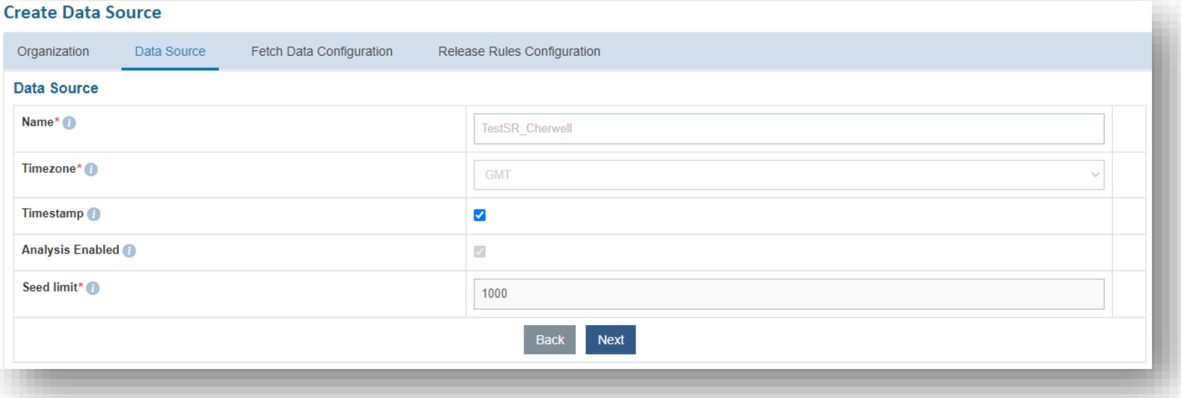
Organization Details

Organization*	BigfixRunbookAI
Module*	Service Request Task
Service*	CherwellTool
Integration Type*	REST API

Figure 198 – Create Data Source (cont.)

– On the **Data Source** tab,

- Type the new data source in the **Name** field.
- Select the **Timezone** to specify the time zone of the selected data source.
- Select **Timestamp** to view the present data with date and time.
- Select **Analysis Enabled** if you want to analyze the data retrieved from the data source.
- Click **Next**.



Create Data Source

Data Source

Name*	TestSR_Cherwell
Timezone*	GMT
Timestamp	☑
Analysis Enabled	☑
Seed limit*	1000

Figure 199 – Create Data Source (cont.)

– On the **Fetch Data Configuration** tab, populate the details as per the environment.

– In the **Connection Details** section enter the following details:

- **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.

- **Sample URL** - http://<iAutomate_API_URL>/iAutomateAPI/Request/ GetSRTicketData/<Org_ID>?start_date>=#Start_Date#&end_date<=#End_Date#&
- Here, < iAutomate_API_URL > is the API URL of BigFix Runbook AI where Push APIs are present and <Org_ID> is the OrgID for the organization for which you are creating the data source. It is available in Organization Master in Database.
- **Authentication Type** – Select one of the Authentication Types from Basic / Windows, JWT, OAuth 2.0

The user details that are entered here should be an API User

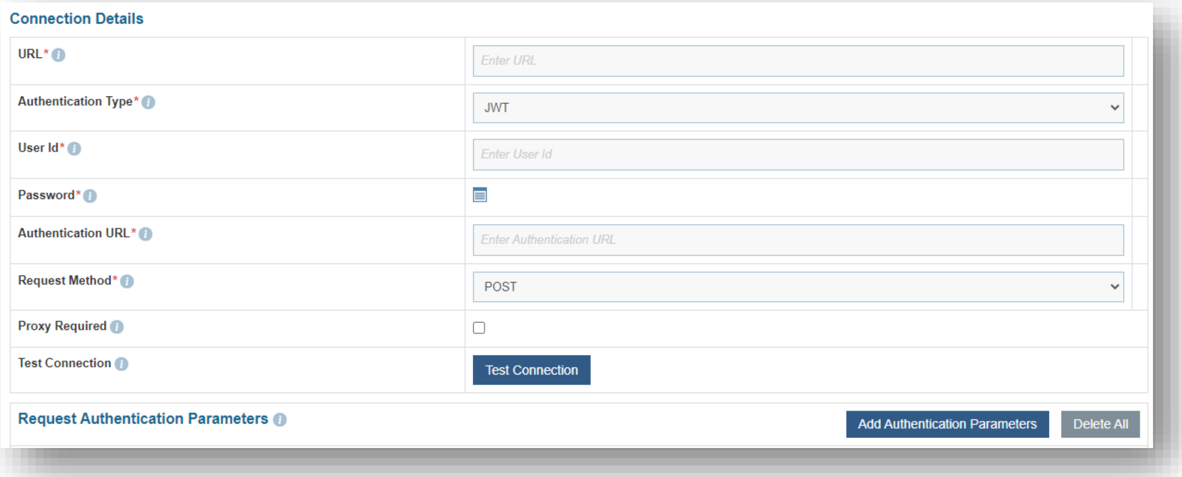
Selection of **Basic / Windows** requires you to enter -

- User Id
- Password

Selection of **JWT / OAuth 2.0** requires you to enter -

- User Id
- Password
- Authentication URL

- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



Connection Details

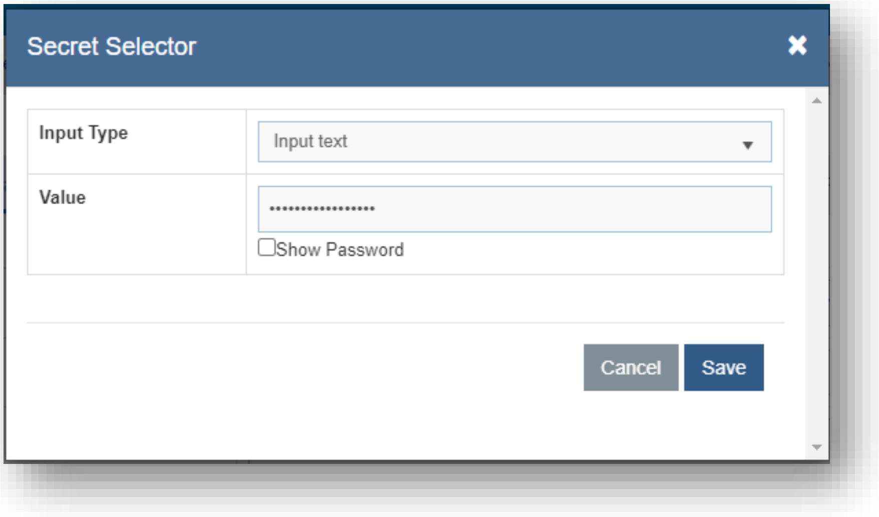
URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	POST
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

Figure 200 – Create Data Source (Connection Details)

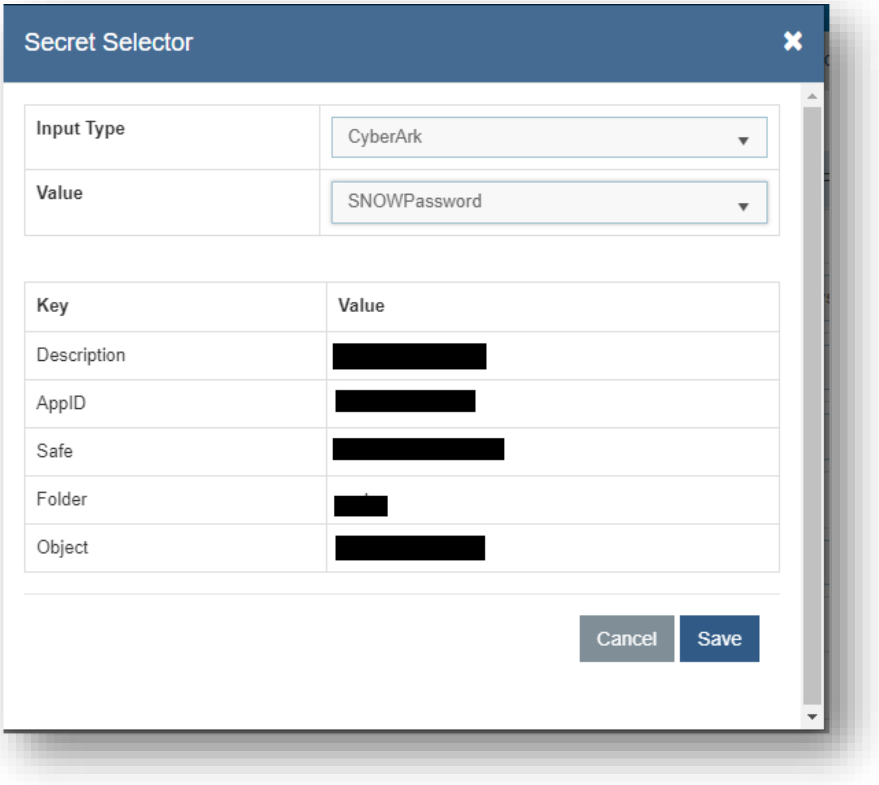
- For **Password**, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key

Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Input Type	Input text ▾
Value	•••••••••••••••• □ Show Password

Figure 201 – Password in plaintext



Input Type	CyberArk ▾
Value	SNOWPassword ▾

Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

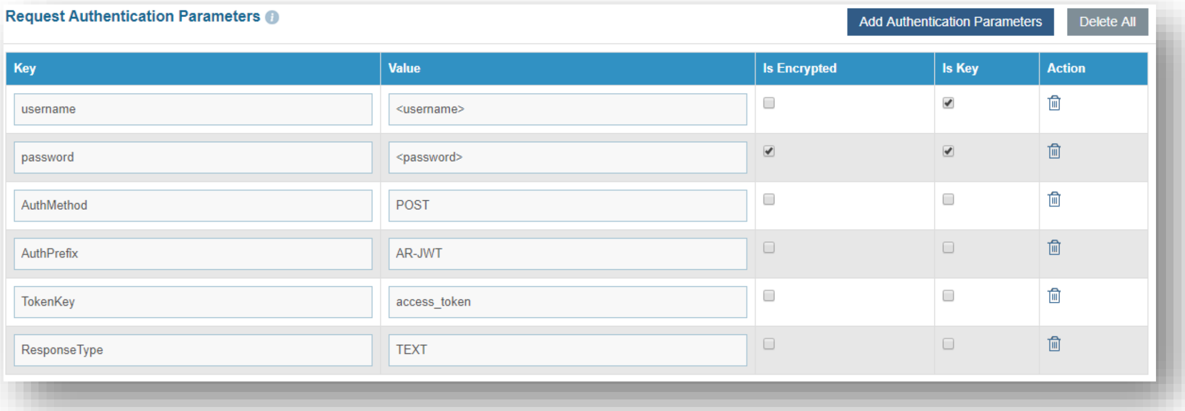
Figure 202 – Password from Key Vault (CyberArk)

– **Request Authentication Parameters** – If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.

– Based on the **Authentication Type**, add the parameters mentioned in the below table.

Table 37– Sample Authentication Parameters

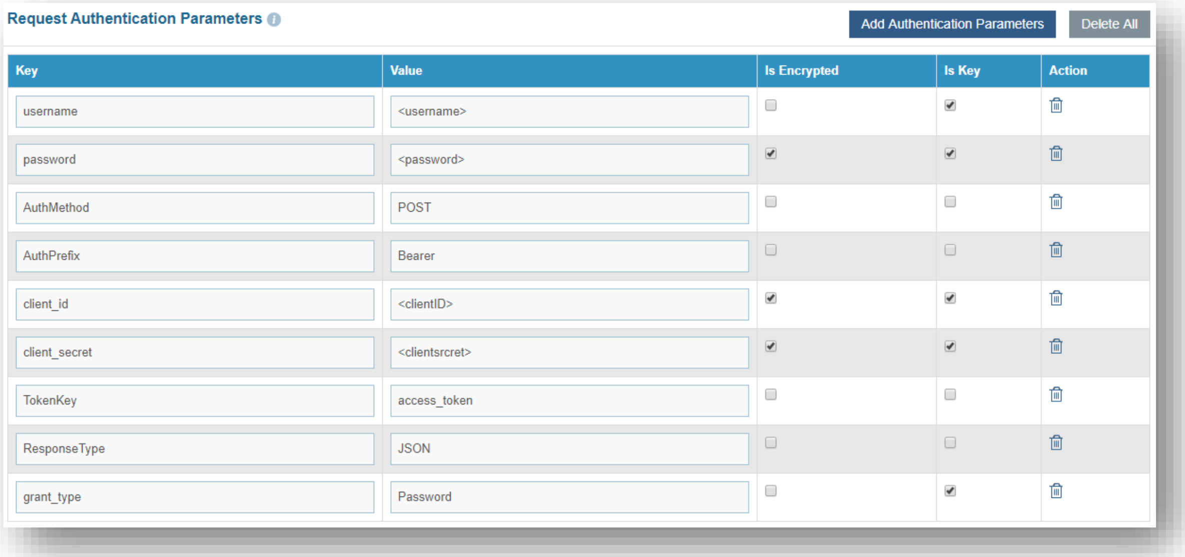
Authentication Type	Key	Value	Is Encrypted?	Is Key?
JWT	username	<username>	NO	YES
JWT	password	<password>	YES	YES
JWT	AuthMethod	POST	NO	NO
JWT	AuthPrefix	AR-JWT	NO	NO
JWT	TokenKey	access_token	NO	NO
JWT	ResponseType	TEXT	NO	NO
OAuth2.0	username	<username	NO	YES
OAuth2.0	password	<password>	YES	YES
OAuth2.0	AuthMethod	POST	NO	NO
OAuth2.0	AuthPrefix	Bearer	NO	NO
OAuth2.0	client_id	<clientID>	YES	YES
OAuth2.0	client_secret	<clientsecret>	YES	YES
OAuth2.0	TokenKey	access_token	NO	NO
OAuth2.0	ResponseType	JSON	NO	NO
OAuth2.0	grant_type	Password	NO	YES



Request Authentication Parameters

Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	AR-JWT	☐	☐	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	TEXT	☐	☐	🗑

Figure 203 – Create Data Source (Request Authentication Parameters for JWT)



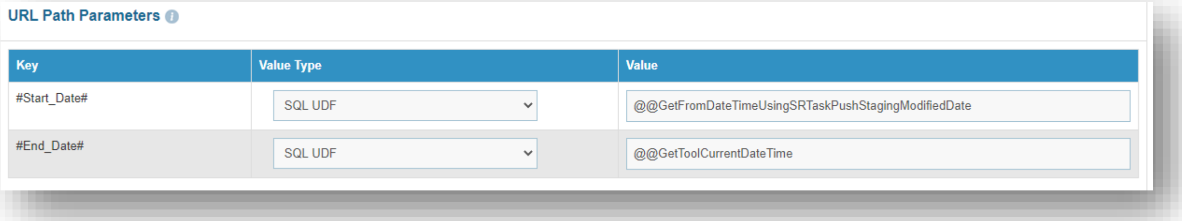
Request Authentication Parameters

Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	Bearer	☐	☐	🗑
client_id	<clientID>	☑	☑	🗑
client_secret	<clientsrcret>	☑	☑	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	JSON	☐	☐	🗑
grant_type	Password	☐	☑	🗑

Figure 204 – Create Data Source (Request Authentication Parameters for OAuth2.0)

- **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #StartDate#

ValueType: SQL UDF

VALUE: @@GetFromDateTimeUsingSRTaskPushStagingModifiedDate


Key: #EndDate#

ValueType: SQL UDF

VALUE: @@GetToolCurrentDateTime
```



URL Path Parameters

Key	Value Type	Value
#Start_Date#	SQL UDF	@@GetFromDateTimeUsingSRTaskPushStagingModifiedDate
#End_Date#	SQL UDF	@@GetToolCurrentDateTime

Figure 205– URL Path Parameters

- **Request Header Parameters –** Please enter the request header parameters as required.

- **Response Body –** In this section, please enter the output of URL query for one of the incidents in JSON format. A sample response is mentioned below.

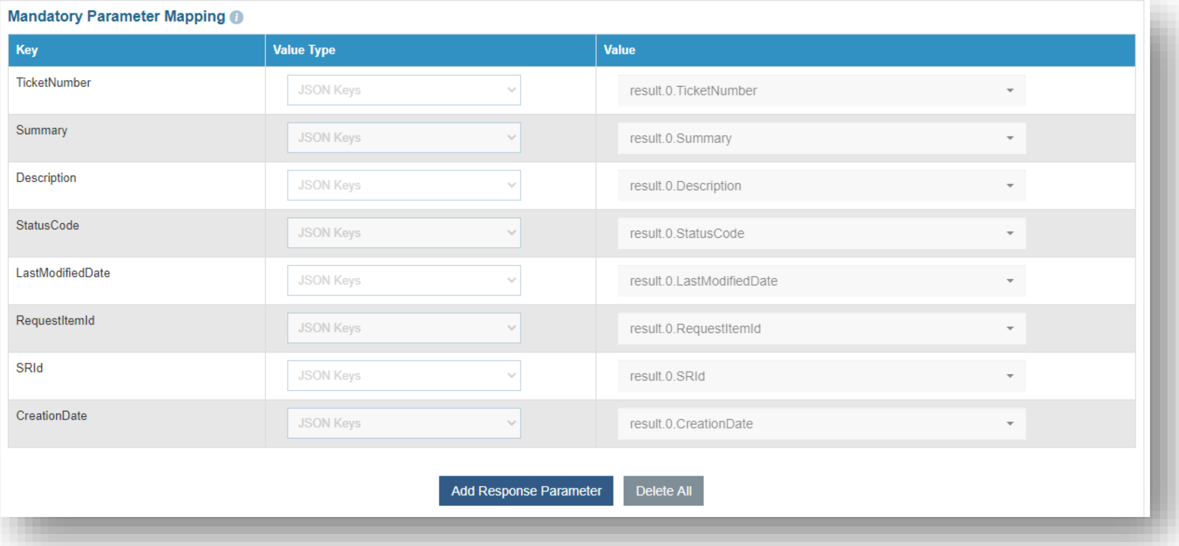
```
Response Body –

{"result": [{

            "TicketNumber": "SRTask0303863",

            "Summary": "testing",

            "Description": "testing data",

            "RequestItemId": "12345",

            "SRId": "2b535ab3dbc988506d7550d3dc96190e",

            "AssignedGroup": "",

            "StatusCode": "1",

            "CreationDate": "2020-05-07 05:06:05.000",

            "LastModifiedDate": "2020-05-07 05:54:54.000",

            "sys_id": "",

            "Col1": "",

            "Col2": "",

            "Col3": "",

            "Col4": "",

            "Col5": "",

            "iAutomate_CreatedDateInGMT": "2020-05-08
09:14:24.903",

            "iAutomate_UpdatedDateInGMT": "2020-05-08
09:14:24.903"

        }

]}
```

- After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section.

– **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below:

Table 38– Sample Mandatory Parameter Mapping

Key	Value Type	Value
TicketNumber	JSON.Keys	result.0.TicketNumber
Summary	JSON.Keys	result.0.Summary
Description	JSON.Keys	result.0.Description
StatusCode	JSON.Keys	result.0.StatusCode
LastModifiedDate	JSON.Keys	result.0.LastModifiedDate
RequestItemId	JSON.Keys	result.0.RequestItemId
SRId	JSON.Keys	result.0.SRId
CreationDate	JSON.Keys	result.0.CreationDate



Mandatory Parameter Mapping

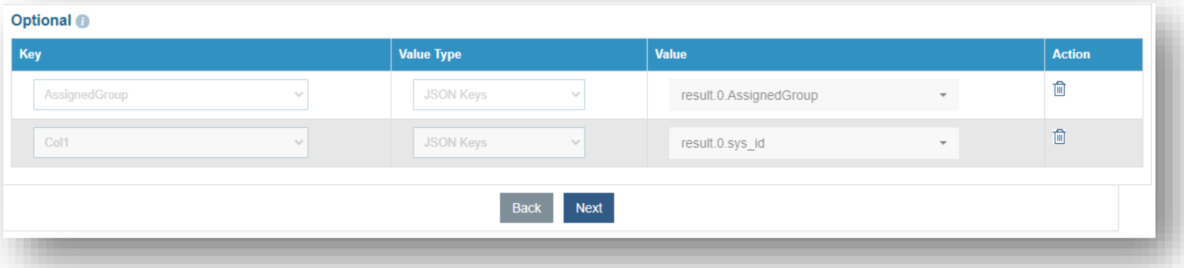
Key	Value Type	Value
TicketNumber	JSON Keys	result.0.TicketNumber
Summary	JSON Keys	result.0.Summary
Description	JSON Keys	result.0.Description
StatusCode	JSON Keys	result.0.StatusCode
LastModifiedDate	JSON Keys	result.0.LastModifiedDate
RequestItemId	JSON Keys	result.0.RequestItemId
SRId	JSON Keys	result.0.SRId
CreationDate	JSON Keys	result.0.CreationDate

Figure 206 – Mandatory Parameter Mapping

– If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 39– Sample Optional Parameters

Key	Value Type	Value
AssignedGroup	JSON.Keys	result.0.AssignedGroup
Col1	JSON.Keys	result.0.sys_id

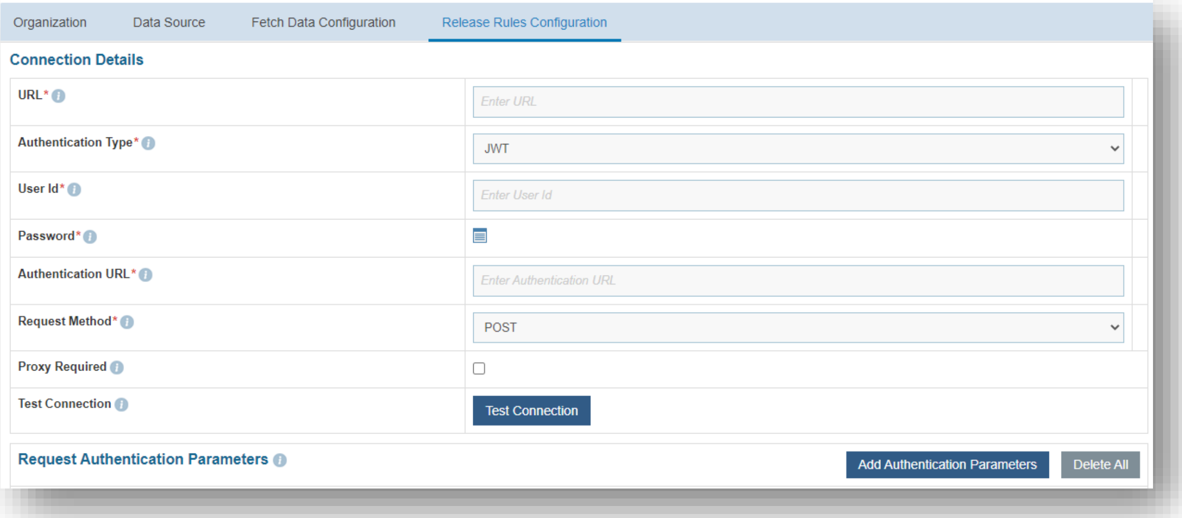


Optional

Key	Value Type	Value	Action
AssignedGroup	JSON Keys	result.0.AssignedGroup	🗑
Col1	JSON Keys	result.0.sys_id	🗑

Figure 207 – Optional Parameter Mapping

- Click Next to proceed to Release Rules Configuration.
- On **Release Rules Configuration** tab, type in the details as per the requirement.
- In the **Connection Details** section, enter the following details:
  - **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.
  - **Sample URL** - https://<url>. cherwellondemand.com/CherwellAPI/api/V1/savebusinessobjectbatch
  - **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously. For e.g., **JWT**.
  - **Request Method** – Select Request Method as POST from the drop-down.
  - **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
  - Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



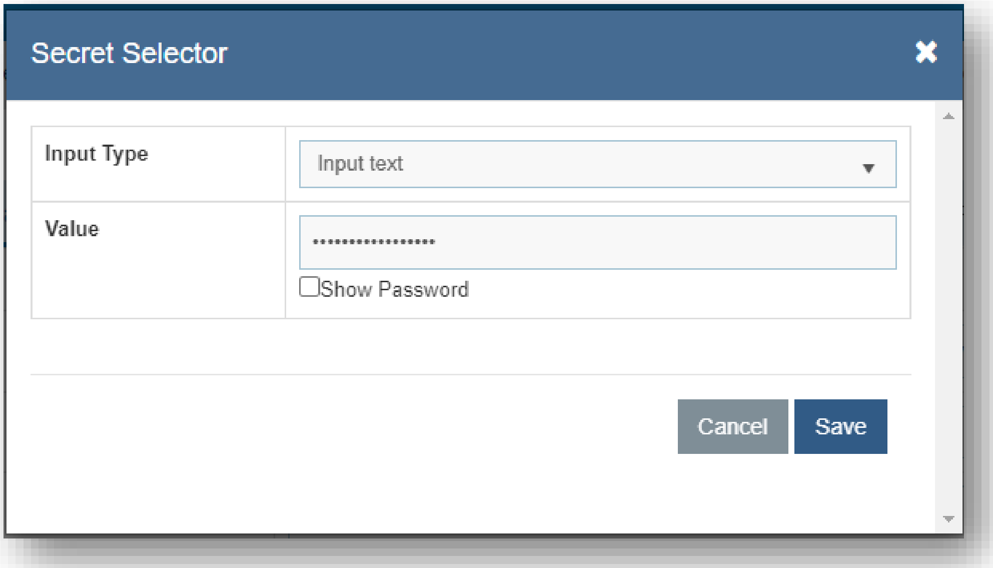
Connection Details

URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	POST
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

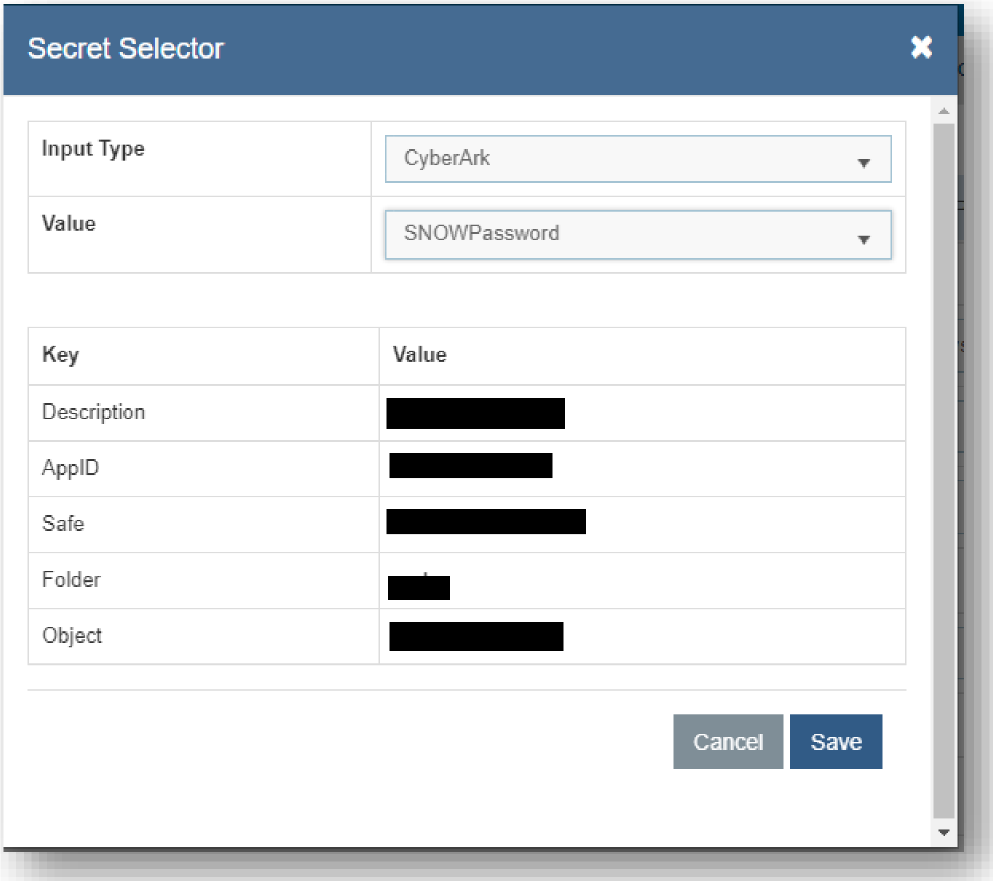
Figure 208 – Release Rules Configuration (Connection Details)

- For **Password**, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.


Secret Selector

Input Type	Input text
Value	••••••••••••••••• □Show Password

Figure 209 – Password in plaintext



Secret Selector

Input Type	CyberArk
Value	SNOWPassword

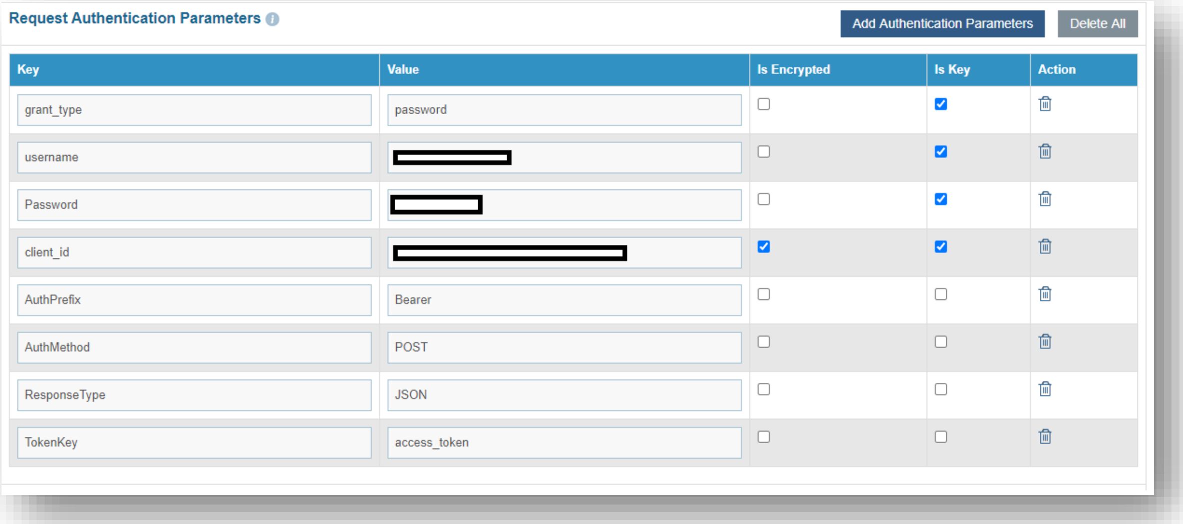
Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 210 – Password from Key Vault (CyberArk)

- Request Authentication Parameters - If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.
- Based on the **Authentication Type**, add the parameters mentioned in the below table

Table 40– Sample Authentication Parameters

Key	Value	Is Encrypted?	Is Key?
grant_type	password	N	Y
username	<username>	N	Y
Password	<password>	Y	Y
client_id	<client_id>	N	Y
AuthPrefix	Bearer	N	N
AuthMethod	POST	N	N
ResponseType	JSON	N	N
TokenKey	access_token	N	N



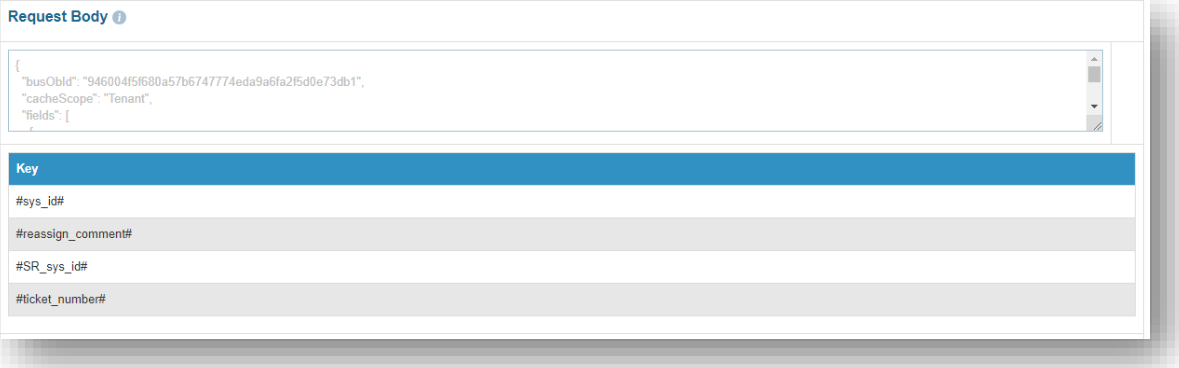
Request Authentication Parameters

Key	Value	Is Encrypted	Is Key	Action
grant_type	password	□	☑	🗑
username	[redacted]	□	☑	🗑
Password	[redacted]	□	☑	🗑
client_id	[redacted]	☑	☑	🗑
AuthPrefix	Bearer	□	□	🗑
AuthMethod	POST	□	□	🗑
ResponseType	JSON	□	□	🗑
TokenKey	access_token	□	□	🗑

Figure 211 – Create Data Source (Request Authentication Parameters)

- **Request Body -** In this section, please enter the request body in JSON format. A sample request is mentioned below:

```
Request Body –
{
  "busObId": "946004f5f680a57b6747774eda9a6fa2f5d0e73db1",
  "cacheScope": "Tenant",
  "fields": [
    {
      "dirty": true,
      "displayName": "Task RecID",
      "fieldId": "94600535397402549 8ed1d4068936d72c8992d015c",
      "value": "#sys_id#"
    },
    {
      "dirty": true,
      "displayName": "Parent RecID",
      "fieldId": "9460053dd53d9888efddc34d3db0360cc5be25f567",
      "value": "#SR_sys_id#"
    },
    {
      "dirty": true,
      "displayName": "Journal Details",
      "fieldId": "946005008899c5f5c31caa43c99083519668f0ff33",
      "value": "#reassign_comment#"
    },
{
      "dirty": true,
      "displayName": "Ticket Number",
      "fieldId": "94602e208e8947bff420df4016b30962152556d5e2",
      "value": "#ticket_number#"
    },
    {
      "dirty": true,
      "displayName": "Assignment Team",
      "fieldId": "946005013472134fdc1b0649a685d41a4c73f6e179",
      "value": "Service Desk"
    },
    {
      "dirty": true,
      "displayName": "Status",
      "fieldId": "946004ff47672c8cda67da43a1945ce56f2f617855",
      "value": "New"
    },
    {
      "dirty": true,
      "displayName": "Task Type",
      "fieldId": "946004feb10853e55a192849c780773b2133028cc0",
      "value": "SR Task"
    },
    {
      "dirty": true,
      "displayName": "Reassigning",
      "fieldId": "946005a199ecde0a9cf0b748bb94e4040c2007540f",
      "value": "True"
    }
  ],
  "persist": true
}
```



Request Body

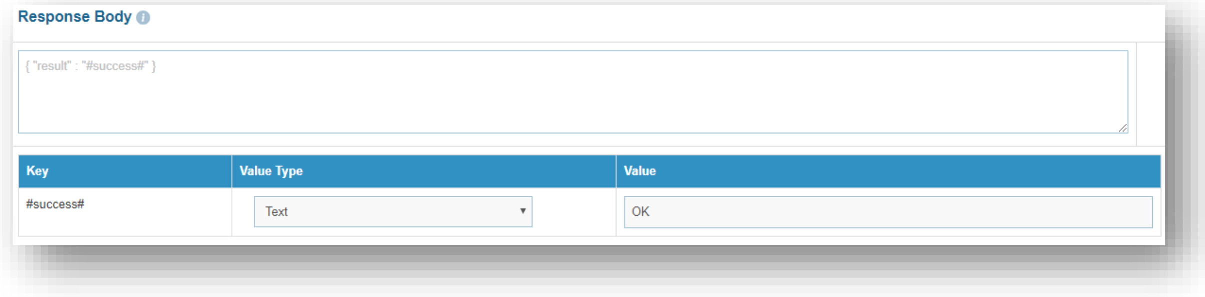
```
{
  "busObId": "946004f5f680a57b6747774eda9a6fa2f5d0e73db1",
  "cacheScope": "Tenant",
  "fields": [
```

Key
#sys_id#
#reassign_comment#
#SR_sys_id#
#ticket_number#

Figure 212 – Release Rules Configuration (Request Body)

- **Response Body –** In this section, please enter the response body in JSON format. A sample request is mentioned below.

```
Response Body –
{ "result" : "#success#" }
```



Response Body

{ "result" : "#success#" }

Key	Value Type	Value
#success#	Text	OK

Figure 213 – Release Rules Configuration (Response Body)

– **Response Key Value** mapping can be done as per the below table.

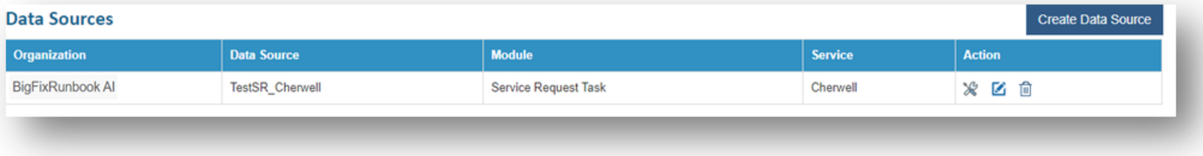
Table 41– Sample Response Key Value Mapping

#success#	Text	OK

– Click **Submit** to add the data source.

– In order to bring the tickets within BigFix Runbook AI scope, a specific queue needs to be configured in the ITSM tool and same has to be configured in BigFix Runbook AI. This is achieved through **Manage the Entry Criteria**. Please perform the below steps:

- Go to Actions tab and click Manage Data Sources.
- On the **Data Sources** tab, click ✂ next to the data source user wants to manage. **Manage Entry Criteria** screen appears.

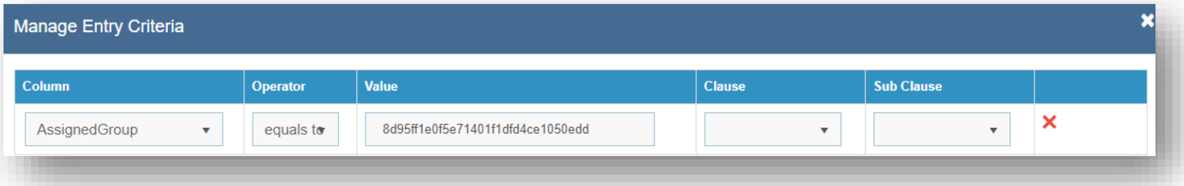


Data Sources

Organization	Data Source	Module	Service	Action
BigFixRunbook AI	TestSR_Cherwell	Service Request Task	Cherwell	

Figure 214 – Manage Entry Criteria

- Select 'AssignedGroup' for the **Column** field and 'equals to' for the **Operator** field.
- Enter the sys_id of the assignment group in Cherwell in the **Value** field.
- **Clause** and **Sub-Clause** fields can also be added based on requirement.



Manage Entry Criteria

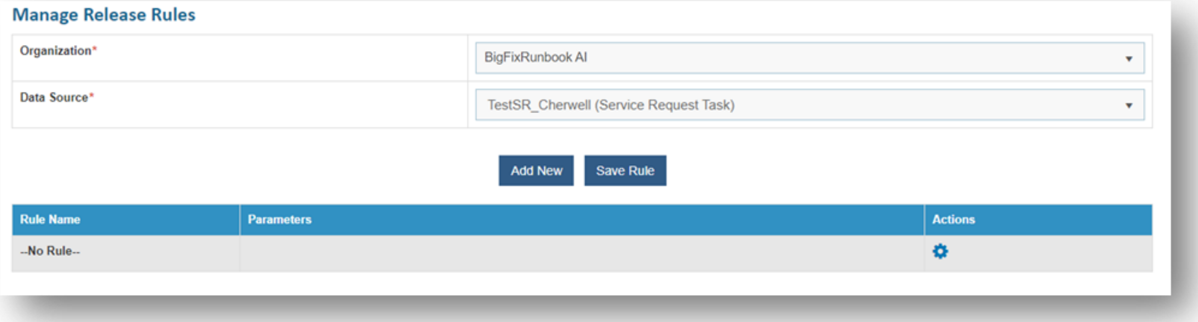
Column	Operator	Value	Clause	Sub Clause	
AssignedGroup	equals to	8d95ff1e0f5e71401f1dfd4ce1050edd			✕

Figure 215 – Manage Entry Criteria (cont.)

- Click **Save**.

– To configure the Release rules for the data source created earlier, perform the below steps:

- Go to Actions Tab → Runbooks and click Manage Rules.
- Select the **Organization** and the data source created from **Data Source** dropdown.



Manage Release Rules

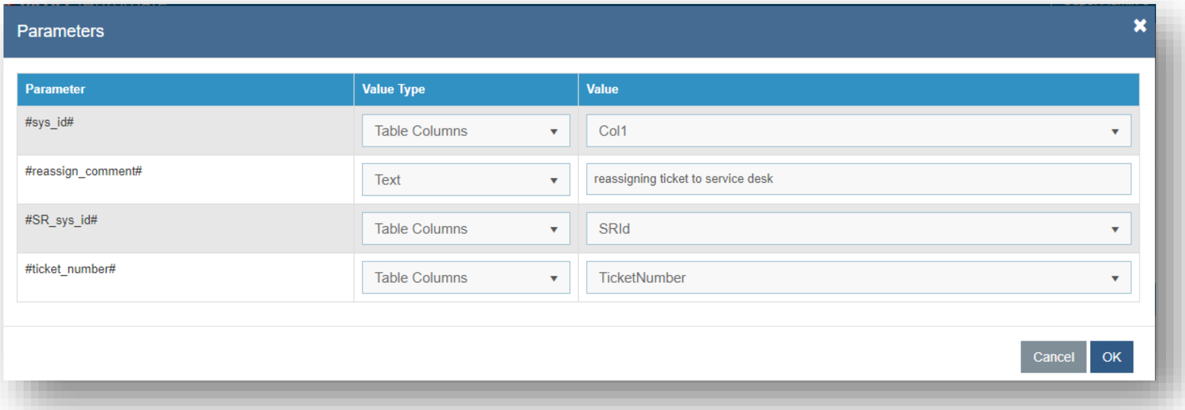
Organization*: BigFixRunbook AI

Data Source*: TestSR_Cherwell (Service Request Task)

Rule Name	Parameters	Actions
--No Rule--		⚙

Figure 216 – Manage Release Rules

- Click on ⚙ corresponding to **–No Rule—**.
- Map the parameters #sys_id# to the column in which sys_id was mapped while performing the mandatory parameter mapping while data source creation.
- Mention the reason for releasing ticket in #reassign_comments#.
- Map # SR_sys_id # again to the column in which SRId was mapped while performing the mandatory parameter mapping while data source creation.

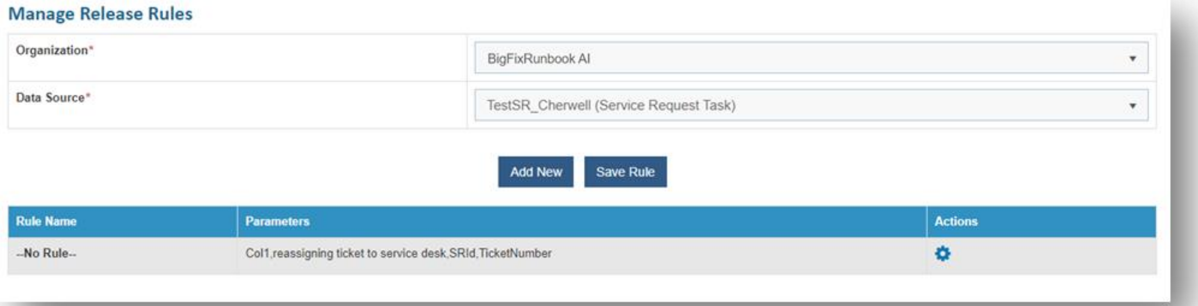


Parameters

Parameter	Value Type	Value
#sys_id#	Table Columns	Col1
#reassign_comment#	Text	reassigning ticket to service desk
#SR_sys_id#	Table Columns	SRId
#ticket_number#	Table Columns	TicketNumber

Figure 217 – Manage Release Rules (cont.)

- Click **OK**.



Manage Release Rules

Organization*	BigFixRunbook AI
Data Source*	TestSR_Cherwell (Service Request Task)

Rule Name	Parameters	Actions
--No Rule--	Col1,reassigning ticket to service desk,SRId,TicketNumber	⚙

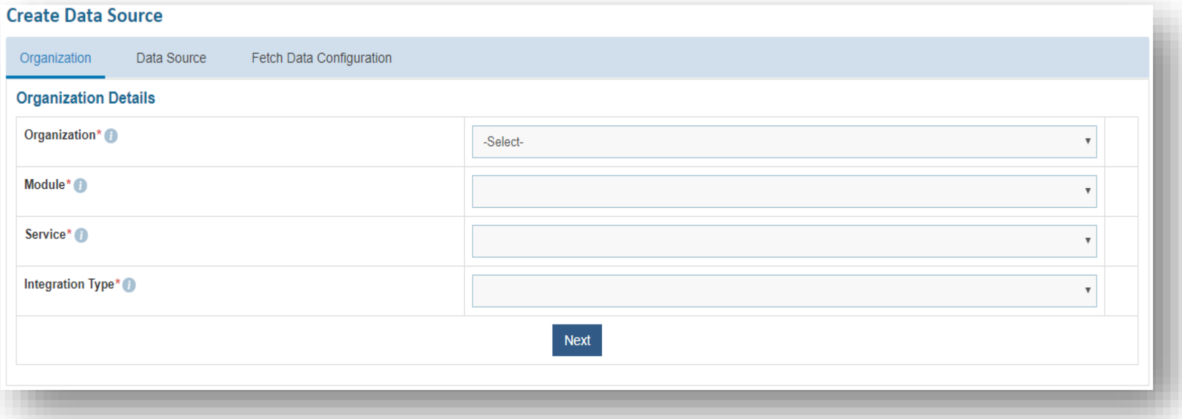
Figure 218 – Manage Release Rules (cont.)

- Click Save Rule.

4.4.3 Change Request Task Management

To create a data source for Change Request Task Management, perform the following steps:

- On the main menu bar, click **Actions tab → Manage Data Sources**.
- The **Create Data Source** page appears with the following tabs:
  - Organization
  - Data Source
  - Fetch Data Configuration
  - Release Rules Configuration



Create Data Source

Organization Details

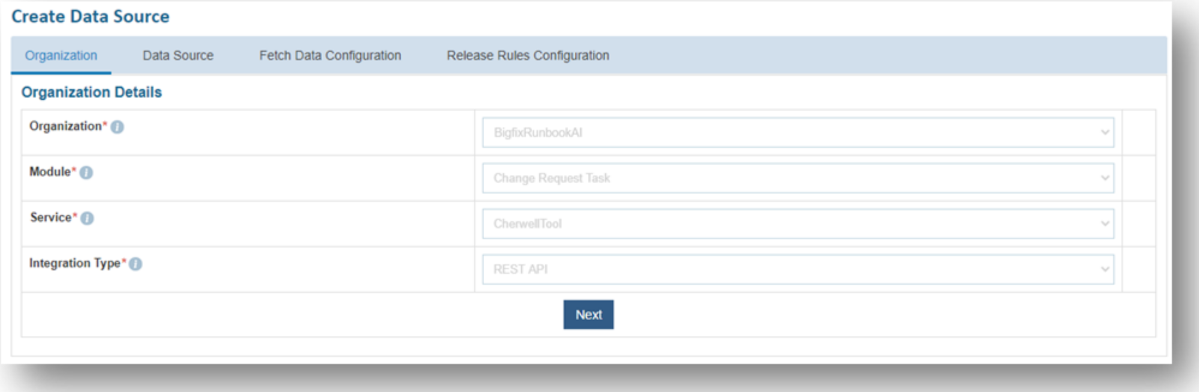
Organization*	-Select-
Module*	
Service*	
Integration Type*	

Figure 219 – Create Data Source

Release Rules Configuration is only applicable for the following **Module** types- **Incident Management, Change Request Task and Service Request Task.** This tab will not be activated for other module types.

– On the **Organization** tab,

- Select the **Organization Name** from the dropdown.
- Select the **Module** as **Change Request Task** since we are configuring this data source for pulling the change request task tickets.
- Select the **Service** as **Cherwell Tool** as we are configuring the data source for Cherwell
- Select the **Integration Type** as **REST**, since we will be integrating through REST APIs.
- Click **Next**.



Create Data Source

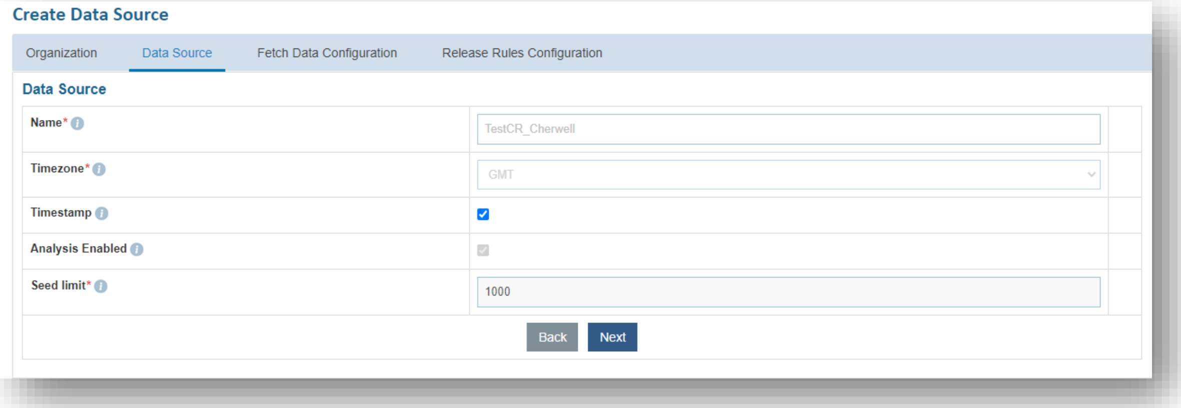
Organization Details

Organization*	BigfixRunbookAI
Module*	Change Request Task
Service*	CherwellTool
Integration Type*	REST API

Figure 220 – Create Data Source (cont.)

– On the **Data Source** tab,

- Type the new data source in the **Name** field.
- Select the **Timezone** to specify the time zone of the selected data source.
- Select **Timestamp** to view the present data with date and time.
- Select **Analysis Enabled?** if user wants to analyze the data retrieved from the data source.
- Click Next.



Name*	TestCR_Cherwell
Timezone*	GMT
Timestamp	☑
Analysis Enabled	☑
Seed limit*	1000

Figure 221 – Create Data Source (cont.)

– On the **Fetch Data Configuration** tab, type in the details as per the environment.

– In the **Connection Details** section, enter the following details:

- **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.
- **Sample URL** - http://<iAutomate_API_URL>/iAutomateAPI/Request/GetChangeTicketData/<Org_ID>?start_date>=#Start_Date#&end_date<=#End_Date#&
- Here, < iAutomate_API_URL > is the API URL of BigFix Runbook AI where Push APIs are present and <Org_ID> is the OrgID for the organization for which you are creating the data source. It is available in Organization Master in Database.
- **Authentication Type** – Select one of the Authentication Types from Basic / Windows, JWT, OAuth 2.0

The user details that are entered here should be an API User

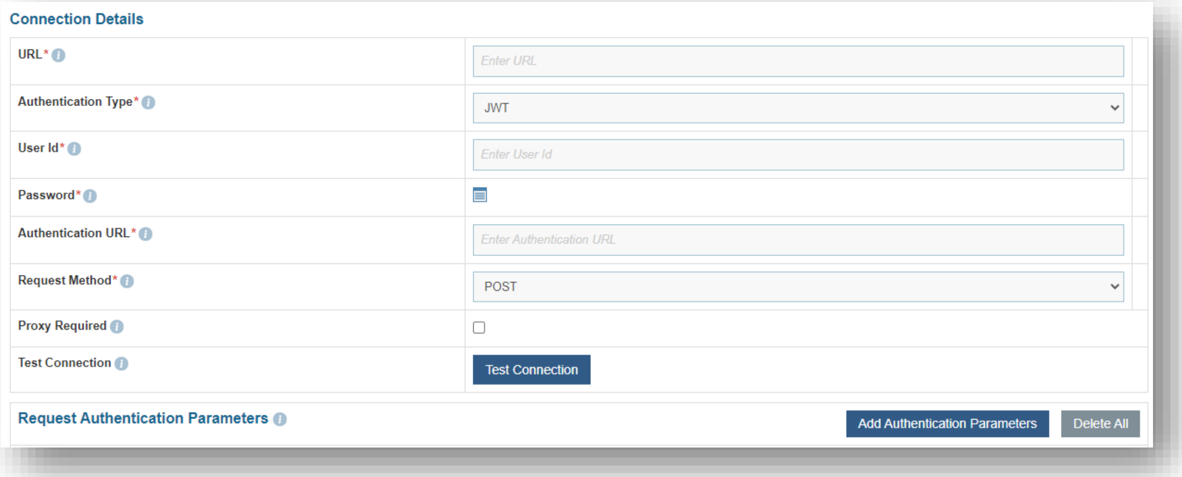
Selection of **Basic / Windows** requires you to enter -

- User Id
- Password.

Selection of **JWT / OAuth 2.0** requires you to enter -

- User Id
- Password
- Authentication URL

- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



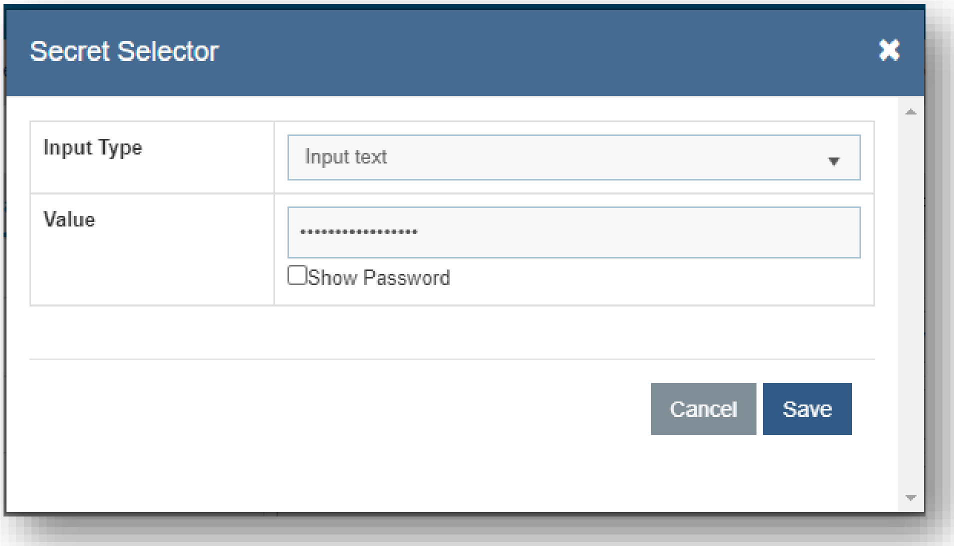
Connection Details

URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	POST
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

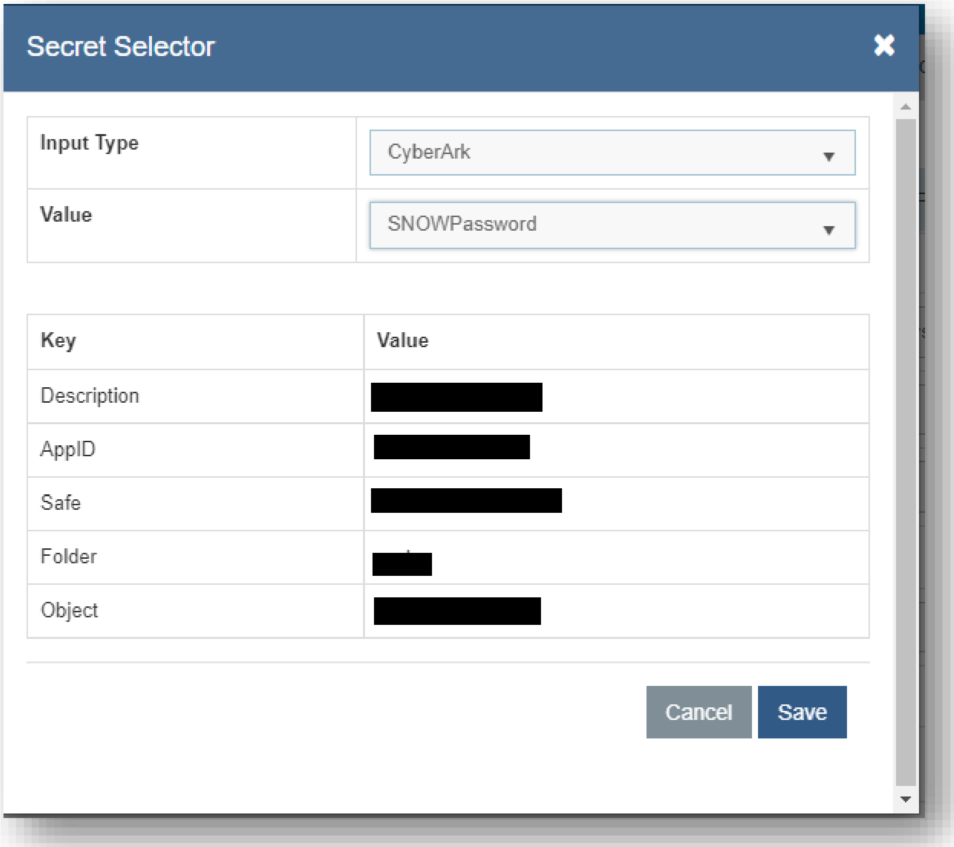
Figure 222 – Create Data Source (Connection Details)

- For **Password**, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Input Type	Input text
Value	•••••••••••••••• ☐ Show Password

Figure 223 – Password in plaintext



Input Type	CyberArk
Value	SNOWPassword

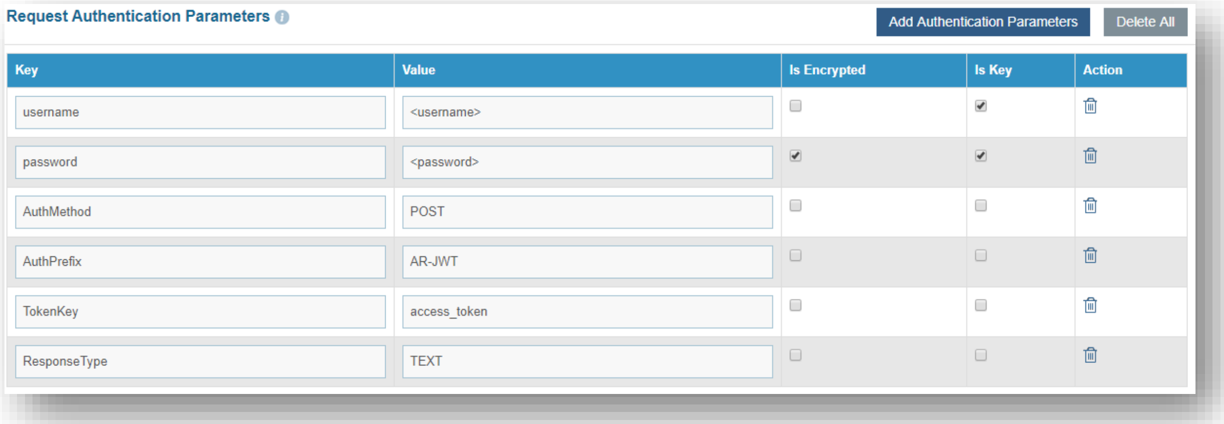
Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 224 – Password from Key Vault (CyberArk)

- **Request Authentication Parameters** – If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.
- Based on the **Authentication Type**, add the parameters mentioned in the below table.

Table 42– Sample Authentication Parameters

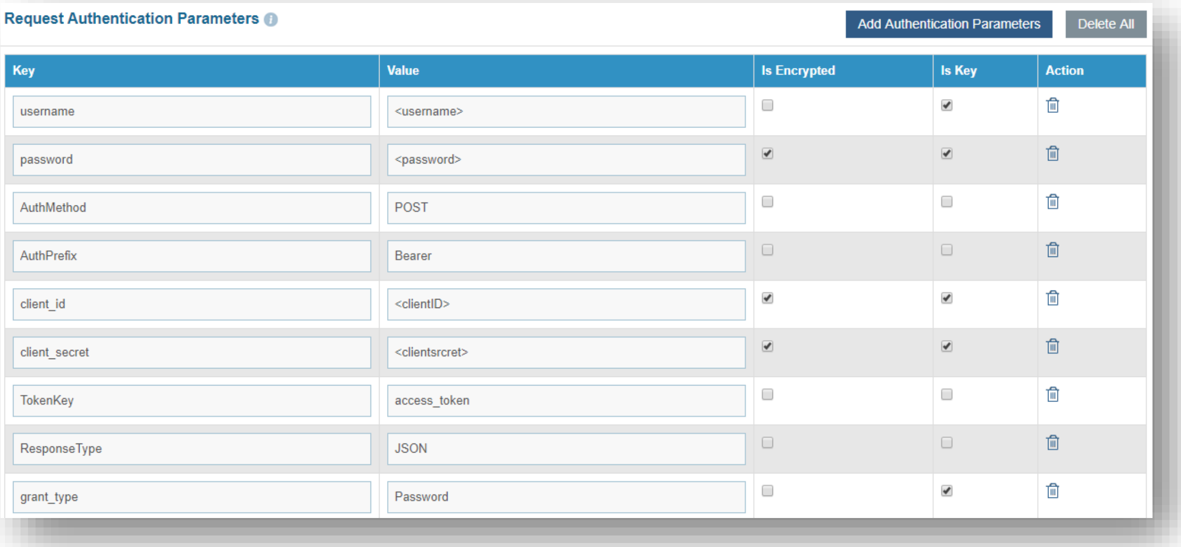
Authentication Type	Key	Value	Is Encrypted?	Is Key?
JWT	username	<username>	NO	YES
JWT	password	<password>	YES	YES
JWT	AuthMethod	POST	NO	NO
JWT	AuthPrefix	AR-JWT	NO	NO
JWT	TokenKey	access_token	NO	NO
JWT	ResponseType	TEXT	NO	NO
OAuth2.0	username	<username	NO	YES
OAuth2.0	password	<password>	YES	YES
OAuth2.0	AuthMethod	POST	NO	NO
OAuth2.0	AuthPrefix	Bearer	NO	NO
OAuth2.0	client_id	<clientID>	YES	YES
OAuth2.0	client_secret	<clientsecret>	YES	YES
OAuth2.0	TokenKey	access_token	NO	NO
OAuth2.0	ResponseType	JSON	NO	NO
OAuth2.0	grant_type	Password	NO	YES



Request Authentication Parameters

Key	Value	Is Encrypted	Is Key	Action
username	<username>	□	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	□	□	🗑
AuthPrefix	AR-JWT	□	□	🗑
TokenKey	access_token	□	□	🗑
ResponseType	TEXT	□	□	🗑

Figure 225 – Create Data Source (Request Authentication Parameters for JWT)



Request Authentication Parameters

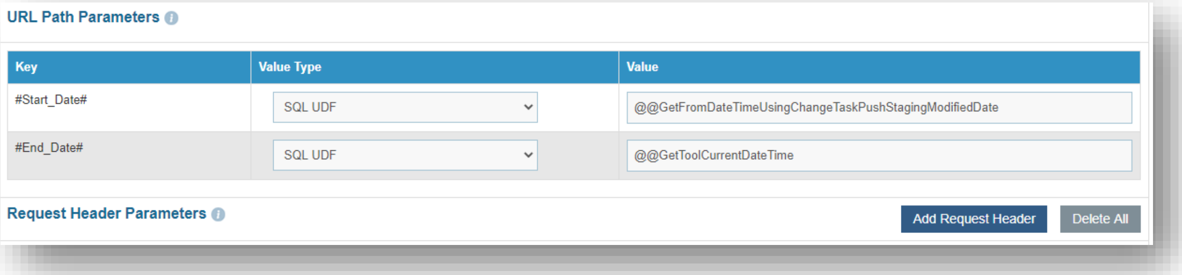
Key	Value	Is Encrypted	Is Key	Action
username	<username>	□	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	□	□	🗑
AuthPrefix	Bearer	□	□	🗑
client_id	<clientID>	☑	☑	🗑
client_secret	<clientsrcret>	☑	☑	🗑
TokenKey	access_token	□	□	🗑
ResponseType	JSON	□	□	🗑
grant_type	Password	□	☑	🗑

Figure 226 – Create Data Source (Request Authentication Parameters for OAuth2.0)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs.

```
Key: #StartDate#

ValueType: SQL UDF

VALUE: @@GetFromDateTimeUsingChangeTaskPushStagingModifiedDate
```

```
Key: #EndDate#

ValueType: SQL UDF

VALUE: @@GetToolCurrentDateTime
```



URL Path Parameters

Key	Value Type	Value
#Start_Date#	SQL UDF	@@GetFromDateTimeUsingChangeTaskPushStagingModifiedDate
#End_Date#	SQL UDF	@@GetToolCurrentDateTime

Request Header Parameters

Figure 227– URL Path Parameters

– **Request Header Parameters –** Please enter the request header parameters as required.

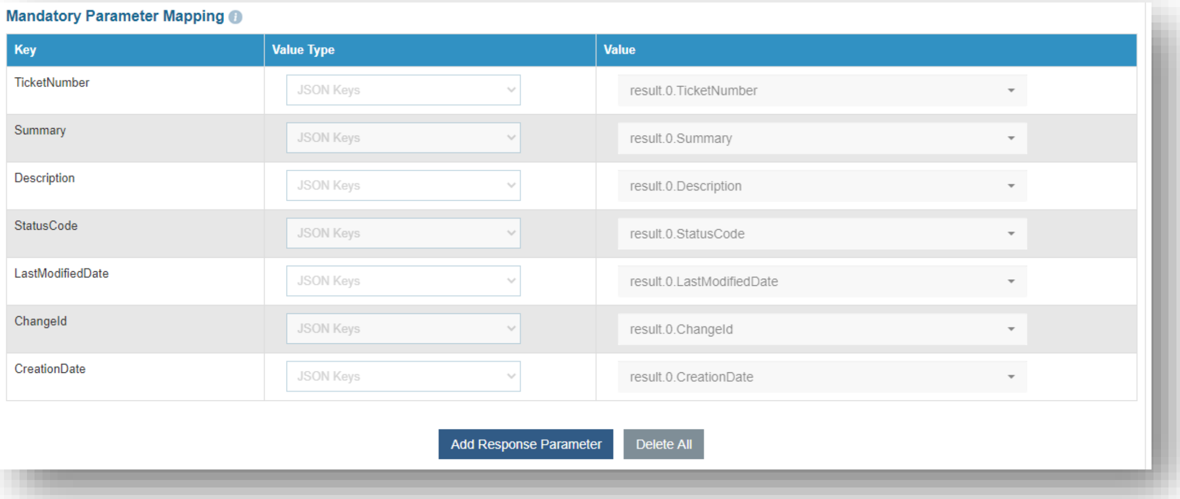
– **Response Body –** In this section, please enter the output of URL query for one of the incidents in JSON format. A sample response is mentioned below.

```
Response Body –

{

    "result": [

        {

            "TicketNumber": "12662",

            "Summary": "Test Task",

            "Description": "Test Task",

            "AssignedGroup":
"945e4f5b7ba0108fd5ba6d4685ab66fce83af21369",

            "ChangeId":
"945f06a5aeb28c6a4fd6c4488a860863594361e721",

            "StatusCode": "1",

            "LastModifiedDate": "2020-05-13 05:11:47.000",

            "sys_id":
"945f06b5cf9a2367a851ef48c99e87910fbd656fcf",

            "CreationDate": "2020-05-13 05:08:10.000",

            "Col1": "",

            "Col2": "",

            "Col3": "",

            "Col4": "",

            "Col5": "",

            "iAutomate_CreatedDateInGMT": "2020-05-13
05:29:47.987",

            "iAutomate_UpdatedDateInGMT": "2020-05-13
05:29:47.987"

        }
        ]
  }
```

- After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section**.**
- **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below.

Table 43– Sample Mandatory Parameter Mapping

Key	Value Type	Value
TicketNumber	JSON.Keys	result.0.TicketNumber
Summary	JSON.Keys	result.0.Summary
Description	JSON.Keys	result.0.Description
StatusCode	JSON.Keys	result.0.StatusCode
LastModifiedDate	JSON.Keys	result.0.LastModifiedDate
ChangeId	JSON.Keys	result.0.ChangeId
CreationDate	JSON.Keys	result.0.CreationDate



Mandatory Parameter Mapping

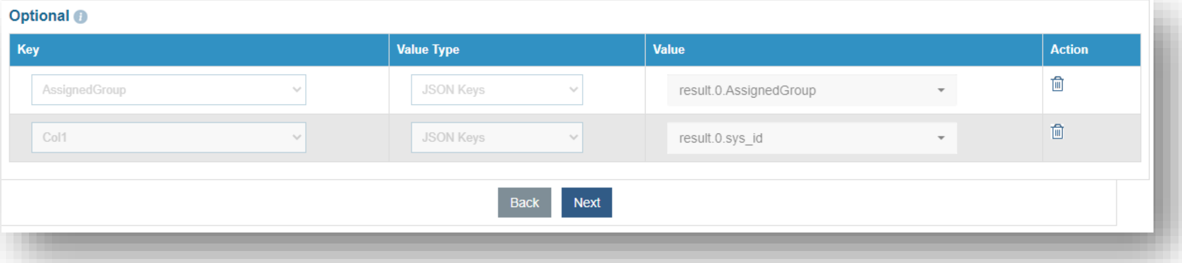
Key	Value Type	Value
TicketNumber	JSON Keys	result.0.TicketNumber
Summary	JSON Keys	result.0.Summary
Description	JSON Keys	result.0.Description
StatusCode	JSON Keys	result.0.StatusCode
LastModifiedDate	JSON Keys	result.0.LastModifiedDate
ChangeId	JSON Keys	result.0.ChangeId
CreationDate	JSON Keys	result.0.CreationDate

Figure 228 – Mandatory Parameter Mapping

- If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 44– Sample Optional Parameters

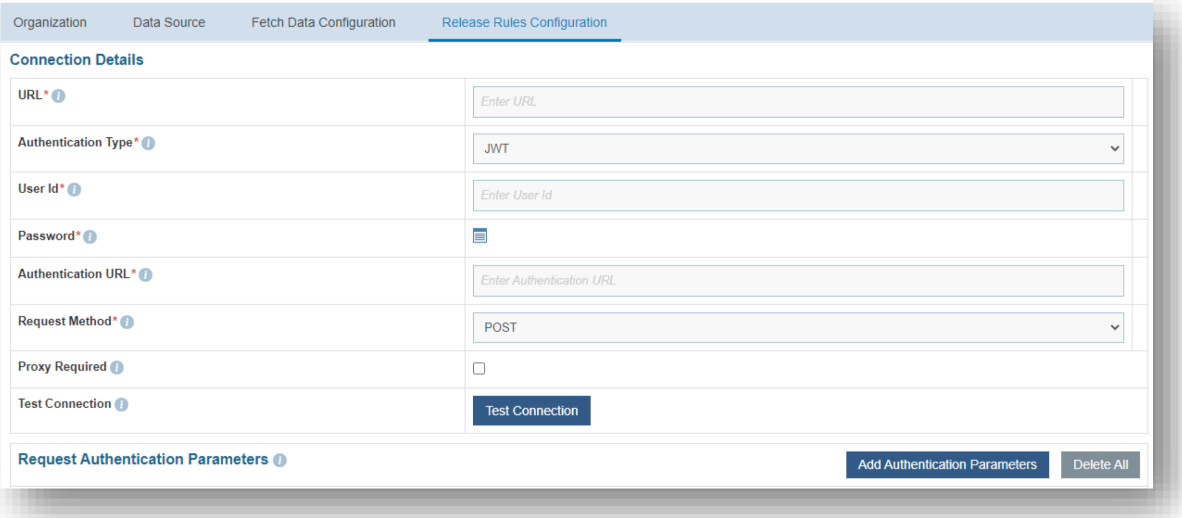
Key	Value Type	Value
AssignedGroup	JSON.Keys	result.0.AssignedGroup
Col1	JSON.Keys	result.0.sys_id


Optional

Key	Value Type	Value	Action
AssignedGroup	JSON Keys	result.0.AssignedGroup	🗑
Col1	JSON Keys	result.0.sys_id	🗑

Figure 229 – Optional Parameter Mapping

- Click Next to proceed to Release Rules Configuration.
- On **Release Rules Configuration** tab, type in the details as per the requirement.
- In the **Connection Details** section, enter the following details:
    - **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.
    - **Sample URL** - https://<url>. cherwellondemand.com/CherwellAPI/api/V1/savebusinessobjectbatch
    - **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously. For e.g., **JWT**.
    - **Request Method** – Select Request Method as POST from the drop-down.
    - **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
    - Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



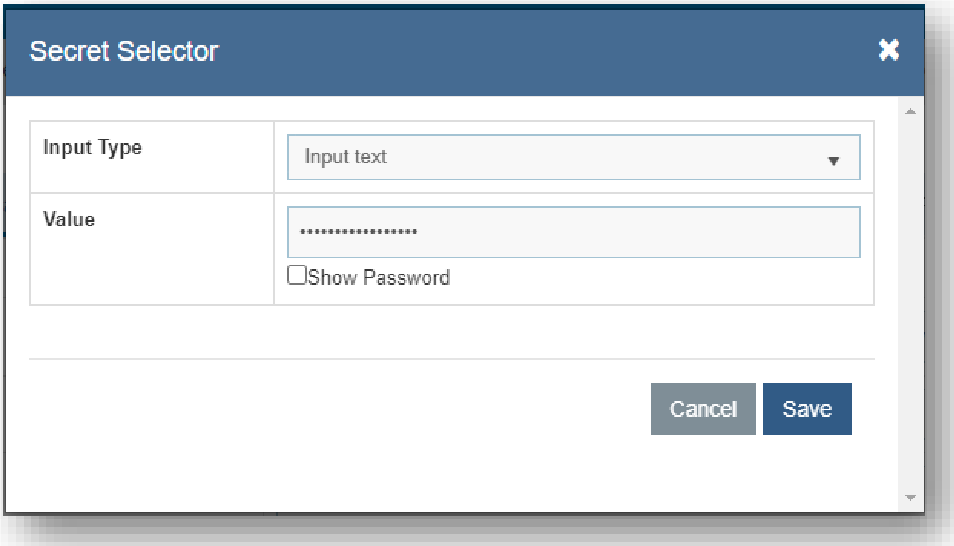
Connection Details

URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	POST
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

Figure 230 – Release Rules Configuration (Connection Details)

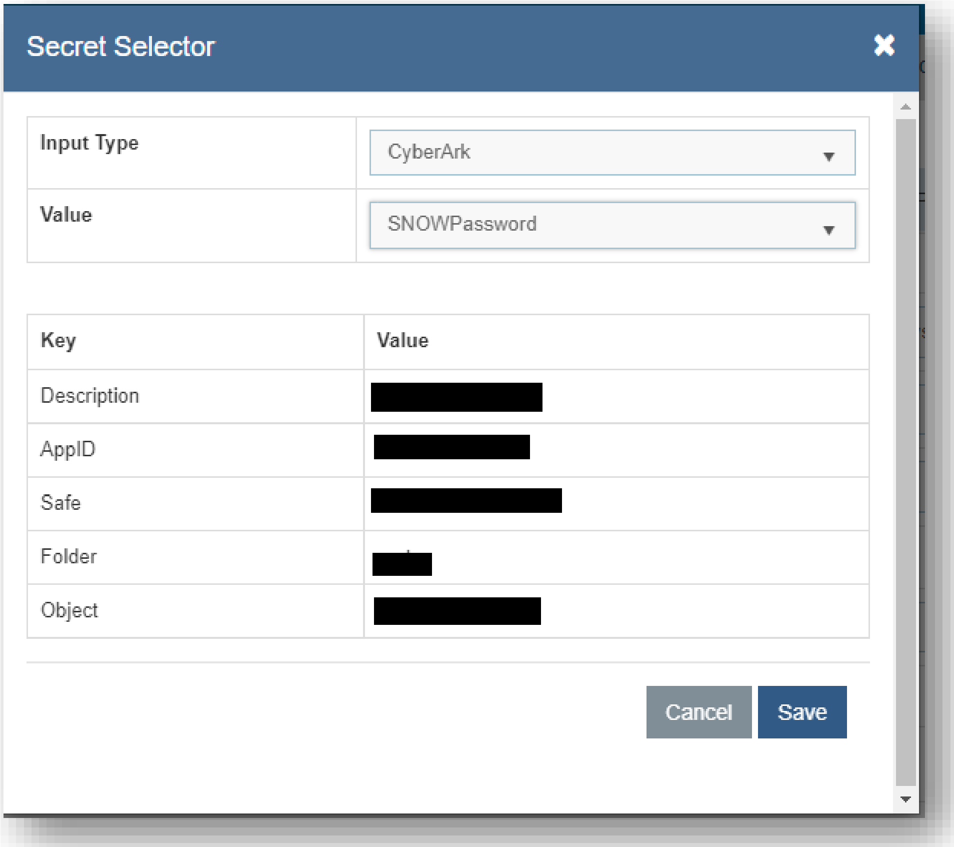
- For **Password**, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Secret Selector

Input Type	Input text
Value	•••••••••••••••• □Show Password

Figure 231 – Password in plaintext



Secret Selector

Input Type	CyberArk
Value	SNOWPassword

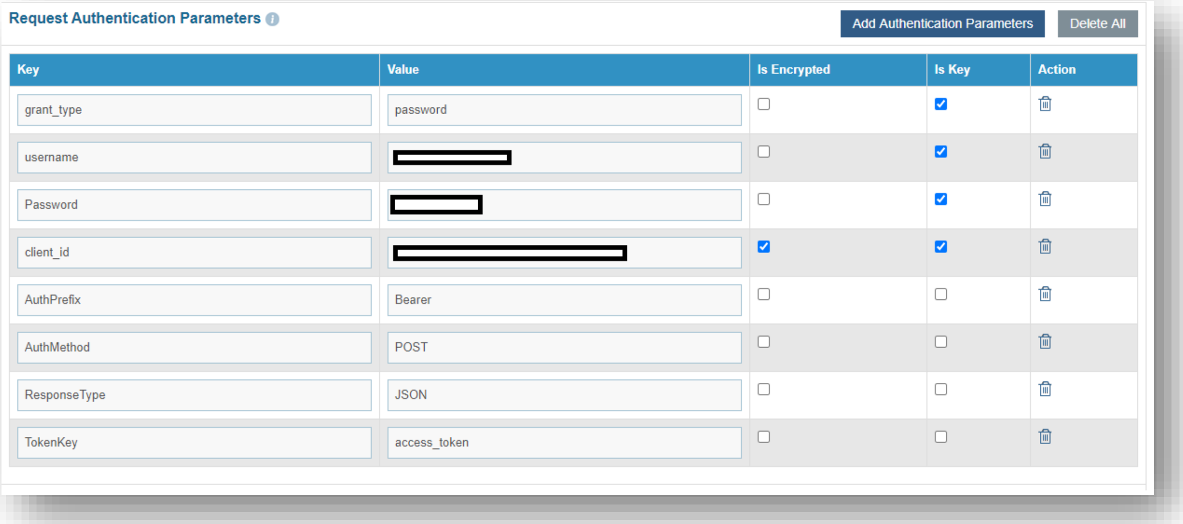
Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 232 – Password from Key Vault (CyberArk)

- Request Authentication Parameters - If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.
- Based on the Authentication Type, **JWT**, add the parameters mentioned in the below table

Table 45– Sample Authentication Parameters

Key	Value	Is Encrypted?	Is Key?
grant_type	password	N	Y
username	<username>	N	Y
Password	<password>	Y	Y
client_id	<client_id>	N	Y
AuthPrefix	Bearer	N	N
AuthMethod	POST	N	N
ResponseType	JSON	N	N
TokenKey	access_token	N	N



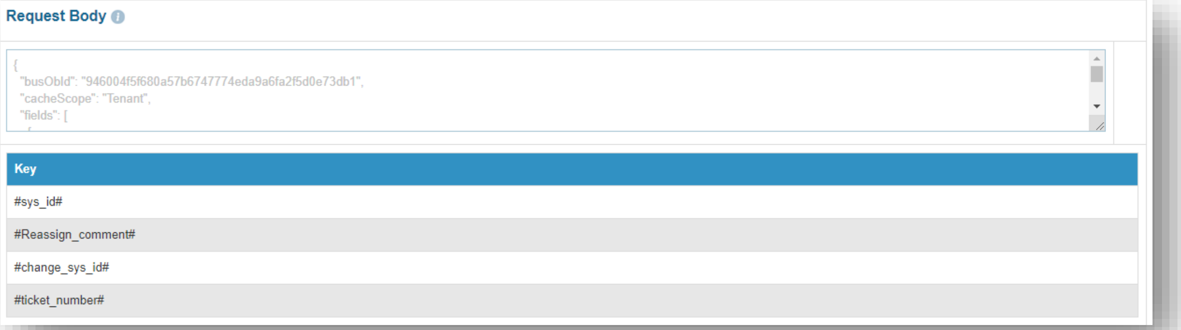
Request Authentication Parameters
Add Authentication Parameters
Delete All

Key	Value	Is Encrypted	Is Key	Action
grant_type	password	☐	☑	🗑
username	[redacted]	☐	☑	🗑
Password	[redacted]	☐	☑	🗑
client_id	[redacted]	☑	☑	🗑
AuthPrefix	Bearer	☐	☐	🗑
AuthMethod	POST	☐	☐	🗑
ResponseType	JSON	☐	☐	🗑
TokenKey	access_token	☐	☐	🗑

Figure 233 – Create Data Source (Request Authentication Parameters)

– **Request Body -** In this section, please enter the request body in JSON format. A sample request is mentioned below:

```
Request Body –
{
  "busObId": "946004f5f680a57b6747774eda9a6fa2f5d0e73db1",
  "cacheScope": "Tenant",
  "fields": [
    {
      "dirty": true,
      "displayName": "Task RecID",
      "fieldId": "9460053539740254498ed1d4068936d72c8992d015c",
      "value": "#sys_id#"
    },
 {
      "dirty": true,
      "displayName": "Ticket Number",
      "fieldId": "94602e208e8947bff420df4016b30962152556d5e2",
      "value": "#ticket_number#"
    },
    {
      "dirty": true,
      "displayName": "Parent RecID",
      "fieldId": "9460053dd53d9888efddc34d3db0360cc5be25f567",
      "value": "#change_sys_id#"
    },
    {
      "dirty": true,
      "displayName": "Journal Details",
      "fieldId": "946005008899c5f5c31caa43c99083519668f0ff33",
      "value": "#Reassign_comment#"
    },
    {
      "dirty": true,
      "displayName": "Assignment Team",
      "fieldId": "946005013472134fdc1b0649a685d41a4c73f6e179",
      "value": "GBP Change Management"
    },
    {
      "dirty": true,
      "displayName": "Status",
      "fieldId": "946004ff47672c8cda67da43a1945ce56f2f617855",
      "value": "Acknowledged"
    },
    {
      "dirty": true,
      "displayName": "Task Type",
      "fieldId": "946004feb10853e55a192849c780773b2133028cc0",
      "value": "Change Task"
    },
    {
      "dirty": true,
      "displayName": "Reassigning",
      "fieldId": "946005a199ecde0a9cf0b748bb94e4040c2007540f",
      "value": "True"
    }
  ],
  "persist": true
}
```



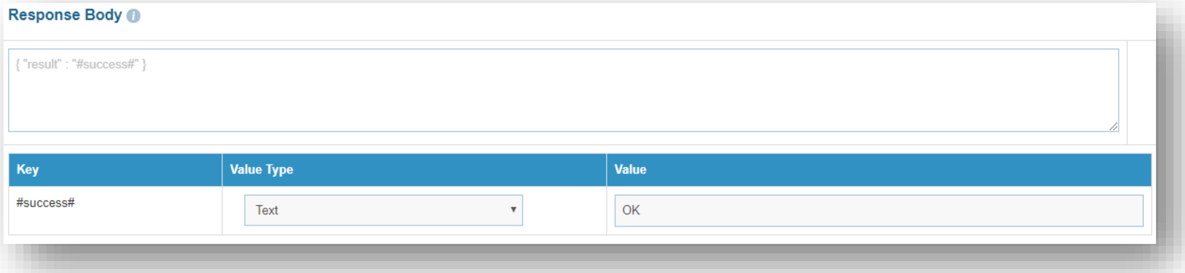
```
{
  "busObId": "946004f5f680a57b6747774eda9a6fa2f5d0e73db1",
  "cacheScope": "Tenant",
  "fields": [
```

Key
#sys_id#
#Reassign_comment#
#change_sys_id#
#ticket_number#

Figure 234 – Release Rules Configuration (Request Body)

- **Response Body –** In this section, please enter the response body in JSON format. A sample request is mentioned below.

```
Response Body –
{ "result" : "#success#" }
```



Response Body

{ "result" : "#success#" }

Key	Value Type	Value
#success#	Text	OK

Figure 235 – Release Rules Configuration (Response Body)

– **Response Key Value** mapping can be done as per the below table.

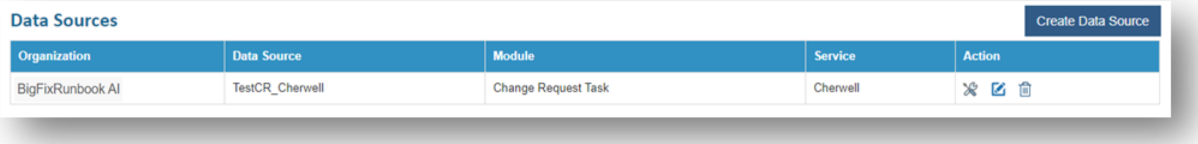
Table 46– Sample Response Key Value Mapping

#success#	Text	OK

– Click **Submit** to add the data source.

– In order to bring the tickets within BigFix Runbook AI scope, a specific queue needs to be configured in the ITSM tool and same has to be configured in BigFix Runbook AI. This is achieved through **Manage the Entry Criteria**. Please perform the below steps:

- Go to Actions tab and click Manage Data Sources.
- On the **Data Sources** tab, click ⚒ next to the data source user wants to manage. **Manage Entry Criteria** screen appears.

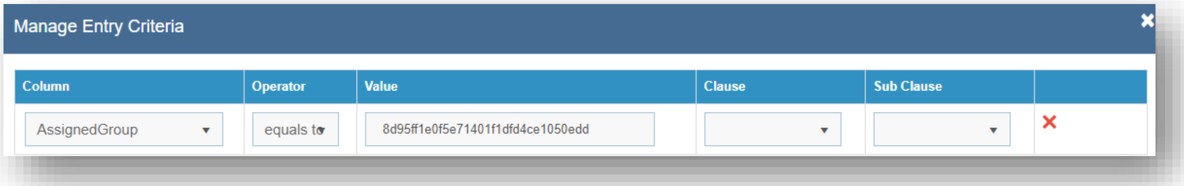


Data Sources

Organization	Data Source	Module	Service	Action
BigFixRunbook AI	TestCR_Cherwell	Change Request Task	Cherwell	⚒ ✎ 🗑

Figure 236 – Manage Entry Criteria

- Select 'AssignedGroup' for the **Column** field and 'equals to' for the **Operator** field.
- Enter the sys_id of the assignment group in Cherwell in the **Value** field.
- **Clause** and **Sub-Clause** fields can also be added based on requirement.



Manage Entry Criteria

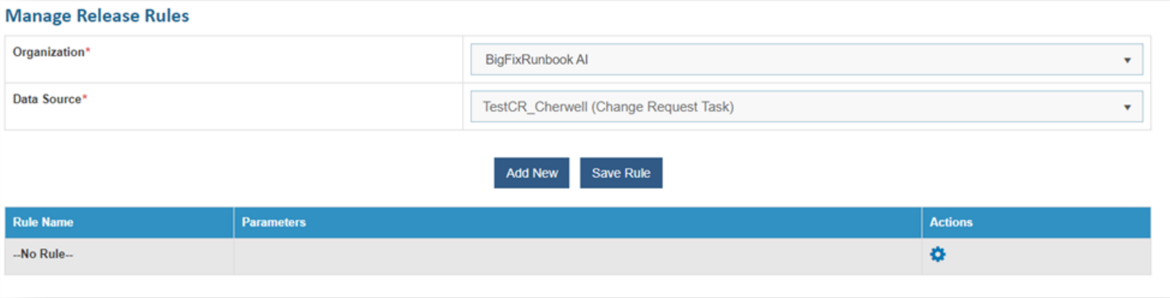
Column	Operator	Value	Clause	Sub Clause	
AssignedGroup ▾	equals to ▾	8d95ff1e0f5e71401f1dfd4ce1050edd	▾	▾	✕

Figure 237 – Manage Entry Criteria (cont.)

- Click **Save**.

– To configure the Release rules for the data source created earlier, perform the below steps:

- Go to **Actions tab → Runbooks** and click Manage Rules.
- Select the **Organization** and the data source created from **Data Source** dropdown.



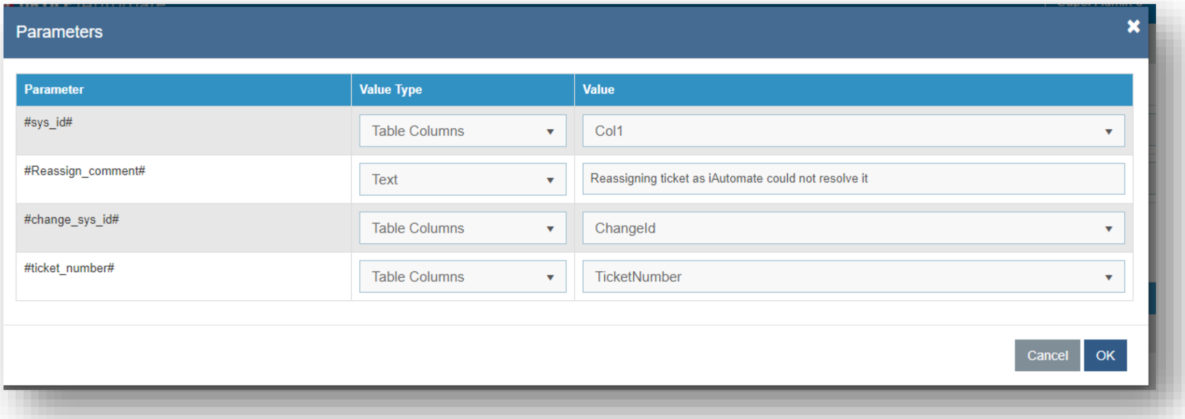
Manage Release Rules

Organization*	BigFixRunbook AI ▾
Data Source*	TestCR_Cherwell (Change Request Task) ▾

Rule Name	Parameters	Actions
--No Rule--		⚙

Figure 238 – Manage Release Rules

- Click on ⚙ corresponding to **–No Rule—**.
- Map the parameters #sys_id# to the column in which sys_id was mapped while performing the mandatory parameter mapping while data source creation.
- Mention the reason for releasing ticket in #reassign_comments#.
- Map #change_sys_id # again to the column in which ChangeId was mapped while performing the mandatory parameter mapping while data source creation.

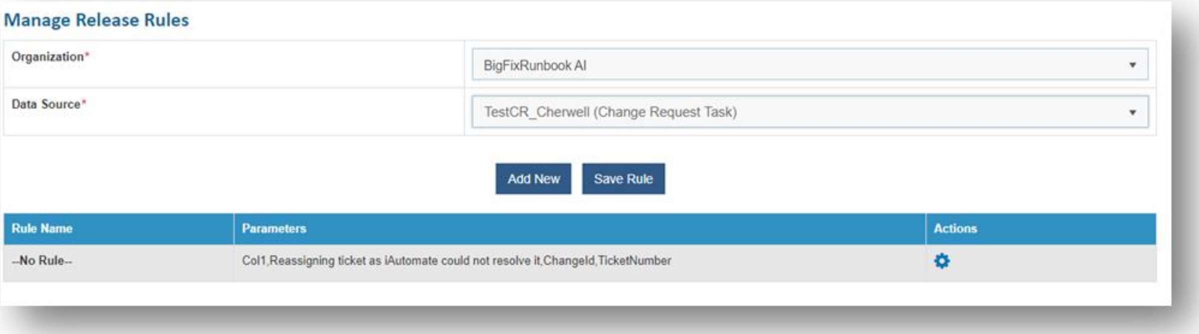


Parameters

Parameter	Value Type	Value
#sys_id#	Table Columns	Col1
#Reassign_comment#	Text	Reassigning ticket as iAutomate could not resolve it
#change_sys_id#	Table Columns	ChangeId
#ticket_number#	Table Columns	TicketNumber

Figure 239 – Manage Release Rules (cont.)

- Click **OK**.



Manage Release Rules

Organization*	BigFixRunbook AI
Data Source*	TestCR_Cherwell (Change Request Task)

Rule Name	Parameters	Actions
--No Rule--	Col1,Reassigning ticket as iAutomate could not resolve it,ChangeId,TicketNumber	⚙

Figure 240 – Manage Release Rules (cont.)

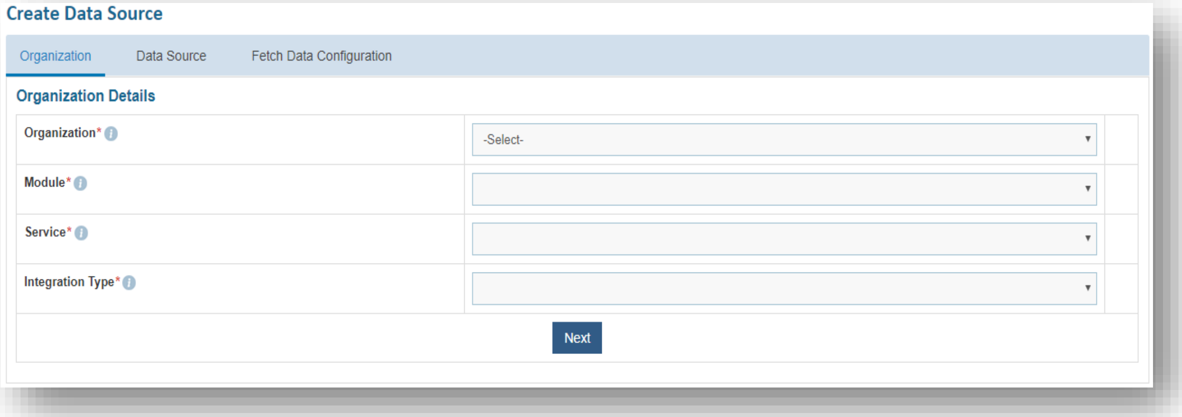
- Click Save Rule.

4.5 Integration with BMC Remedyforce

4.5.1 Incident Management

To create a data source for Incident Management, perform the following steps:

- On the main menu bar, click **Actions → Manage Data Sources**.
- The **Create Data Source** page appears with the following tabs:
  - Organization
  - Data Source
  - Fetch Data Configuration
  - Manage Rules Configuration

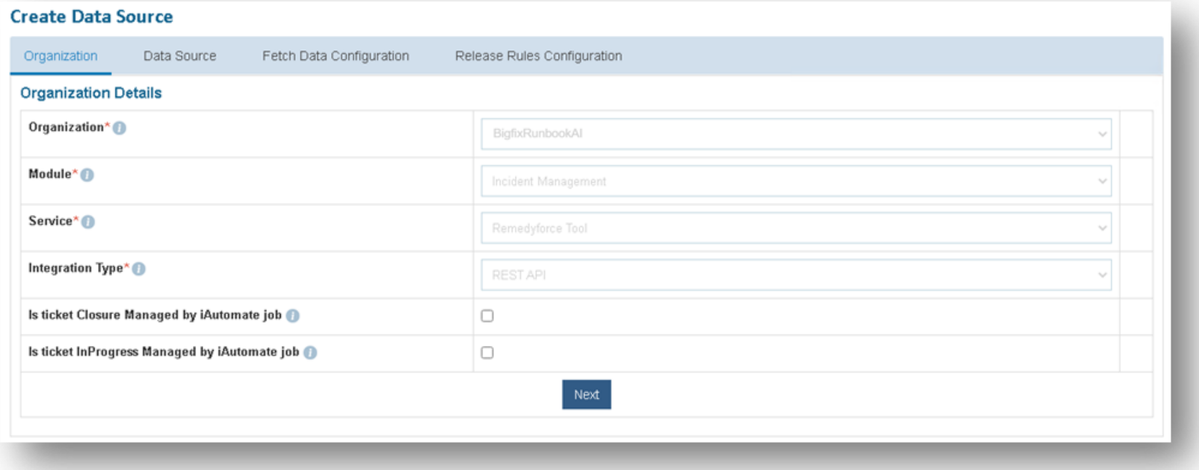


Organization*	-Select-
Module*	
Service*	
Integration Type*	

Figure 241 - Create Data Source

Release Rules Configuration is only applicable for the following **Module** types- **Incident Management, Change Request Task and Service Request Task.** This tab will not be activated for other module types.

- On the **Organization** tab,
  - Select the **Organization Name** from the dropdown.
  - Select the **Module** as **Incident Management,** since we are configuring this data source for pulling the incident tickets.
  - Select the **Service** as **Remedyforce Tool** as we are configuring the data source for BMC Remedyforce.
  - Select the **Integration Type** as **REST**, since we will be integrating through REST APIs.
  - Click **Next**.



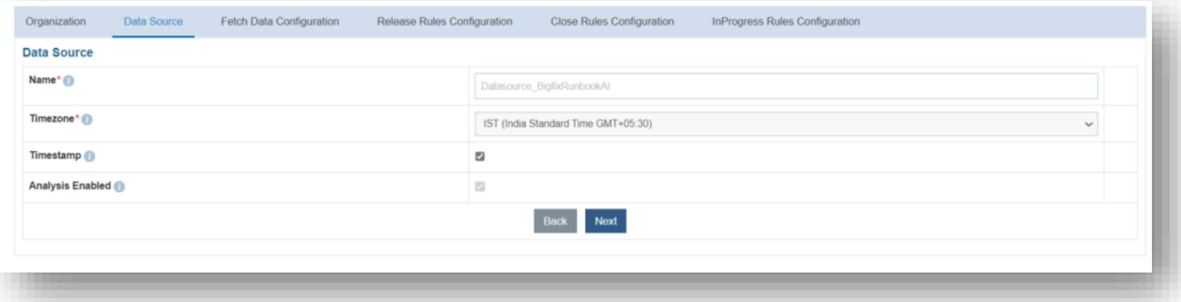
Create Data Source

Organization Details

Organization*	BigfixRunbookAI
Module*	Incident Management
Service*	Remedyforce Tool
Integration Type*	REST API
Is ticket Closure Managed by iAutomate job	□
Is ticket InProgress Managed by iAutomate job	□

Figure 242 - Create Data Source (cont.)

- On the **Data Source** tab,
  - Type the new data source in the **Name** field.
  - Select the **Timezone** to specify the time zone of the selected data source.
  - Select **Timestamp** to view the present data with date and time.
  - Select **Analysis Enabled** if user wants to analyze the data retrieved from the data source.
  - Click Next.



Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

Data Source

Name*	Datasource_BigfixRunbookAI
Timezone*	IST (India Standard Time GMT+05:30)
Timestamp	☑
Analysis Enabled	☑

Figure 243 - Create Data Source (cont.)

- On the **Fetch Data Configuration** tab, type in the details as per the environment.
- In the **Connection Details** section, enter the following details:
  - **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.

- **Sample URL** - https://localhost/services/data/v45.0/query?q=SELECT+#Fields#+from+BMCServiceDesk__Incident__c+WHERE+BMCServiceDesk__queueName__c+=+'#AssignmentGroup#'+AND+BMCServiceDesk__Status_ID__c+IN+(#State#)
- **Authentication Type** – Select one of the Authentication Types from Basic / Windows, JWT, OAuth 2.0

  Selection of **Basic / Windows** requires you to enter -

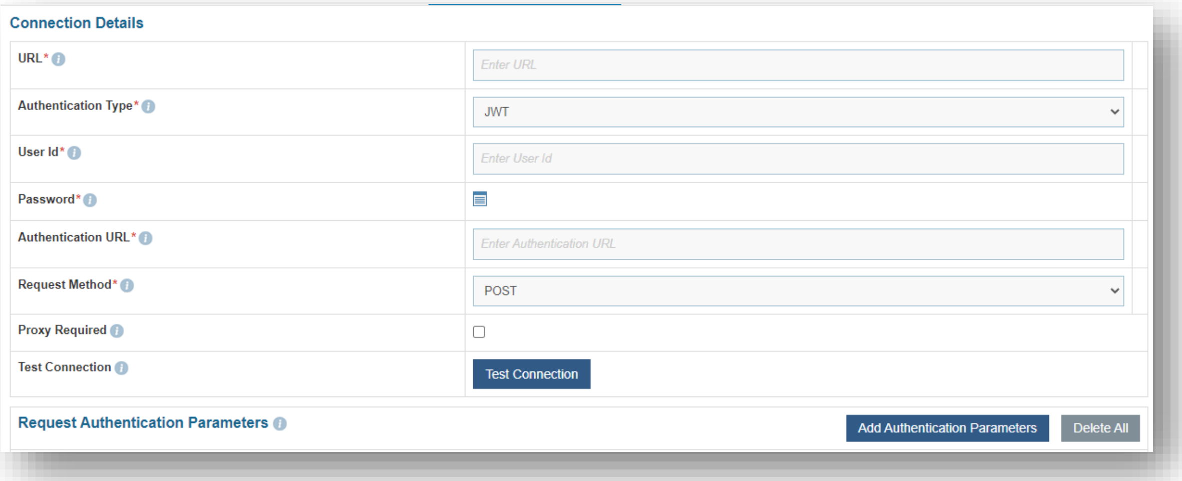
  - User Id
  - Password.

  Selection of **JWT / OAuth 2.0** requires you to enter -

  - User Id
  - Password
  - Authentication URL

  Here, we will be using **JWT** as the **Authentication Type**.

- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



Connection Details

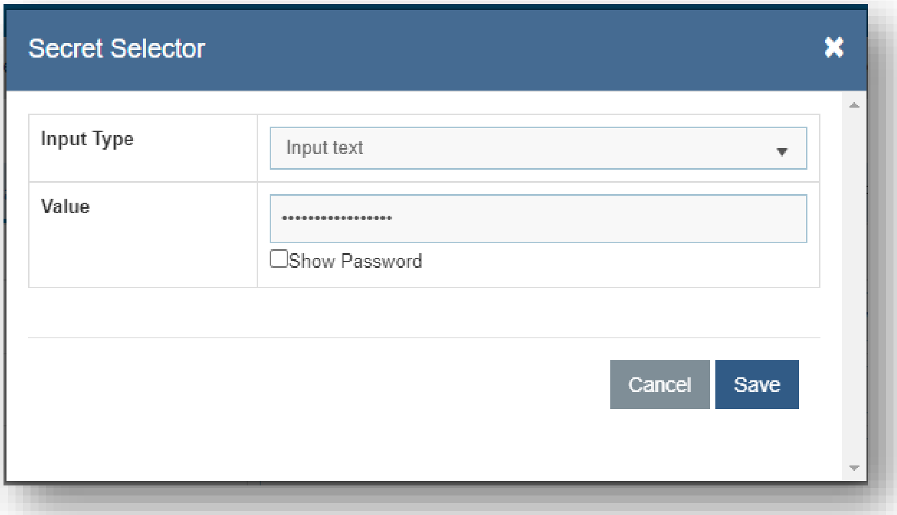
URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	POST
Proxy Required	☐
Test Connection	Test Connection

Request Authentication Parameters

Figure 244 – Create Data Source (Connection Details)

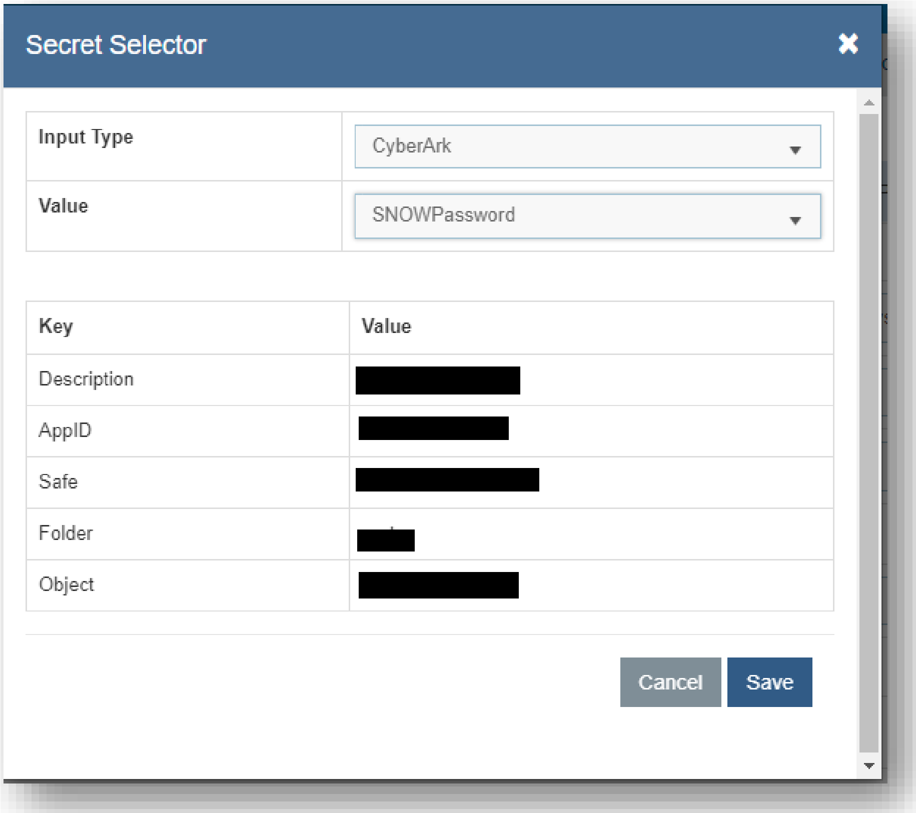
- For **Password**, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key

Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Input Type	Input text
Value	•••••••••••••••• ☐ Show Password

Figure 245 – Password in plaintext



Input Type	CyberArk
Value	SNOWPassword

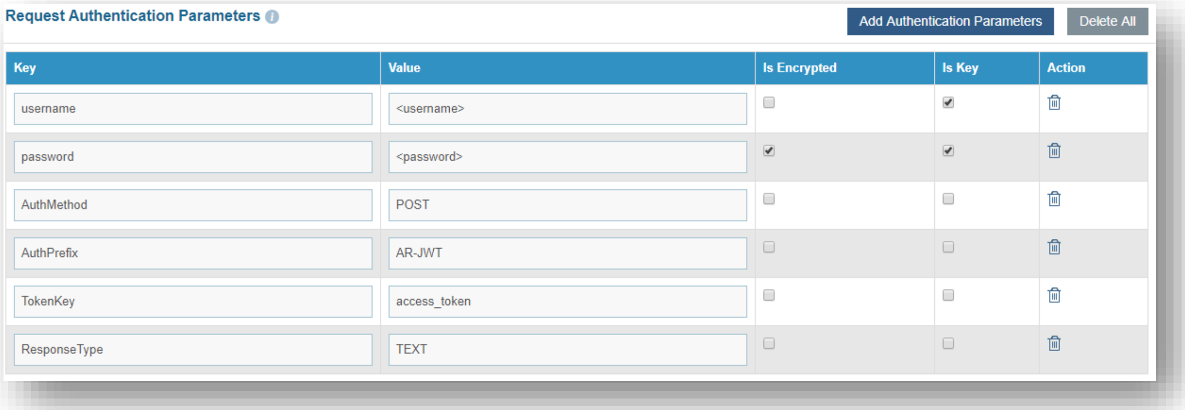
Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 246 – Password from Key Vault (CyberArk)

- **Request Authentication Parameters** – If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.
- Based on the **Authentication Type**, add the parameters mentioned in the below table:

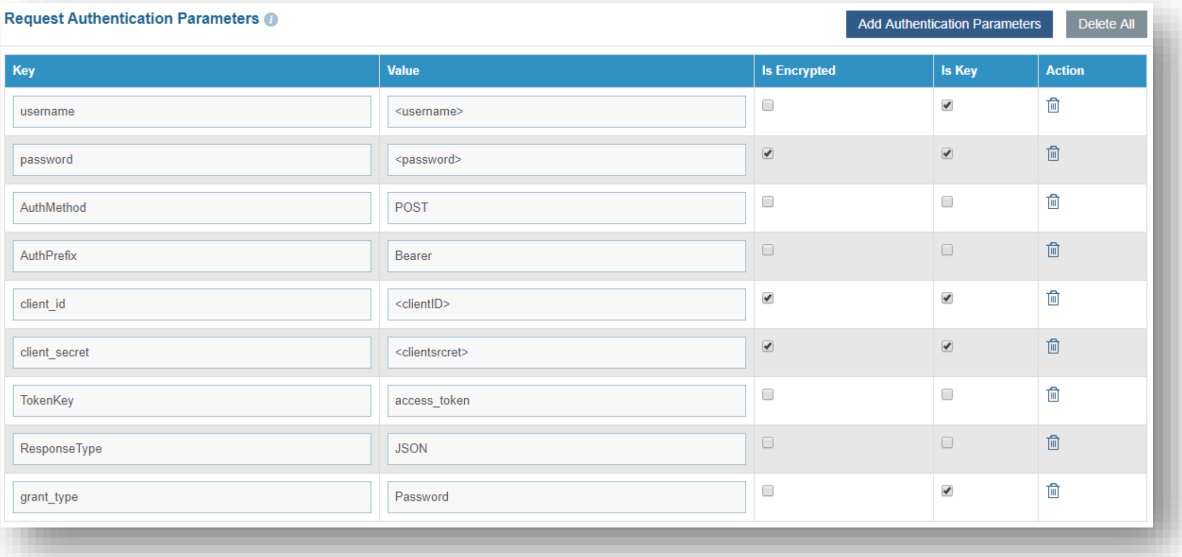
Table 47– Sample Authentication Parameters

Authentication Type	Key	Value	Is Encrypted?	Is Key?
JWT	username	<username>	NO	YES
JWT	password	<password>	YES	YES
JWT	AuthMethod	POST	NO	NO
JWT	AuthPrefix	AR-JWT	NO	NO
JWT	TokenKey	access_token	NO	NO
JWT	ResponseType	TEXT	NO	NO
OAuth2.0	username	<username	NO	YES
OAuth2.0	password	<password>	YES	YES
OAuth2.0	AuthMethod	POST	NO	NO
OAuth2.0	AuthPrefix	Bearer	NO	NO
OAuth2.0	client_id	<clientID>	YES	YES
OAuth2.0	client_secret	<clientsecret>	YES	YES
OAuth2.0	TokenKey	access_token	NO	NO
OAuth2.0	ResponseType	JSON	NO	NO
OAuth2.0	grant_type	Password	NO	YES


Request Authentication Parameters
Add Authentication Parameters
Delete All

Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	AR-JWT	☐	☐	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	TEXT	☐	☐	🗑

Figure 247 – Create Data Source (Request Authentication Parameters for JWT)



Request Authentication Parameters
Add Authentication Parameters
Delete All

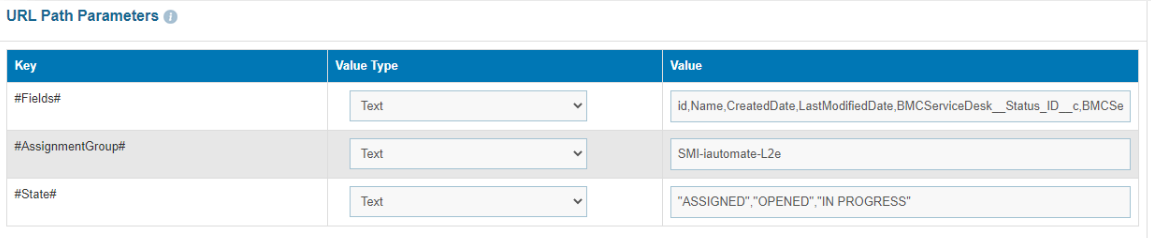
Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	Bearer	☐	☐	🗑
client_id	<clientID>	☑	☑	🗑
client_secret	<clientsrcret>	☑	☑	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	JSON	☐	☐	🗑
grant_type	Password	☐	☑	🗑

Figure 248 – Create Data Source (Request Authentication Parameters for JWT)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #Fields#

ValueType: Text

Value:

id,Name,CreatedDate,LastModifiedDate,BMCServiceDesk__Status_ID__c,
BMCServiceDesk__FKStatus__c,BMCServiceDesk__shortDescription__c,BM
CServiceDesk__incidentDescription__c,BMCServiceDesk__queueName__c,
OwnerID


Note – These columns are mandatory. User can add more columns if
more data is required to be fetched from ITSM tool.


Key: #AssignmentGroup#

ValueType: Text

VALUE: SMI-iautomate-L2e
```

```
Key: #State#

ValueType: Text

VALUE: ''ASSIGNED'',''OPENED'',''IN PROGRESS''
```



URL Path Parameters

Key	Value Type	Value
#Fields#	Text	id,Name,CreatedDate,LastModifiedDate,BMCServiceDesk__Status_ID__c,BMCSe
#AssignmentGroup#	Text	SMI-iautomate-L2e
#State#	Text	"ASSIGNED","OPENED","IN PROGRESS"

Figure 249 – URL Path Parameters (BMC Remedy – Incident Management)

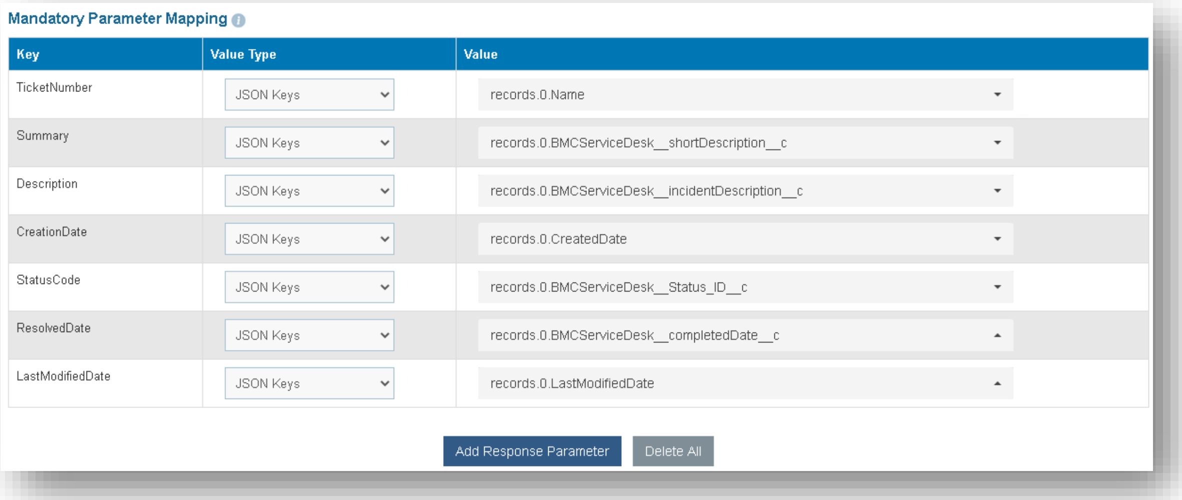
- **Request Header Parameters –** Please enter the request header parameters as required.
- **Response Body –** In this section, please enter the output of URL query for one of the incidents in JSON format. A sample response is mentioned below.

```
Response Body – {
    "totalSize": 1,
    "done": true,
    "records": [
        {
            "attributes": {
                "type": "BMCServiceDesk__Incident__c",
                "url":
"/services/data/v45.0/sobjects/BMCServiceDesk__Incident__c/a1T3H0000008bssUAA"
            },
            "Id": "a1T3H0000008bssUAA",
            "Name": "00238924",
            "CreatedDate": "2020-07-14T14:48:04.000+0000",
            "LastModifiedDate": "2020-07-20T11:28:24.000+0000",
            "BMCServiceDesk__completedDate__c": "2020-07-
20T10:28:14.000+0000",
            "BMCServiceDesk__Status_ID__c": "CLOSED",
            "BMCServiceDesk__FKStatus__c": "a2958000000NzamAAC",
            "BMCServiceDesk__shortDescription__c": "Test Ticket
for BigFix Runbook AI",
            "BMCServiceDesk__incidentDescription__c": "Test Ticket
for BigFix Runbook AI",
            "BMCServiceDesk__queueName__c": "SMI-iautomate-L2e",
            "OwnerId": "00G3H000000W37OUAS"
        }
    ]
}
```

- After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section**.**
- **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below.

Table 48– Sample Mandatory Mapping Parameters

Key	Value Type	Value
TicketNumber	JSON.Keys	records.0.Name
Summary	JSON.Keys	records.0.BMCServiceDesk__shortDescription__c
Description	JSON.Keys	records.0.BMCServiceDesk__incidentDescription__c
CreationDate	JSON.Keys	records.0.CreatedDate
StatusCode	JSON.Keys	records.0.BMCServiceDesk__Status_ID__c
ResolvedDate	JSON.Keys	records.0.BMCServiceDesk__completedDate__c
LastModifiedDate	JSON.Keys	records.0.LastModifiedDate



Mandatory Parameter Mapping

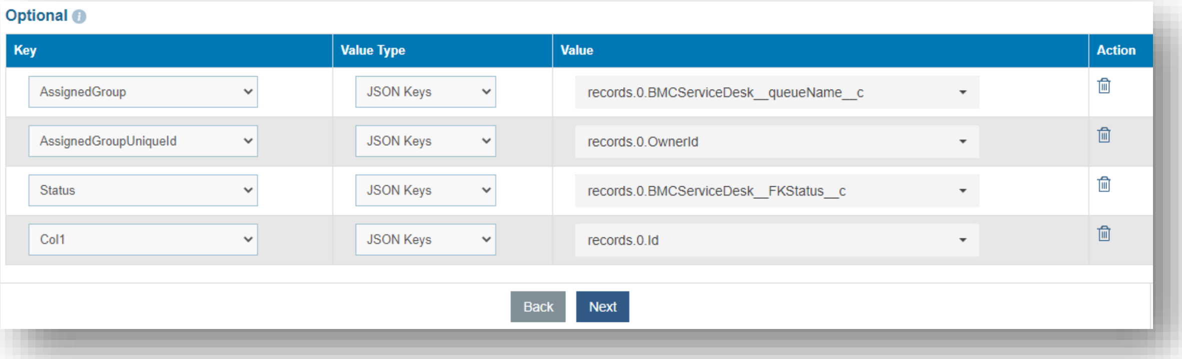
Key	Value Type	Value
TicketNumber	JSON Keys	records.0.Name
Summary	JSON Keys	records.0.BMCServiceDesk__shortDescription__c
Description	JSON Keys	records.0.BMCServiceDesk__incidentDescription__c
CreationDate	JSON Keys	records.0.CreatedDate
StatusCode	JSON Keys	records.0.BMCServiceDesk__Status_ID__c
ResolvedDate	JSON Keys	records.0.BMCServiceDesk__completedDate__c
LastModifiedDate	JSON Keys	records.0.LastModifiedDate

Figure 250 – Mandatory Parameter Mapping

– If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 49– Sample Optional Mapping Parameters

Key	Value Type	Value
AssignedGroup	JSON.Keys	records.0.BMCServiceDesk__queueName__c
Col1	JSON.Keys	records.0.id
AssignedGroupUniqueId	JSON.Keys	records.0.BMCServiceDesk__queueName__c
Status	JSON.Keys	records.0.BMCServiceDesk__FKStatus__c



Optional

Key	Value Type	Value	Action
AssignedGroup	JSON Keys	records.0.BMCServiceDesk__queueName__c	🗑
AssignedGroupUniqueId	JSON Keys	records.0.OwnerId	🗑
Status	JSON Keys	records.0.BMCServiceDesk__FKStatus__c	🗑
Col1	JSON Keys	records.0.Id	🗑

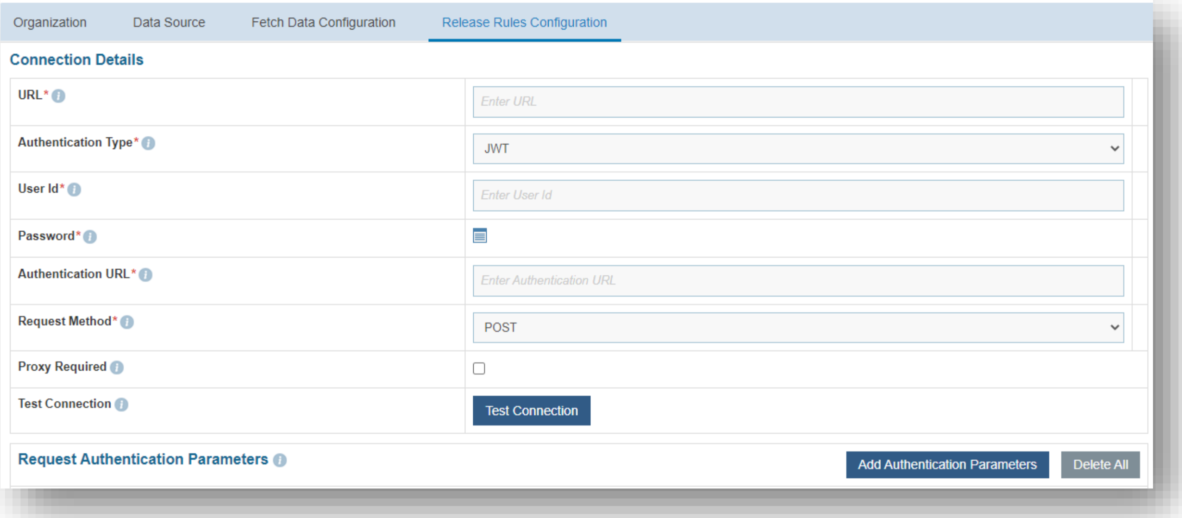
Figure 251 – Optional Parameter Mapping

– Click Next to proceed to Release Rules Configuration.

– On **Release Rules Configuration** tab, type in the details as per the requirement.

– In the **Connection Details** section, enter the following details:

- **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.
- **Sample URL** - http://localhost:8005/Release/#TicketID#
- **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously.
- Request Method – Select Request Method as PUT from the drop-down.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



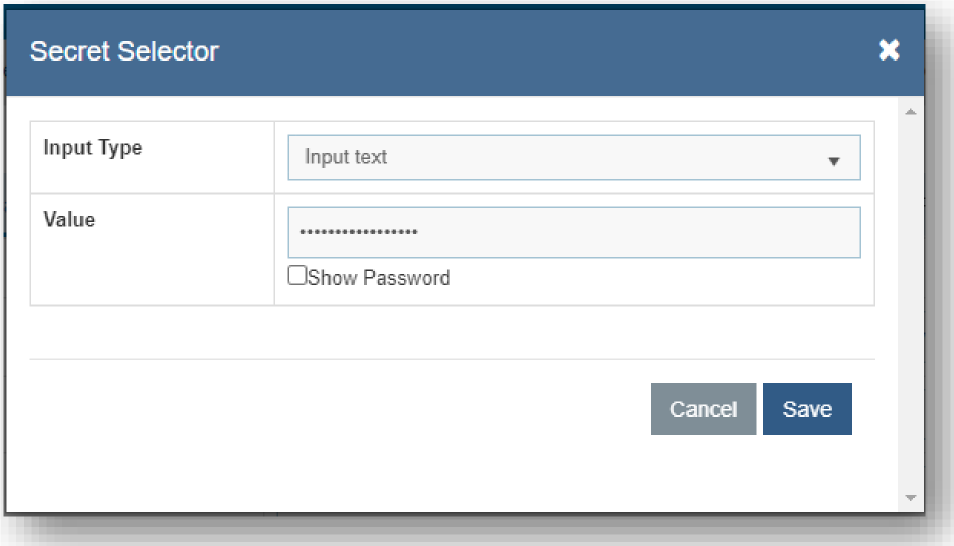
Connection Details

URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	POST
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

Figure 252 – Test Connection

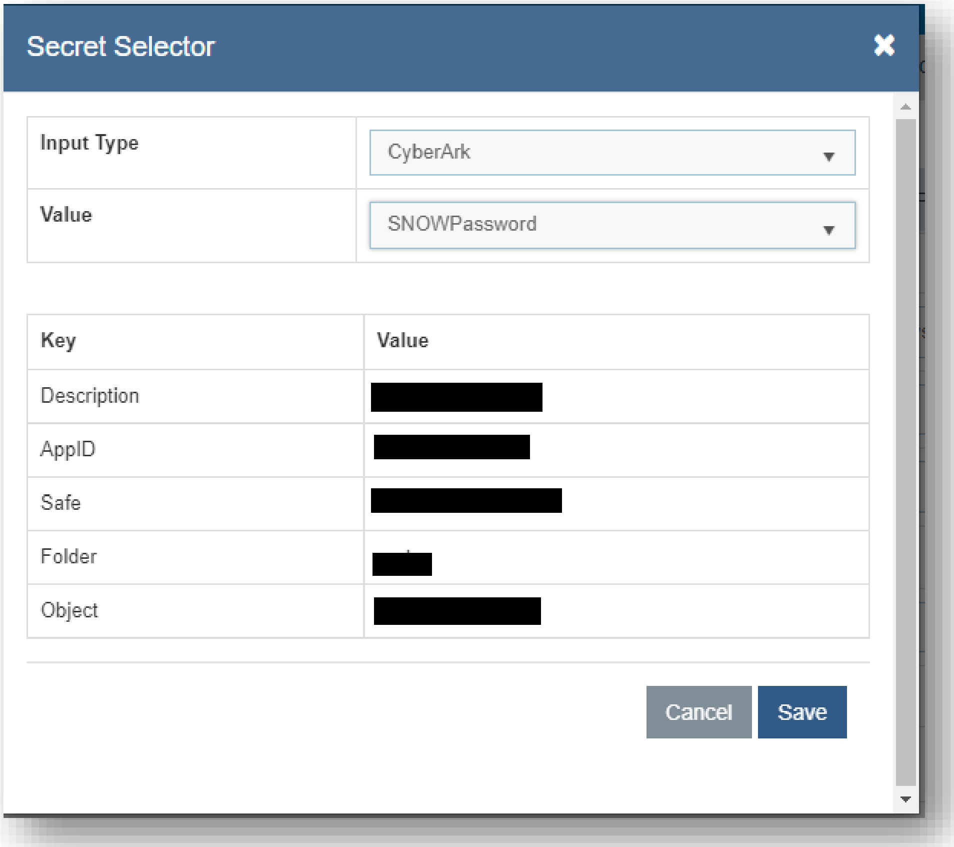
- For **Password**, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Secret Selector

Input Type	Input text
Value	•••••••••••••••• □Show Password

Figure 253 - Password in plaintext



Secret Selector

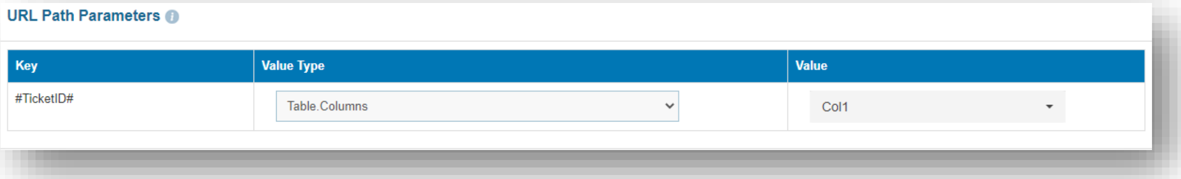
Input Type	CyberArk
Value	SNOWPassword

Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 254 - Password from Key Vault (CyberArk)

- **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs.

```
Key: #TicketId#

ValueType: Table Columns

Value:

Select from dropdown that mapped to sys_id from previous screen
“Col1”
```



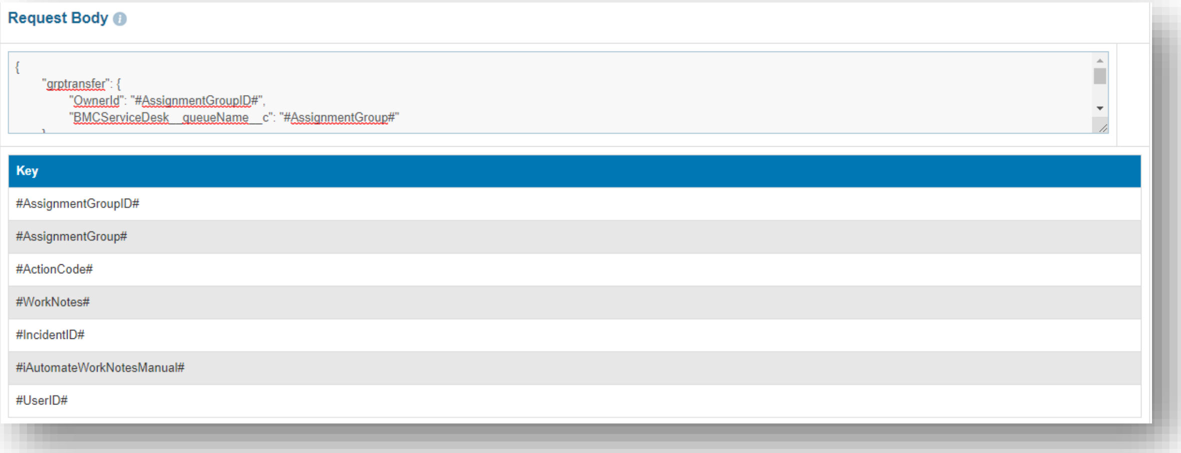
URL Path Parameters

Key	Value Type	Value
#TicketID#	Table.Columns	Col1

Figure 255 – Release Rules Configuration (URL Path Parameters)

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body –** In this section, please enter the request body in JSON format. A sample request is mentioned below:

```
Request Body – {

"grptransfer": {

"OwnerId": "#AssignmentGroupID#",

"BMCServiceDesk__queueName__c": "#AssignmentGroup#"

},

"workorder": {

"BMCServiceDesk__FKAction__c": "#ActionCode#",

"BMCServiceDesk__note__c": "#WorkNotes#",

"BMCServiceDesk__FKIncident__c": "#IncidentID#",

"BMCServiceDesk__description__c": "#BigFix Runbook
AIWorkNotesManual#",

"BMCServiceDesk__FKUser__c": "#UserID#"
}

}
```



Request Body

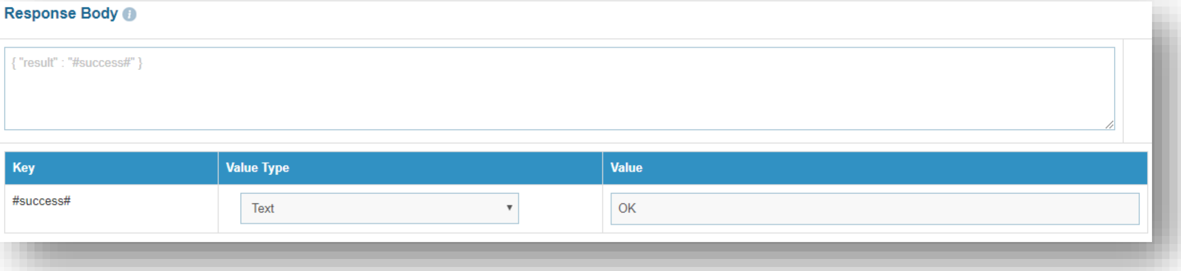
```
{
    "grptransfer": {
        "OwnerId": "#AssignmentGroupID#",
        "BMCServiceDesk__queueName__c": "#AssignmentGroup#"
```

Key
#AssignmentGroupID#
#AssignmentGroup#
#ActionCode#
#WorkNotes#
#IncidentID#
#iAutomateWorkNotesManual#
#UserID#

Figure 256 – Release Rules Configuration (Request Body)

– **Response Body –** In this section, please enter the response body in JSON format. A sample response is mentioned below.

```
Response Body –

{ "result" : "#success#" }
```



Response Body

```
{ "result" : "#success#" }
```

Key	Value Type	Value
#success#	Text	OK

Figure 257 – Release Rules Configuration (Response Body)

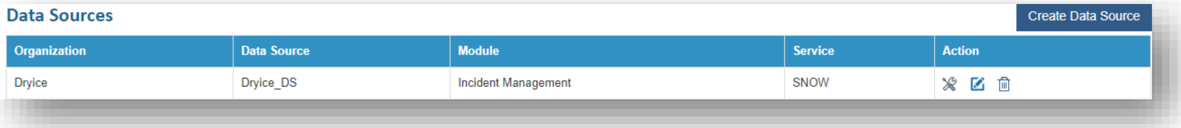
– **Response Key Value** mapping can be done as per the below table.

Table 50– Sample Response Key Value Mapping

#success#	Text	Success

– Click **Submit** to add the data source.

- In order to bring the tickets within BigFix Runbook AI scope, a specific queue needs to be configured in the ITSM tool and same has to be configured in BigFix Runbook AI. This is achieved through **Manage the Entry Criteria**. Please perform the below steps:
  - Go to Action tab and click Manage Data Sources.
  - On the **Data Sources** tab, click ✂ next to the data source user wants to manage. **Manage Entry Criteria** screen appears.

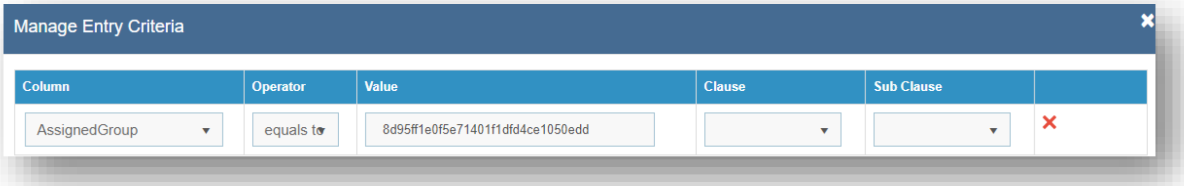


Data Sources

Organization	Data Source	Module	Service	Action
Dryice	Dryice_DS	Incident Management	SNOW	

Figure 258 – Manage Entry Criteria

  - Select 'AssignedGroup' for the **Column** field and 'equals to' for the **Operator** field.
  - Enter the sys_id of the assignment group in Remedyforce in the **Value** field.
  - **Clause** and **Sub-Clause** fields can also be added based on requirement.



Manage Entry Criteria

Column	Operator	Value	Clause	Sub Clause	
AssignedGroup	equals to	8d95ff1e0f5e71401f1dfd4ce1050edd			✕

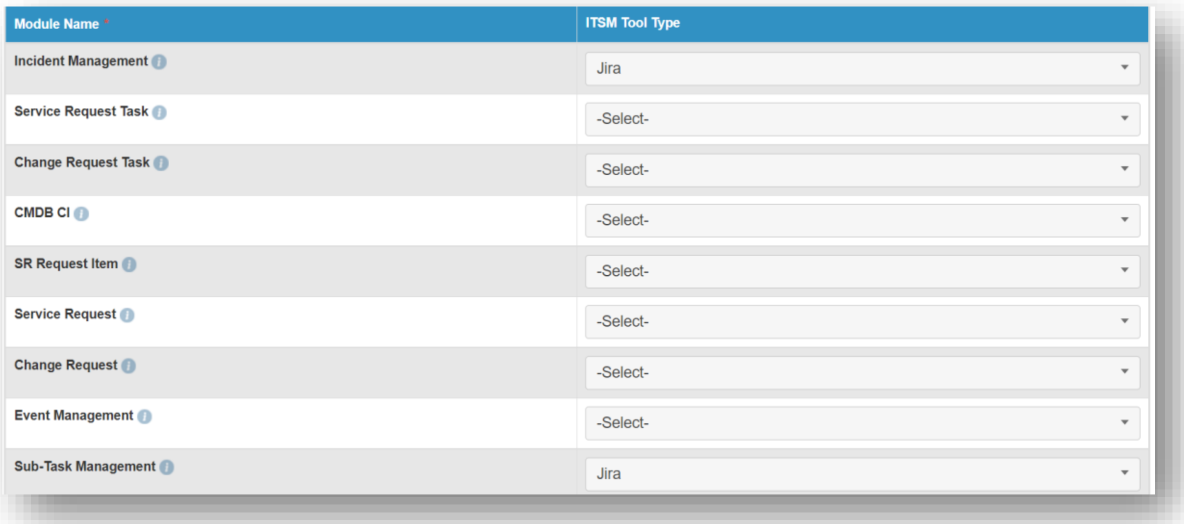
Figure 259 – Manage Entry Criteria (cont.)

  - Click **Save**.

4.6 Integration with JIRA

4.6.1 Incident Management

For Integration of Jira ITSM tool with BigFix Runbook AI, perform the following steps:



Module Name *	ITSM Tool Type
Incident Management	Jira
Service Request Task	-Select-
Change Request Task	-Select-
CMDB CI	-Select-
SR Request Item	-Select-
Service Request	-Select-
Change Request	-Select-
Event Management	-Select-
Sub-Task Management	Jira

Figure 260 – Integration with Jira ITSM Tool

Create Data Source:

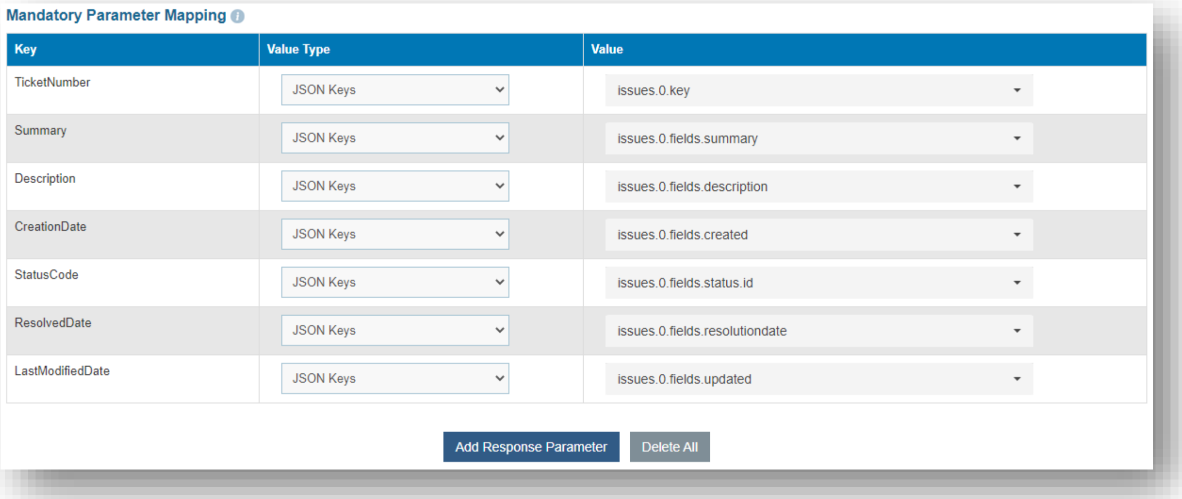
- Fetch Data Configuration:
- **URL**: <URL>/rest/api/2/search?fields=#columns#&jql=issuetype=Incident AND status=Open AND updated >= "#start_date#" AND updated <= "#end_date#" ORDER BY updated DESC
- Authentication Type: Basic
- Request Method: GET
- URL Path Parameters:

Key	Value Type	Value
#columns#	Text	key,description,summary,created,updated, status,assignee,resolutiondate
#start_date#	SQL UDF	@@GetFromDateTimeUsingIncidentModifiedDate
#end_date#	SQL UDF	@@GetToolCurrentDateTime

- Response Body:

```
{
"expand": "schema,names",
"startAt": 0,
"maxResults": 50,
"total": 3,
"issues": [{
"expand": "operations,versionedRepresentations,editmeta,changelog,renderedFields",
"id": "10102",
"self": "http://10.1.152.20:8080/rest/api/2/issue/10102",
"key": "IT-48",
"fields": {
"summary": "REST ye merry gentlemen. Rest in peace",
                         "resolutiondate": "2021-05-05T13:17:10.000+0530",
"created": "2021-05-05T13:17:10.000+0530",
"description": "Creating of an issue using project keys and issue type names using the REST API",
"assignee": null,
"updated": "2021-05-05T13:17:10.000+0530",
"status": {
"self": "http://10.1.152.20:8080/rest/api/2/status/1",
"description": "The issue is open and ready for the assignee to start work on it.",
"iconUrl": "http://10.1.152.20:8080/images/icons/statuses/open.png",
"name": "Open",
"id": "1",
"statusCategory": {
"self": "http://10.1.152.20:8080/rest/api/2/statuscategory/2",
"id": 2,
"key": "new",
"colorName": "blue-gray",
"name": "To Do"
}
}

}
}]
}
```

– Mandatory Parameter Mapping:

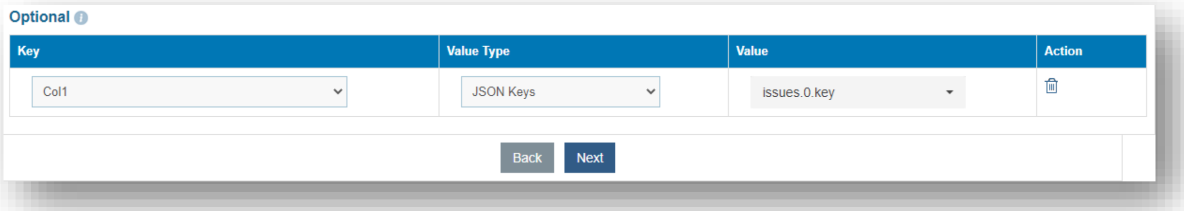


Mandatory Parameter Mapping

Key	Value Type	Value
TicketNumber	JSON Keys	issues.0.key
Summary	JSON Keys	issues.0.fields.summary
Description	JSON Keys	issues.0.fields.description
CreationDate	JSON Keys	issues.0.fields.created
StatusCode	JSON Keys	issues.0.fields.status.id
ResolvedDate	JSON Keys	issues.0.fields.resolutiondate
LastModifiedDate	JSON Keys	issues.0.fields.updated

Figure 261 – Mandatory Parameter Mapping

– Optional:



Optional

Key	Value Type	Value	Action
Col1	JSON Keys	issues.0.key	🗑

Figure 262 – Optional

Release Rule Configuration:

For release, since Jira has 3 different APIs to change the assignee, to add a comment and to add worklog. So, we are using BigFix Runbook AI's Custom Script API to update all 3 operations with one single API.

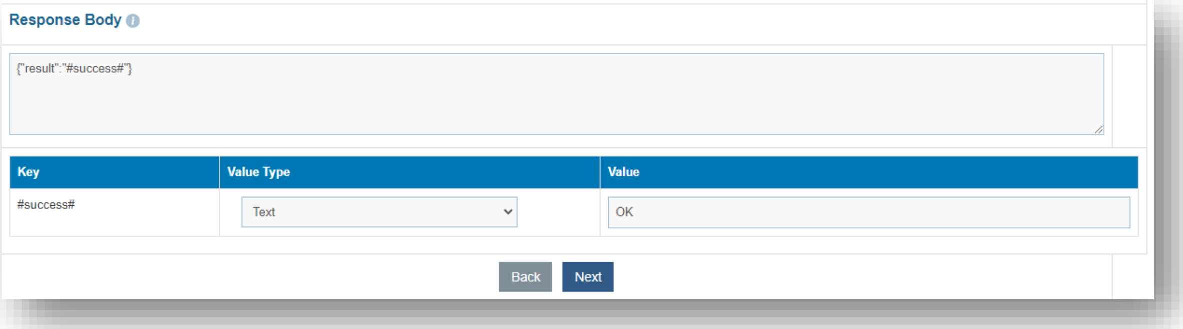
To create Custom API go to Manage Custom Script Section.

- URL: http://10.1.152.20:8080/rest/api/2/issue/#key#/assignee
- Authentication Type: Basic
- **UserId**: ApiUser@hcl.com
- **Password**: user_password
- Request Method: POST
- Request Body:

```
{
    "key": "#ticketId#",
    "URL": "http://10.1.152.20:8080/rest/api/2/issue/",
    "assignee_name": "#assignee_name#",
    "release_comment":"Ticket_released_from_BigFix Runbook AI"
}
```

- Response Body:

```
{"result":"#success#"}
```



Response Body

{"result":"#success#"}

Key	Value Type	Value
#success#	Text	OK

Figure 263 – Response Body

Close Rules Configuration:

- URL: http://10.1.152.20:8080/rest/api/2/issue/#key#/transitions
- Authentication Type: Basic
- Request Method: POST
- URL Path Parameters:

Key	Value Type	Value
#key#	Table.Columns	Col1

- Request Body:

```
{
    "update": {
        "comment": [
            {
                "add": {
                    "body": "#worknote#"
                }
            }
        ]
    },
    "transition": {
        "id": "#statuscode#"
    }
}
```

- Response Body:

```
{ "result" : "ok" }
```

InProgress Rules Configuration:

- **URL**: http://10.1.152.20:8080/rest/api/2/issue/#sysid#/transitions
- Authentication Type: Basic
- Request Method: POST
- URL Path Parameters:

Key	Value Type	Value
#key#	Table.Columns	Col1

- Request Body:

```
{
    "update": {
        "comment": [
            {
                "add": {
                    "body": "#worknote#"
                }
            }
        ]
    },
    "transition": {
        "id": "#statuscode#"
    }
}
```

- Response Body:

```
{ "result" : "ok" }
```

JsResponseConverter: After successful creation of data source,

- Go to CollectIncident job under menu Environment → Manage Jobs.
- Click on ⚙ icon. A popup will be opened.
- Go to parameter tab and search for '**JsResponseConverter**' in the end. Replace its value with below string:

```
if(json.issues){for(var
result=[],i=0;i<json.issues.length;i++)result.push(json.issues[i])
;customJobject.dataCollectorNode.data.issues=result}
```

Manage Rules:

For each of the release, close, and in-progress rules are defined as follows:

- **Release Rules**

Parameter	Value Type	Value
#assignee_name#	Text	Assignee_user
#ticketId#	Table.Columns	Col1

- **Close Rules:**

Parameter	Value Type	Value
#worknote#	Text	Ticket closed from BigFix Runbook AI
#ticketId#	Text	91

- **In Progress Rules:**

Parameter	Value Type	Value
#worknote#	Text	Ticket marked to in progress
#ticketId#	Text	31

Manage Custom Script:

To use multiple Jira APIs that are being used while releasing an incident, you need a python script that contains the calling of all required APIs.

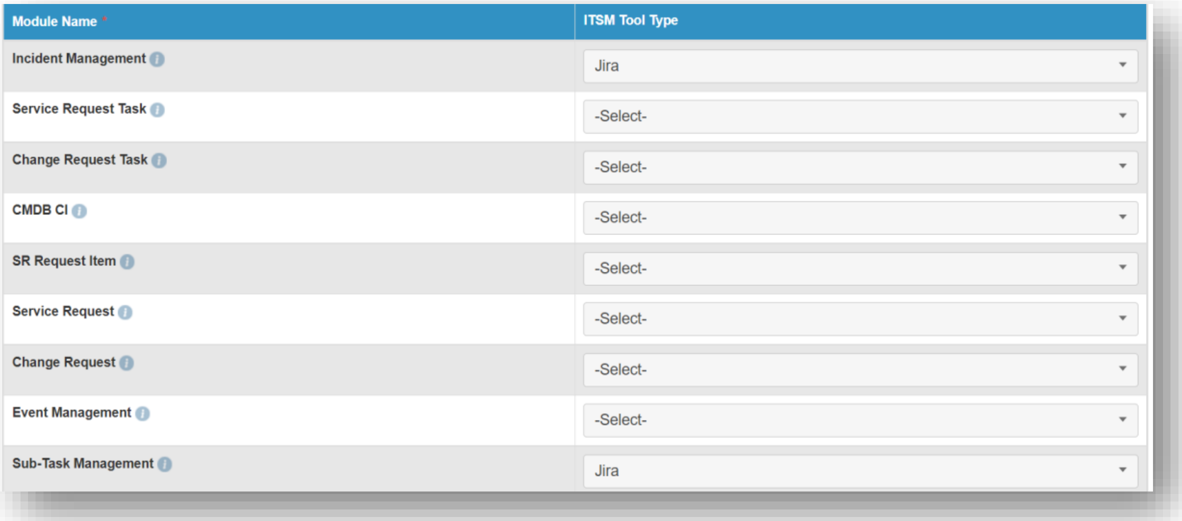
- For that go to page Environment → Manage Custom Script → Create Script.
- Select **Input Mode** as Manual, **Script Language** as Python, enter the name of script in the **Script Name** textbox.
- Enter **Tags** (if needed) and paste below content in the **Script Text** textbox.

```
import json

import requests

import sys


try:

 ##url = "http://10.1.152.20:8080/rest/api/2/issue/IT-90/assignee"
//update assignee

## Mandory


 resp = json.loads(sys.argv[2])

 url = resp["URL"] + resp["key"] + "/assignee"


 payload = json.dumps({

    "name":  resp["assignee_name"]

  })

 headers = {

    'Authorization': 'Basic QXNoaXNoTWlzaHJhOkluZGlhQDEyMw==',

    'Content-Type': 'application/json'

  }
 response = requests.request("PUT", url, headers=headers,
data=payload)

 print(response.text)

 import requests
 import json
 import sys

 ##url = "http://10.1.152.20:8080/rest/api/2/issue/IT-90"   //add
comment
 resp = json.loads(sys.argv[2])
 url = resp["URL"] + resp["key"]
 payload = json.dumps({
   "update": {
     "comment": [
       {
         "add": {
           "body": resp["release_comment"]
         }
       }
     ]
   }
 })

 response = requests.request("PUT", url, headers=headers,
data=payload)
```

```
 print(response.text)
 import requests
 import json
 import sys

 ##url = "http://10.1.152.20:8080/rest/api/2/issue/IT-90/worklog"
//add worklog
## Mandory
 resp = json.loads(sys.argv[2])
 url = resp["URL"] + resp["key"]+"/worklog"
 payload = json.dumps({
    "comment": resp["release_comment"],
    "timeSpentSeconds": 6000
 })
 response = requests.request("POST", url, headers=headers,
data=payload)
 print(response.text)
except Exception as e:
  message = {"Error": "Error in running Script, Error=>" + str(e)}
  message = json.dumps(message)
  code = 400
  print(str(message))
```

4.6.2 Sub-Task Management

For Integration of Jira ITSM Sub-Task with BigFix Runbook AI tool, perform the following steps:



Module Name *	ITSM Tool Type
Incident Management	Jira
Service Request Task	-Select-
Change Request Task	-Select-
CMDB CI	-Select-
SR Request Item	-Select-
Service Request	-Select-
Change Request	-Select-
Event Management	-Select-
Sub-Task Management	Jira

Figure 264 - Integration of Jira IITSM Sub-Task

Create Data Source:

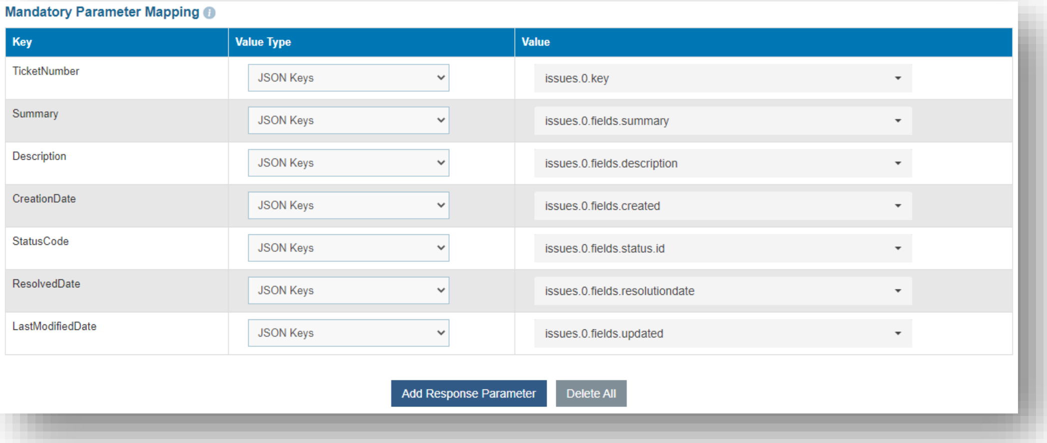
- Fetch Data Configuration:
- **Sample URL:** http://<JIRA_URL>/rest/api/2/search?fields=#columns#&jql=issuetype="Sub-task" AND status=Open AND updated >= "#start_date#" AND updated <= "#end_date#" ORDER BY updated
- Authentication Type: Basic
- Request Method: GET
- URL Path Parameters:

Key	Value Type	Value
#columns#	Text	key,description,summary,created,updated,status,assignee,resolutiondate, issuetype
#start_date#	SQL UDF	@@GetFromDateTimeUsingTaskModifiedDate_Jira
#end_date#	SQL UDF	@@GetToolCurrentDateTime_Jira

- Response Body:

```
{
"expand": "schema,names",
"startAt": 0,
"maxResults": 50,

"total": 3,

"issues": [{

"expand":
"operations,versionedRepresentations,editmeta,changelog,renderedFi
elds",

"id": "10102",

"self": "http://10.1.152.20:8080/rest/api/2/issue/10102",

"key": "IT-48",

"fields": {

"summary": "REST ye merry gentlemen. Rest in peace",

"resolutiondate":"2021-05-05T13:17:10.000+0530",

"created": "2021-05-05T13:17:10.000+0530",

"description": "Creating of an issue using project keys and issue
type names using the REST API",

"assignee": null,

"updated": "2021-05-05T13:17:10.000+0530",

"status": {

"self": "http://10.1.152.20:8080/rest/api/2/status/1",

"description": "The issue is open and ready for the assignee to
start work on it.",

"iconUrl":
"http://10.1.152.20:8080/images/icons/statuses/open.png",

"name": "Open",

"id": "1",

"statusCategory": {

"self": "http://10.1.152.20:8080/rest/api/2/statuscategory/2",

"id": 2,
"key": "new",

"colorName": "blue-gray",

"name": "To Do"

}

}




}

}]

}
```

– Mandatory Parameter Mapping:

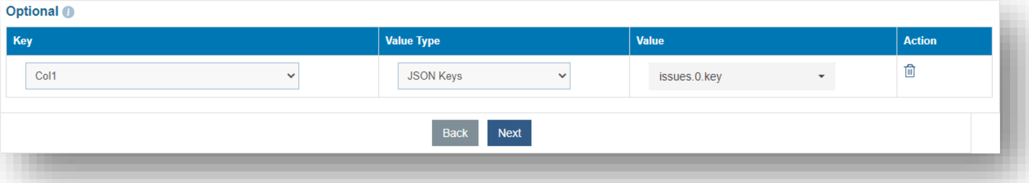


Mandatory Parameter Mapping

Key	Value Type	Value
TicketNumber	JSON Keys	issues.0.key
Summary	JSON Keys	issues.0.fields.summary
Description	JSON Keys	issues.0.fields.description
CreationDate	JSON Keys	issues.0.fields.created
StatusCode	JSON Keys	issues.0.fields.status.id
ResolvedDate	JSON Keys	issues.0.fields.resolutiondate
LastModifiedDate	JSON Keys	issues.0.fields.updated

Figure 265 – Mandatory Parameter Mapping

– Optional:



Optional

Key	Value Type	Value	Action
Col1	JSON Keys	issues.0.key	🗑

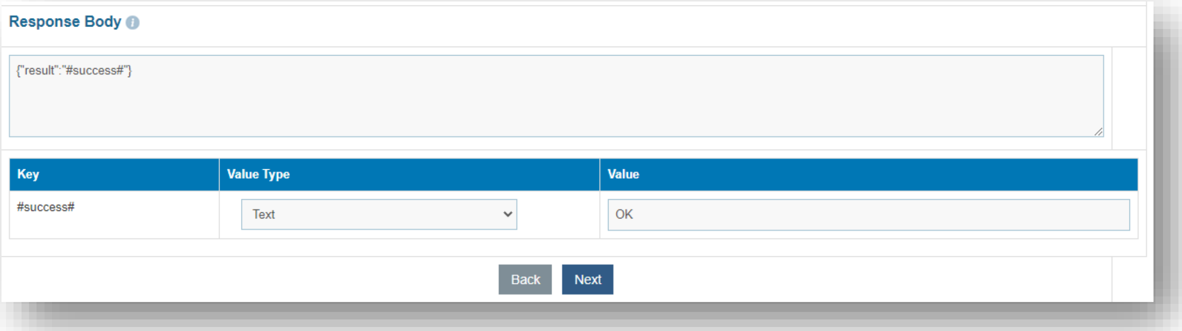
Figure 266 – Optional

Release Rule Configuration:

For release, since Jira has 3 different APIs to change the assignee, to add a comment and to add worklog. So, we are using BigFix Runbook AI's Custom Script API to update all 3 operations with a single API.

- URL: http://10.1.152.20:8080/rest/api/2/issue/#key#/assignee
- Authentication Type: Basic
- **UserId**: <ApiUser@hcl.com>
- **Password**: <user_password>
- Request Method: POST
- Request Body:

```
{
    "key": "#ticketId#",
    "URL": "http://10.1.152.20:8080/rest/api/2/issue/",
    "assignee_name": "#assignee_name#",
    "release_comment":"Ticket released from BigFix Runbook AI"
          }
      Response Body:
{"result":"#success#"}
```



Response Body

{"result":"#success#"}

Key	Value Type	Value
#success#	Text	OK

Figure 267 – Response Body

Close Rules Configuration:

- URL: http://10.1.152.20:8080/rest/api/2/issue/#key#/transitions

- Authentication Type: Basic
- Request Method: POST
- URL Path Parameters:

Key	Value Type	Value
#key#	Table.Columns	Col1

- Request Body:

```
{
    "update": {
        "comment": [
            {
                "add": {
                    "body": "#worknote#"
                }
            }
        ]
    },
    "transition": {
        "id": "#statuscode#"
    }
}
Response Body: { "result" : "ok" }
```

InProgress Rules Configuration:

- **URL**: *http://10.1.152.20:8080/rest/api/2/issue/#sysid#/transitions*
- Authentication Type: Basic
- Request Method: POST
- URL Path Parameters:

Key	Value Type	Value
#key#	Table.Columns	Col1

- Request Body:

```
{
    "update": {
        "comment": [
            {
                "add": {
                    "body": "#worknote#"
                }
            }
        ]
    },
    "transition": {
        "id": "#statuscode#"
    }
}
```

- Response Body:

```
{ "result" : "ok" }
```

JsResponseConverter: After successful creation of data source,

- Go to CollectIncident job under menu **Environment→ Manage Jobs**.
- Click on ⚙ icon. A popup will be opened.
- Go to parameter tab and search for 'JsResponseConverter' in the end.
- Replace its value with below string:

```
if(json.issues){for(var
result=[],i=0;i<json.issues.length;i++)result.push(json.issues[i])
;customJobject.dataCollectorNode.data.issues=result}
```

Manage Rules

For each of the release, close and inprogress, rules will be defined as follows:

- **Release Rules:**

Parameter	Value Type	Value
#assignee_name#	Text	<Assignee_user>
#ticketId#	Table.Columns	Col1

- **Close Rules:**

Parameter	Value Type	Value
#worknote#	Text	Ticket resolved from BigFix Runbook AI
#statuscode#	Text	61

- **In Progress Rules:**

Parameter	Value Type	Value
#worknote#	Text	Ticket marked to in progress
#statuscode#	Text	11

Manage Custom Script:

To use multiple Jira APIs that are being used while releasing an incident, we need a python script that contains the calling of all required APIs.

- For that go to page Environment→Manage Custom Script →Create Script.
- Select Manual as **Input Mode**, Python as **Script Language**, enter the name of script in the **Script Name** textbox.
- Enter tags if needed and paste below content as it is in **Script Text** textbox.

```
import json
import requests
import sys

try:
 ##url = "http://10.1.152.20:8080/rest/api/2/issue/IT-90/assignee"
//update assignee
## Mandory


 resp = json.loads(sys.argv[2])
 url = resp["URL"] + resp["key"] + "/assignee"


 payload = json.dumps({
    "name":  resp["assignee_name"]
  })
 headers = {
    'Authorization': 'Basic QXNoaXNoTWlzaHJhOkluZGlhQDEyMw==',
    'Content-Type': 'application/json'
  }


 response = requests.request("PUT", url, headers=headers,
data=payload)


 print(response.text)


 import requests
 import json
 import sys


 ##url = "http://10.1.152.20:8080/rest/api/2/issue/IT-90"   //add
comment
 resp = json.loads(sys.argv[2])
 url = resp["URL"] + resp["key"]

 payload = json.dumps({

   "update": {

     "comment": [

       {

         "add": {

           "body": resp["release_comment"]

         }

       }

     ]

   }

 })


 response = requests.request("PUT", url, headers=headers,
data=payload)


 print(response.text)

 import requests

 import json

 import sys


 ##url = "http://10.1.152.20:8080/rest/api/2/issue/IT-90/worklog"
//add worklog

## Mandory

 resp = json.loads(sys.argv[2])

 url = resp["URL"] + resp["key"]+"/worklog"

 payload = json.dumps({
    "comment": resp["release_comment"],

    "timeSpentSeconds": 6000

 })


 response = requests.request("POST", url, headers=headers,
data=payload)


 print(response.text)


except Exception as e:

  message = {"Error": "Error in running Script, Error=>" + str(e)}

  message = json.dumps(message)

  code = 400

  print(str(message))
```

4.7 Integration with ServiceXchange

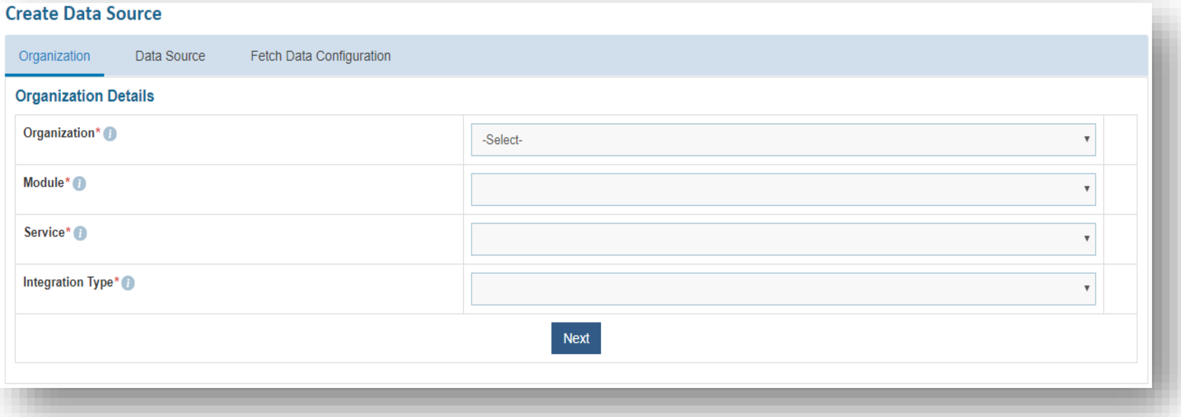
4.7.1 Incident Management

In order to create data source for Incident Management, perform the following steps.

On the main menu bar, click **Action → Manage Data Sources** .

– The **Create Data Source** page appears with the following tabs:

  - Organization
  - Data Source
  - Fetch Data Configuration
  - Release Rules Configuration
  - Close Rules Configuration
  - InProgress Rules Configuration



Create Data Source

Organization Details

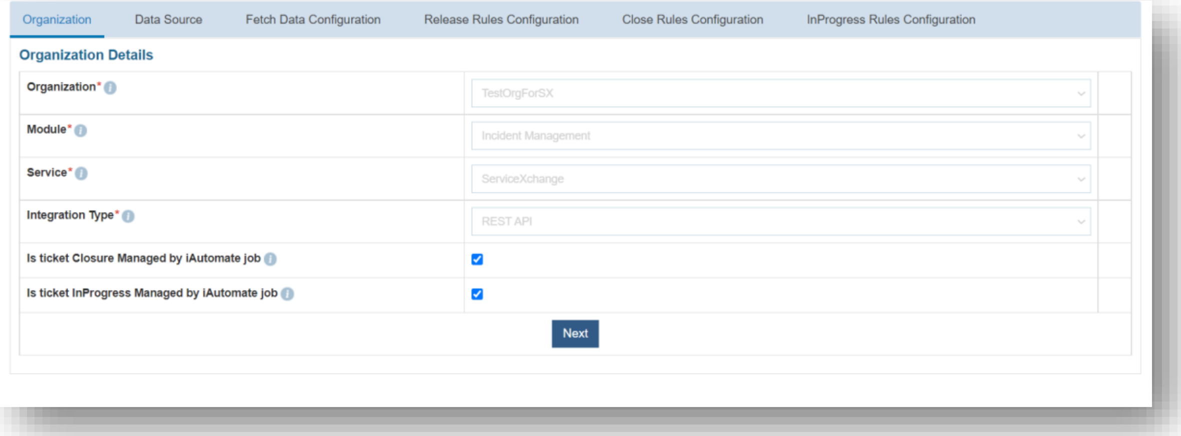
Organization*	-Select-
Module*	
Service*	
Integration Type*	

Figure 268 - Create Data Source

Release Rules Configuration is only applicable for the following **Module** types- **Incident Management, Change Request Task and Service Request Task.** This tab will not be activated for other module types.

– On the **Organization** tab,

- Select the Organization Name from the dropdown.
- Select the Module as Incident Management, since we are configuring this data source for pulling the incident tickets.
- Select the Service as SX Tool as we are configuring the data source for Cherwell
- Select the Integration Type as REST, since we will be integrating through REST APIs.
- Click Next.



Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

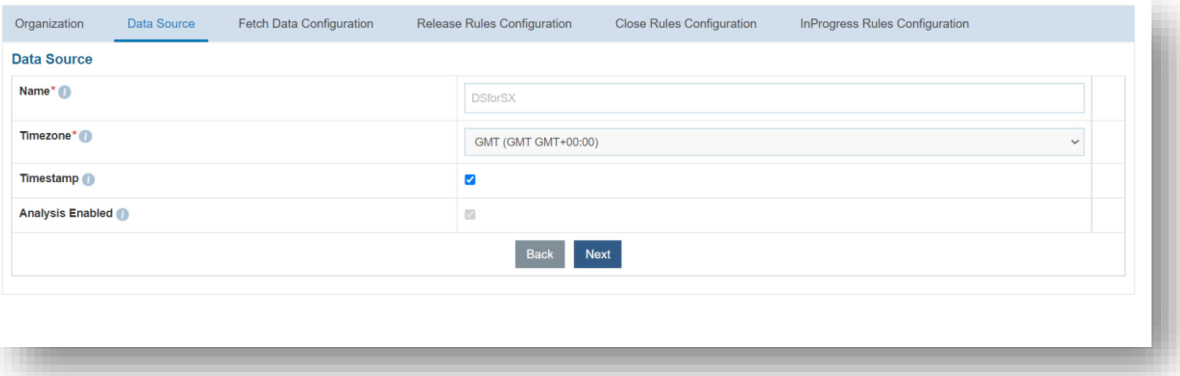
Organization Details

Organization*	TestOrgForSX
Module*	Incident Management
Service*	ServiceXchange
Integration Type*	REST API
Is ticket Closure Managed by iAutomate job	☑
Is ticket InProgress Managed by iAutomate job	☑

Figure 269 – Create Data Source (Contd.)

– On the **Data Source** tab,

- Type the new data source in the **Name** field.
- Select the **Timezone** to specify the time zone of the selected data source.
- Select **Timestamp** to view the present data with date and time.
- Click **Next**.



Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

Data Source

Name*	DSforSX
Timezone*	GMT (GMT GMT+00:00)
Timestamp	☑
Analysis Enabled	☑

Figure 270 – Create Data Source (Contd.)

– On the **Fetch Data Configuration** tab, type in the details as per the environment.

– In the **Connection Details** section, enter the following details:

- **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.
- Sample URL – http://<iAutomate_API_URL>/iAutomateAPI/Request/ GetIncidentTicketData/<Org_ID>? ModuleId=1&start_date>=#Start_Date#&end_date<=#End_Date#&

Here, < iAutomate_API_URL > is the API URL of BigFix Runbook AI where Push APIs are present and <Org_ID> is the OrgID for the organization for which you are creating the data source. It is available in Organization Master in Database.

- **Authentication Type** – Select one of the Authentication Types from Basic / Windows, JWT, OAuth 2.0

The user details that are entered here should be an API User

Selection of **Basic / Windows** requires you to enter -

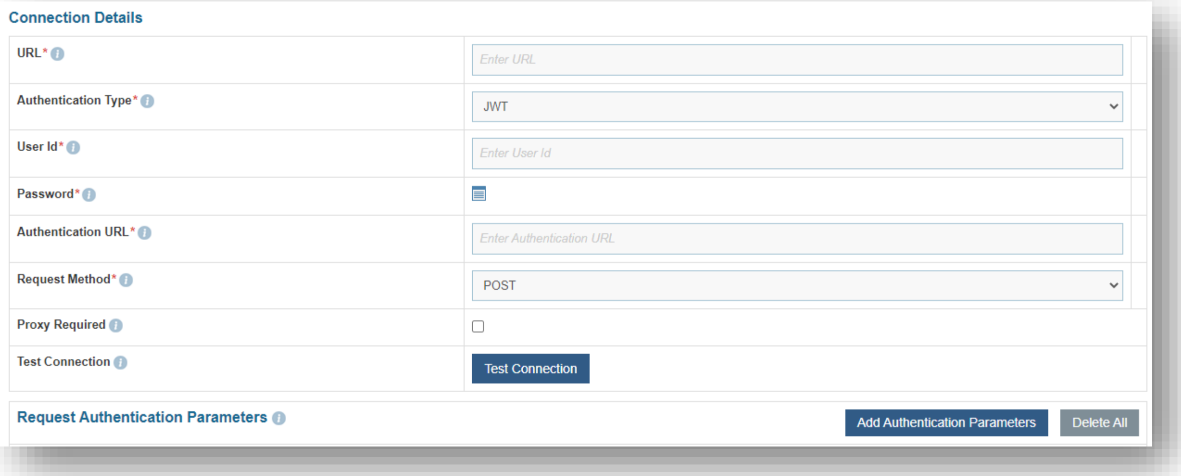
- User Id
- Password.

Selection of **JWT / OAuth 2.0** requires you to enter -

- User Id

- Password
- Authentication URL

- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



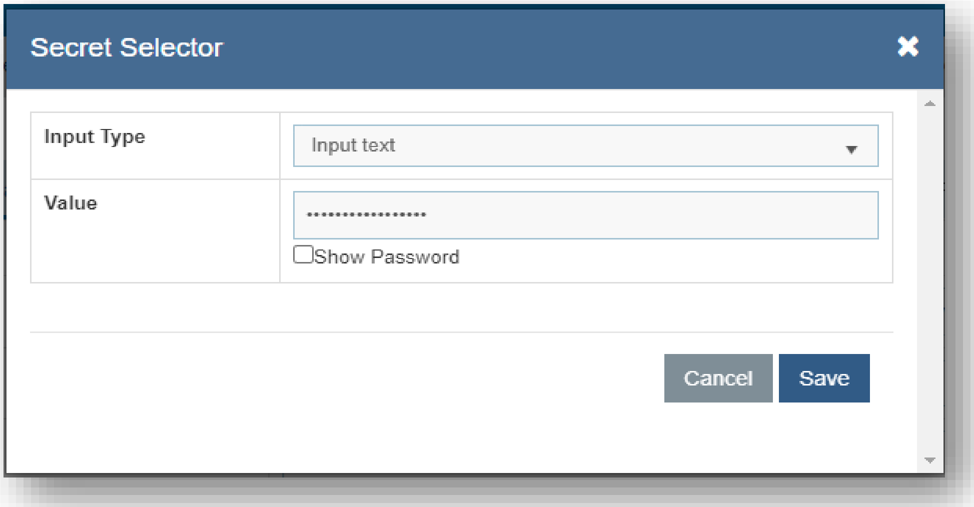
Connection Details

URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	▤
Authentication URL*	Enter Authentication URL
Request Method*	POST
Proxy Required	☐
Test Connection	Test Connection

Request Authentication Parameters

Figure 271 – Create Data Source (Connection Details)

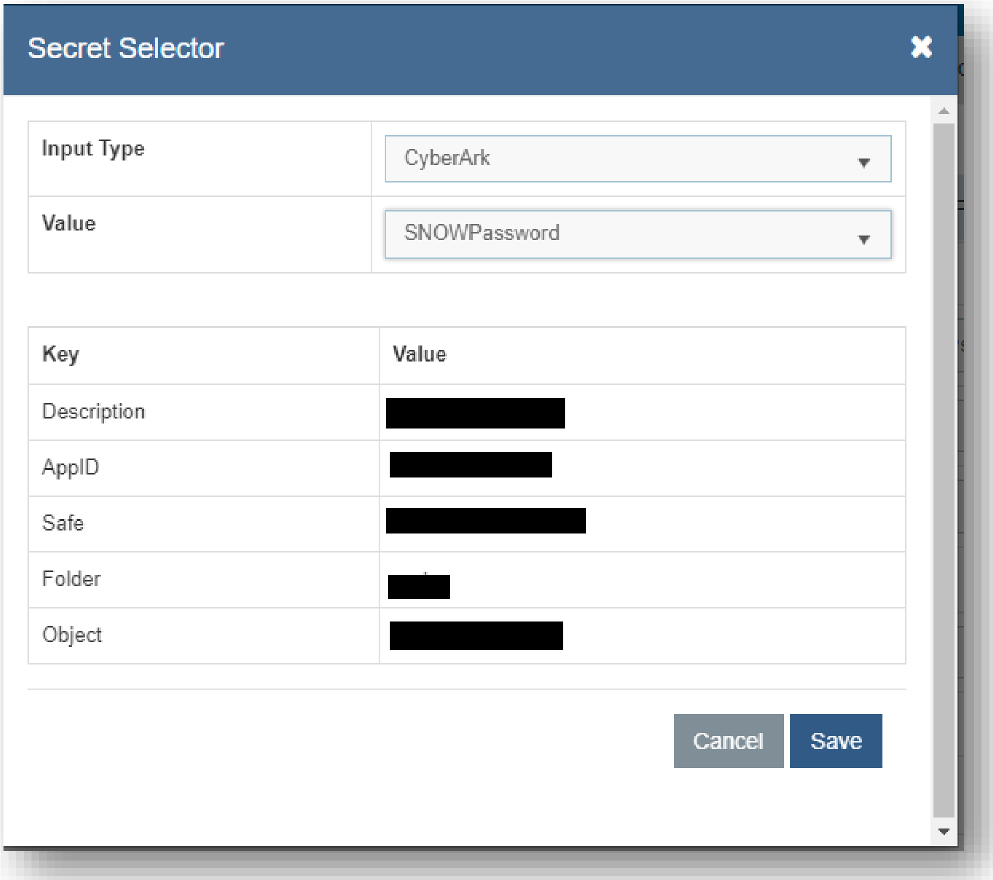
- **Password** – For password, click on icon next to it. If the password is available in plaintext then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Secret Selector

Input Type	Input text
Value	•••••••••••••••••• □ Show Password

Figure 272 – Password in plaintext



Secret Selector

Input Type	CyberArk
Value	SNOWPassword

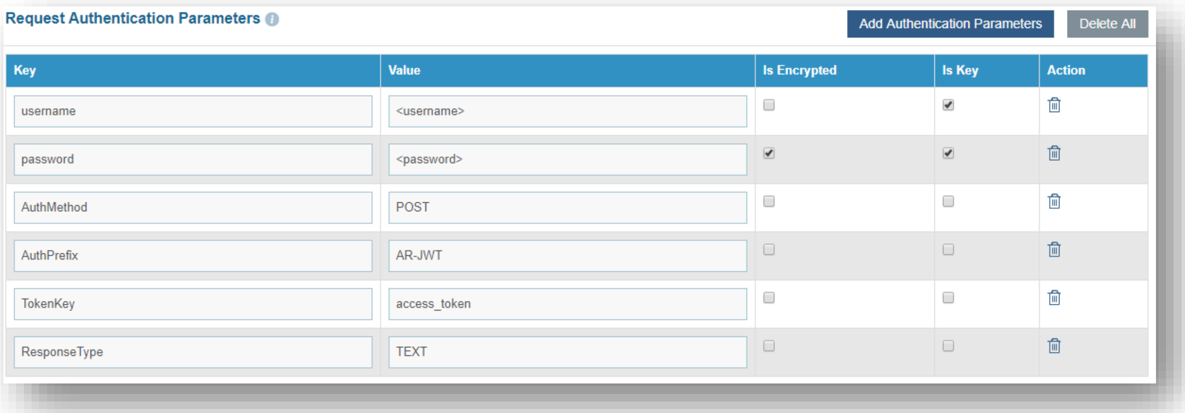
Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 273 – Password from Key Vault (CyerArk)

- **Request Authentication Parameters** – If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab. Based on the **Authentication Type**, add the parameters mentioned in the below table.

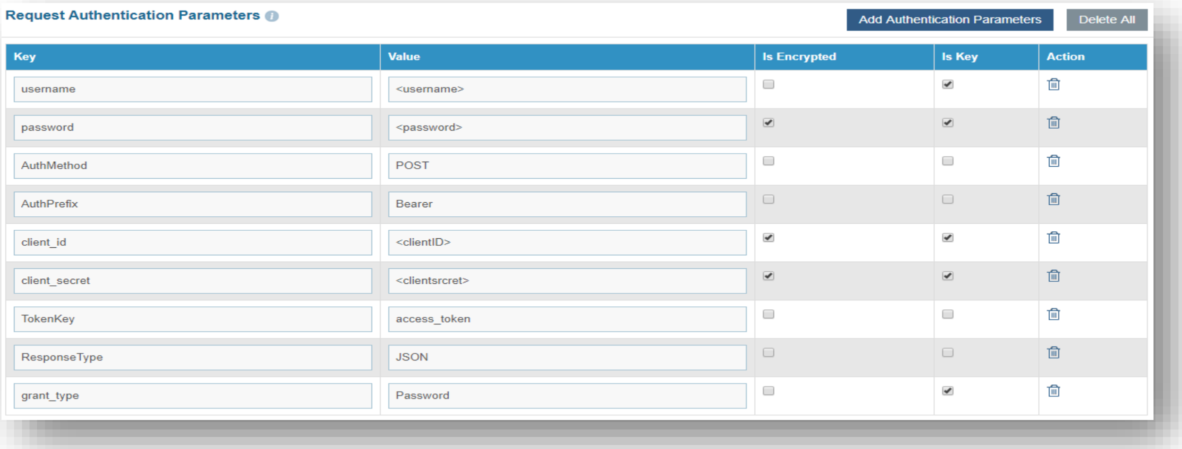
Table 51– Sample Authentication Parameters

Authentication Type	Key	Value	Is Encrypted?	Is Key?
JWT	username	<username>	NO	YES
JWT	password	<password>	YES	YES
JWT	AuthMethod	POST	NO	NO
JWT	AuthPrefix	AR-JWT	NO	NO
JWT	TokenKey	access_token	NO	NO
JWT	ResponseType	TEXT	NO	NO
OAuth2.0	username	<username	NO	YES
OAuth2.0	password	<password>	YES	YES
OAuth2.0	AuthMethod	POST	NO	NO
OAuth2.0	AuthPrefix	Bearer	NO	NO
OAuth2.0	client_id	<clientID>	YES	YES
OAuth2.0	client_secret	<clientsecret>	YES	YES
OAuth2.0	TokenKey	access_token	NO	NO
OAuth2.0	ResponseType	JSON	NO	NO
OAuth2.0	grant_type	Password	NO	YES


Request Authentication Parameters
Add Authentication Parameters
Delete All

Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	AR-JWT	☐	☐	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	TEXT	☐	☐	🗑

Figure 274 – Create Data Source (Request Authentication Parameters for JWT)



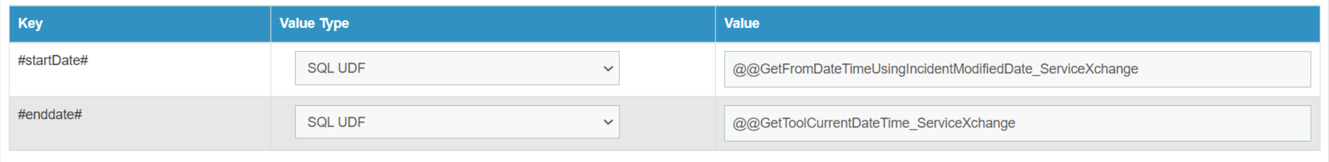
Request Authentication Parameters
Add Authentication Parameters
Delete All

Key	Value	Is Encrypted	Is Key	Action
username	<username>	□	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	□	□	🗑
AuthPrefix	Bearer	□	□	🗑
client_id	<clientID>	☑	☑	🗑
client_secret	<clientsrcret>	☑	☑	🗑
TokenKey	access_token	□	□	🗑
ResponseType	JSON	□	□	🗑
grant_type	Password	□	☑	🗑

Figure 275 – Create Data Source (Request Authentication Parameters for OAuth 2.0)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

Key	Value Type	Value
#start_date#	SQL UDF	@@GetFromDateTimeUsingIncidentModifiedDate_ServiceXchange
#end_date#	SQL UDF	@@GetToolCurrentDateTime_ServiceXchange



Key	Value Type	Value
#startDate#	SQL UDF	@@GetFromDateTimeUsingIncidentModifiedDate_ServiceXchange
#enddate#	SQL UDF	@@GetToolCurrentDateTime_ServiceXchange

Figure 276 – URL Path Parameters

– **Request Header Parameters –** Please enter the request header parameters as required.

– **Response Body –** In this section, please enter the output of URL query for one of the incidents in JSON format. A sample response is mentioned below:

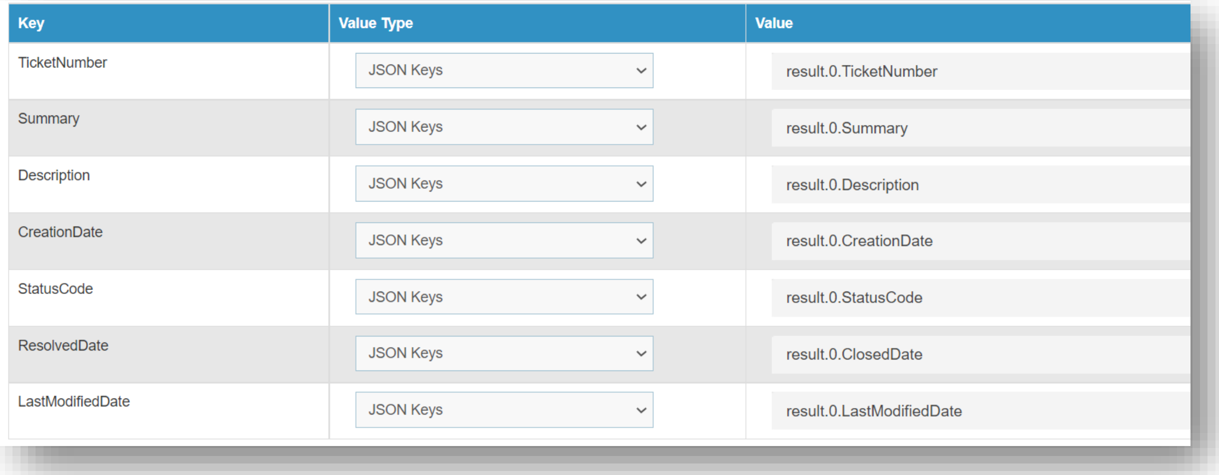
```
Response Body –

{

    "statusCode": 200,
    "status": "Success",

    "message": null,

    "result": [

        {

            "TicketNumber": "INC0303869",

            "Summary": "testing",

            "Description": "testing data",

            "AssignedGroup": "02cc6a39376e4f00c72b2b2943990e68",

            "StatusCode": "1",

            "CreationDate": "2022-09-23 09:26:52.000",

            "LastModifiedDate": "2022-09-23 09:26:52.000",

            "ClosedDate": "2022-09-22 06:24:52.000",

            "sys_id": "2b535ab3dbc988506d7550d3dc96190e",

            "Col1": "",

            "Col2": "",

            "Col3": "",

            "Col4": "",

            "Col5": "",

            "Col6": "",

            "Col7": "",

            "Col8": "",

            "Col9": "",

            "Col10": "",

            "iAutomate_CreatedDateInGMT":"2022-09-23
09:27:22.773",

            "iAutomate_UpdatedDateInGMT": "2022-09-23
09:27:22.773"
```

```
            }
      ]
}
```

- After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section**.**
- **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below:

Table 52– Sample Mandatory Parameter Mapping

Key	Value Type	Value
TicketNumber	JSON.Keys	result.0.TicketNumber
Summary	JSON.Keys	result.0.Summary
Description	JSON.Keys	result.0.Description
CreationDate	JSON.Keys	result.0.CreationDate
StatusCode	JSON.Keys	result.0.StatusCode
ResolvedDate	JSON.Keys	result.0.ClosedDate
LastModifiedDate	JSON.Keys	result.0.LastModifiedDate



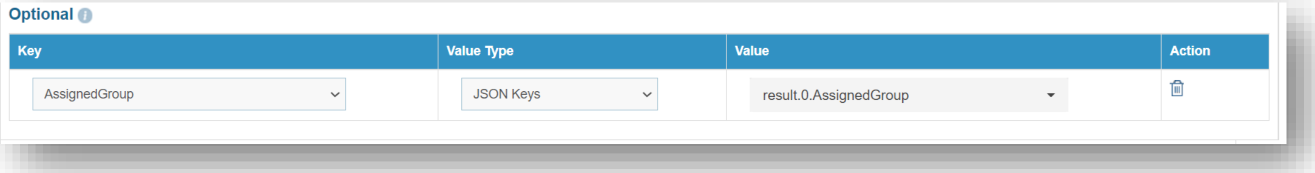
Key	Value Type	Value
TicketNumber	JSON Keys	result.0.TicketNumber
Summary	JSON Keys	result.0.Summary
Description	JSON Keys	result.0.Description
CreationDate	JSON Keys	result.0.CreationDate
StatusCode	JSON Keys	result.0.StatusCode
ResolvedDate	JSON Keys	result.0.ClosedDate
LastModifiedDate	JSON Keys	result.0.LastModifiedDate

Figure 277 – Mandatory Parameter Mapping

- If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 53 - Sample Optional Parameters

Key	Value Type	Value
AssignedGroup	JSON.Keys	result.0.AssignedGroup



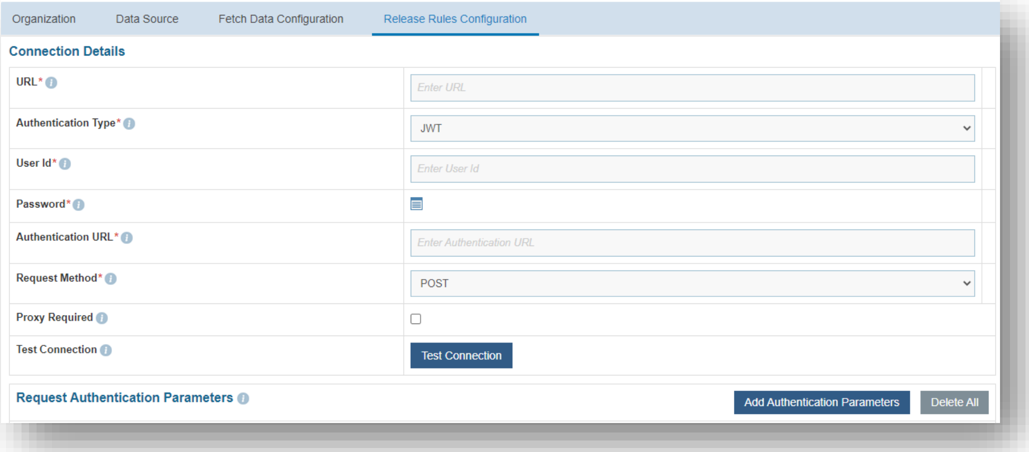
Optional

Key	Value Type	Value	Action
AssignedGroup	JSON Keys	result.0.AssignedGroup	🗑

Figure 278 – Optional Parameter Mapping

- Click Next to proceed to Release Rules Configuration.
- On **Release Rules Configuration** tab, type in the details as per the requirement.
- In the **Connection Details** section, enter the following details:
  - **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.

    Sample URL - https://inboundBoomiDevCHN1.dryicehcl.com/ws/simple/updateIncidentInSX
  - **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously. For e.g., **Basic**.
  - **Request Method** – Select Request Method as POST from the drop-down.
  - **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.



Connection Details

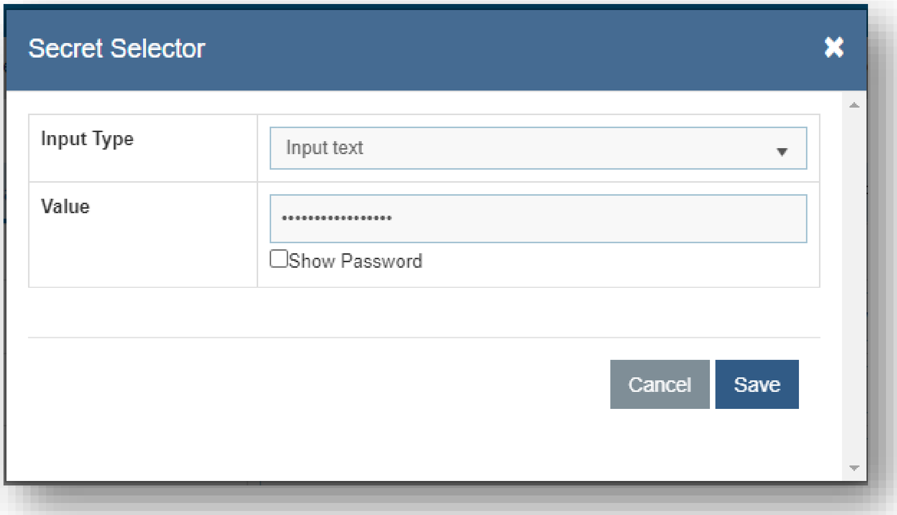
URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	
Authentication URL*	Enter Authentication URL
Request Method*	POST
Proxy Required	☐
Test Connection	Test Connection

Request Authentication Parameters

Figure 279 – Release Rules Configuration (Connection Details)

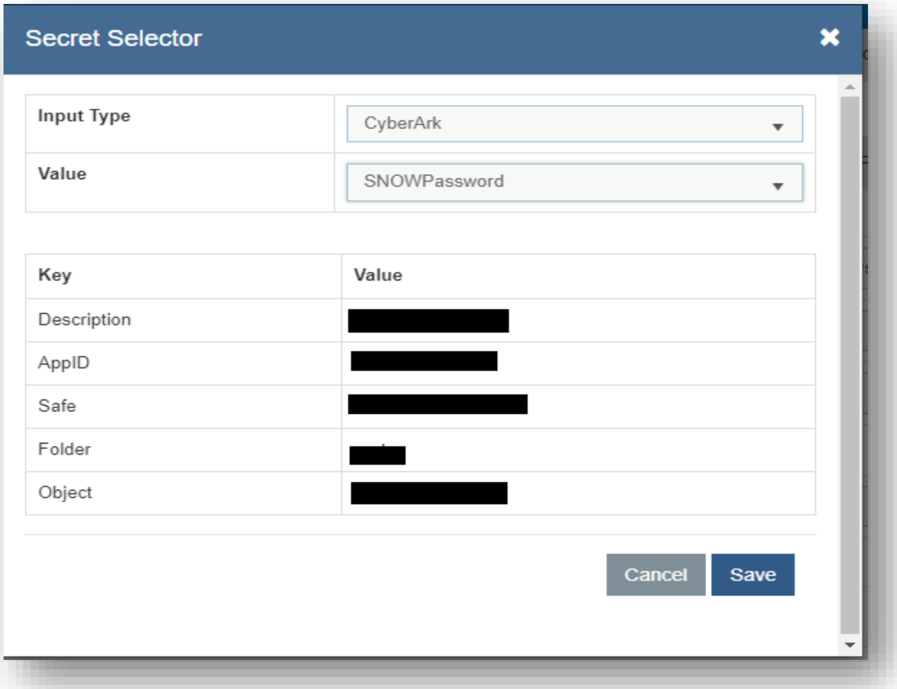
- **Password** – For password, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in

any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Input Type	Input text
Value	•••••••••••••••• □Show Password

Figure 280 – Password in plaintext



Input Type	CyberArk
Value	SNOWPassword

Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

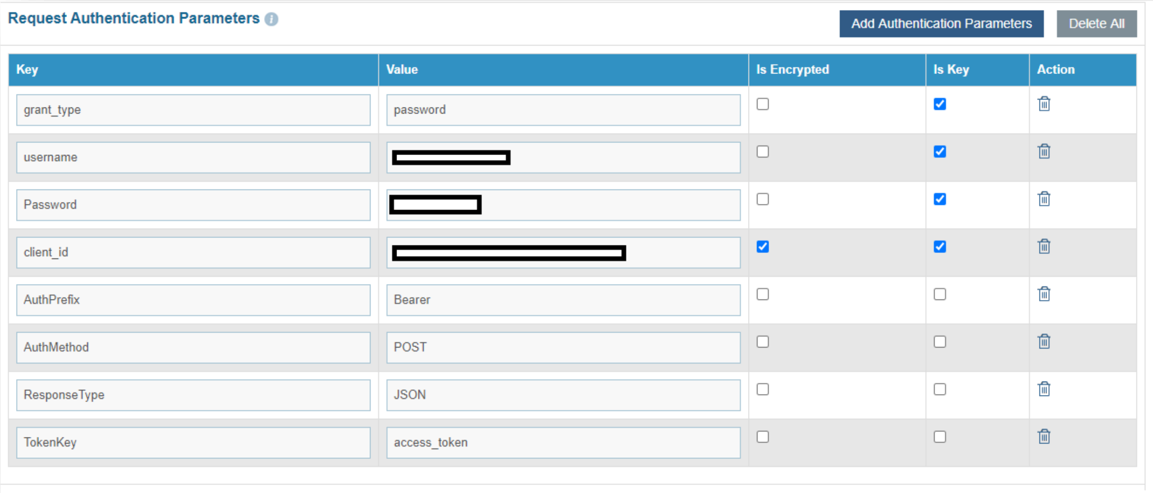
Figure 281 – Password from Key Vault (CyberArk)

– **Request Authentication Parameters -** If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.

– Based on the **Authentication Type**, add the parameters mentioned in the below table

Table 54 - Sample Authentication Parameters

Key	Value	Is Encrypted?	Is Key?
grant_type	password	N	Y
username	<username>	N	Y
Password	<password>	Y	Y
client_id	<client_id>	N	Y
AuthPrefix	Bearer	N	N
AuthMethod	POST	N	N
ResponseType	JSON	N	N
TokenKey	access_token	N	N



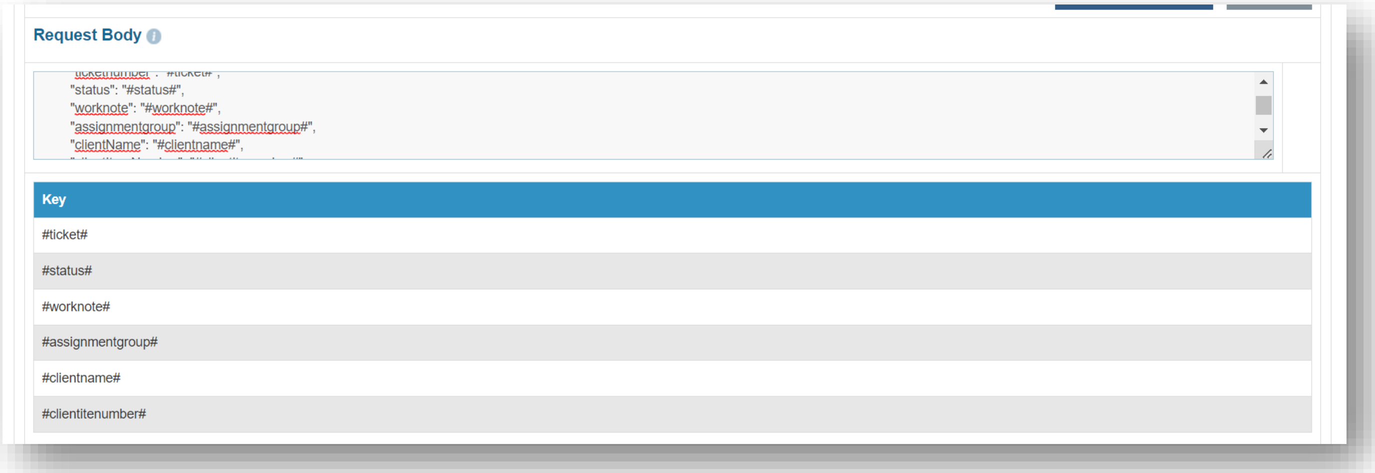
Request Authentication Parameters

Key	Value	Is Encrypted	Is Key	Action
grant_type	password	☐	☑	🗑
username	[redacted]	☐	☑	🗑
Password	[redacted]	☐	☑	🗑
client_id	[redacted]	☑	☑	🗑
AuthPrefix	Bearer	☐	☐	🗑
AuthMethod	POST	☐	☐	🗑
ResponseType	JSON	☐	☐	🗑
TokenKey	access_token	☐	☐	🗑

Figure 282 – Create Data Source (Request Authentication Parameters)

– **Request Body -** In this section, please enter the request body in JSON format. A sample request is mentioned below:

```
Request Body –

{

        "ticketnumber": "#ticket#",

        "status": "#status#",

        "worknote": "#worknote#",

        "assignmentgroup":"#assignmentgroup#",

        "clientName": "#clientname#",
        "clientItemNumber": "#clientitenumber#"

}
```



Request Body

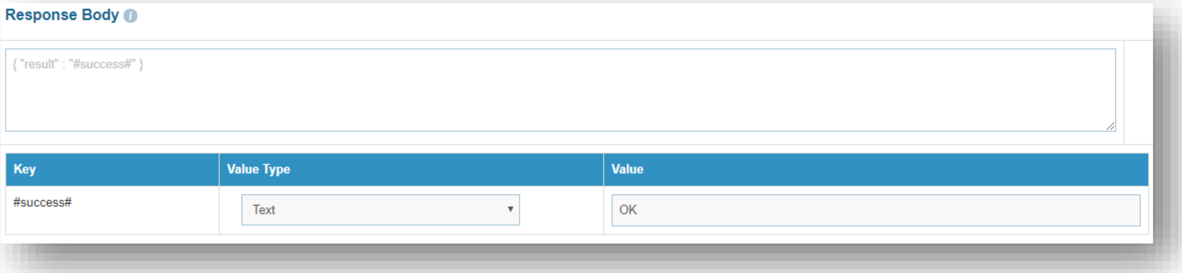
```
        "ticketnumber": "#ticket#",
        "status": "#status#",
        "worknote": "#worknote#",
        "assignmentgroup": "#assignmentgroup#",
        "clientName": "#clientname#",
```

Key
#ticket#
#status#
#worknote#
#assignmentgroup#
#clientname#
#clientitenumber#

Figure 283 – Request Body (Key)

- **Response Body –** In this section, please enter the response body in JSON format. A sample request is mentioned below:

```
Response Body –

{ "result" : "#success#" }
```



Response Body

```
{ "result" : "#success#" }
```

Key	Value Type	Value
#success#	Text	OK

Figure 284 – Release Rules Configuration (Response Body)

- **Response Key Value** mapping can be done as per the below table.

Table 55 - Sample Response Key Value Mapping

#success#	Text	OK

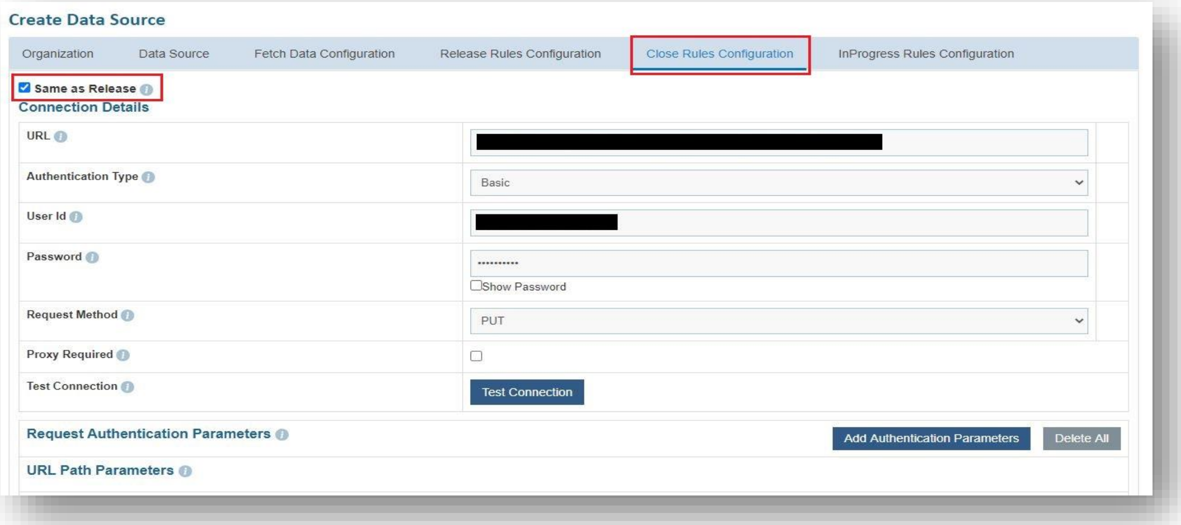
- Click **Submit** to add the data source.
- On **Close Rules Configuration** tab, type in the details as per the requirement.

– In the **Connection Details** section, enter the following details:

- Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.

  Sample URL - https://inboundBoomiDevCHN1.dryicehcl.com/ws/simple/updateIncidentInSX

- **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously. For e.g., **Basic**.
- **Request Method** – Select Request Method as POST from the drop-down.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.


Create Data Source
Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration
Test Connection
Request Authentication Parameters
Add Authentication Parameters
Delete All
URL Path Parameters

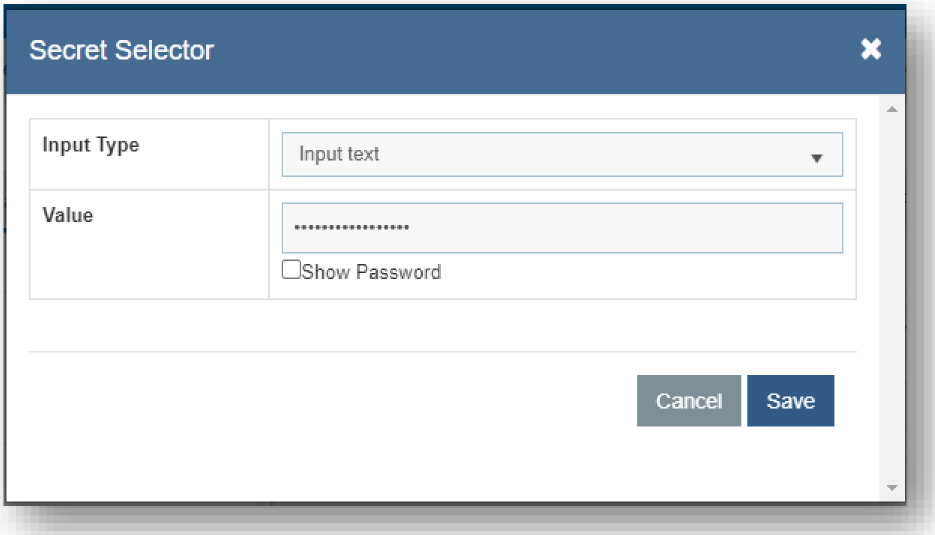
☑ Same as Release

Connection Details

URL	[redacted]
Authentication Type	Basic
User Id	[redacted]
Password	•••••••••• □Show Password
Request Method	PUT
Proxy Required	□

Figure 285 – Close Rules Configuration (Connection Details)

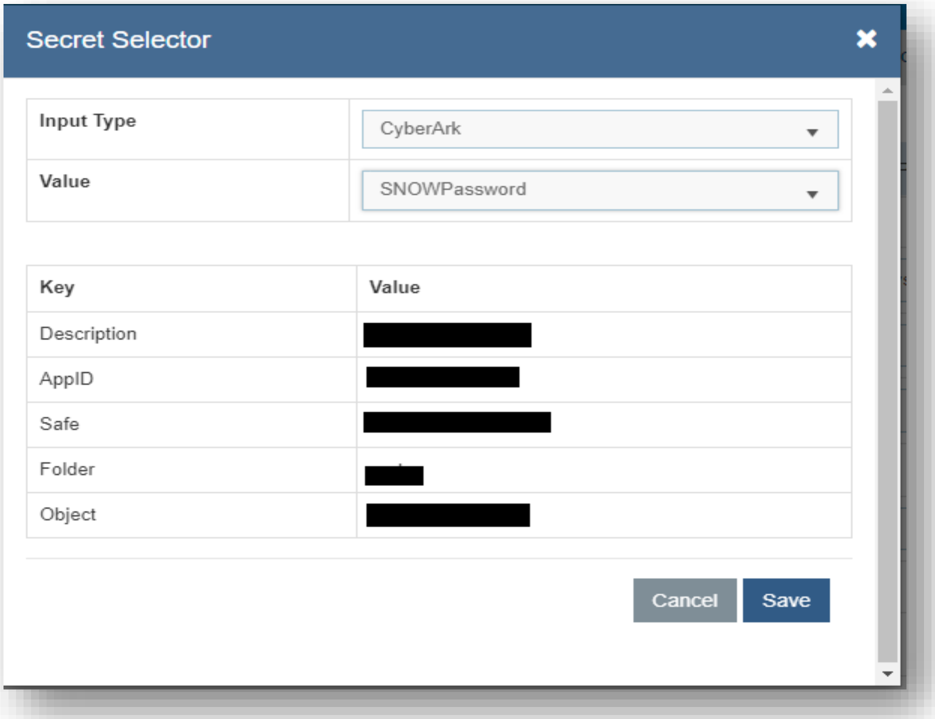
- **Password** – For password, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Secret Selector

Input Type	Input text
Value	••••••••••••••••• □ Show Password

Figure 286 – Password in plaintext



Secret Selector

Input Type	CyberArk
Value	SNOWPassword

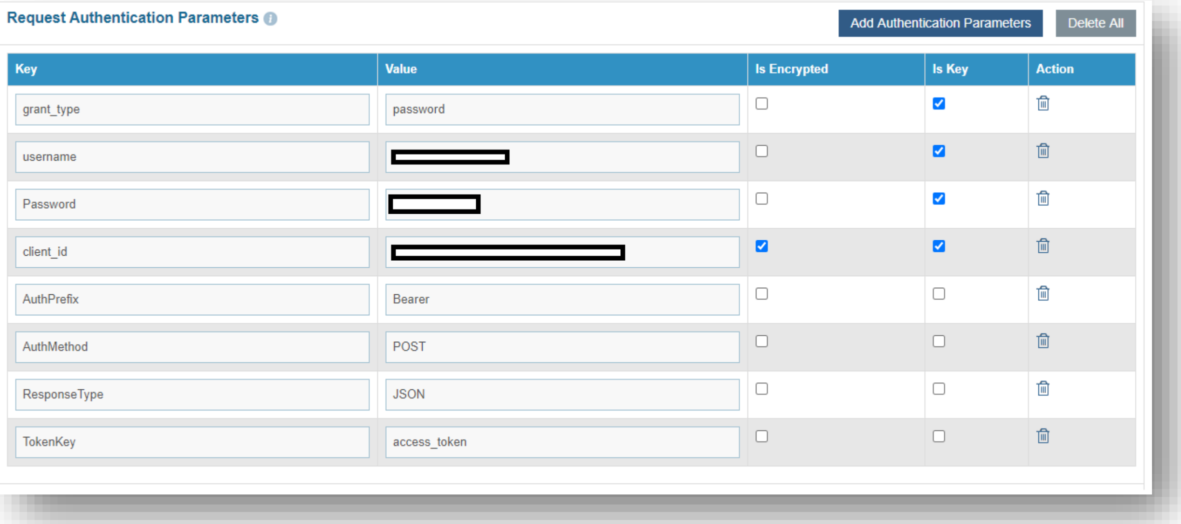
Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 287 – Password from Key Vault (CyberArk)

- **Request Authentication Parameters -** If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.
- Based on the **Authentication Type**, add the parameters mentioned in the below table –

Table 56 - Sample Authentication Parameters

Key	Value	Is Encrypted?	Is Key?
grant_type	password	N	Y
username	<username>	N	Y
Password	<password>	Y	Y
client_id	<client_id>	N	Y
AuthPrefix	Bearer	N	N
AuthMethod	POST	N	N
ResponseType	JSON	N	N
TokenKey	access_token	N	N

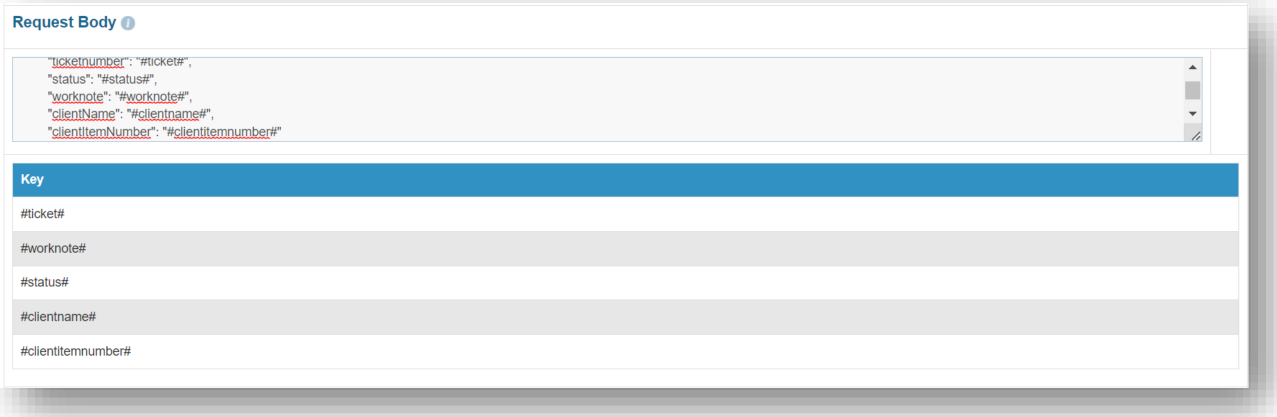

Request Authentication Parameters ⓘ
Add Authentication Parameters
Delete All

Key	Value	Is Encrypted	Is Key	Action
grant_type	password	□	☑	🗑
username	[redacted]	□	☑	🗑
Password	[redacted]	□	☑	🗑
client_id	[redacted]	☑	☑	🗑
AuthPrefix	Bearer	□	□	🗑
AuthMethod	POST	□	□	🗑
ResponseType	JSON	□	□	🗑
TokenKey	access_token	□	□	🗑

Figure 288 – Create Data Source (Request Authentication Parameters)

– **Request Body -** In this section, please enter the request body in JSON format. A sample request is mentioned below:

```
Request Body –

{

        "ticketnumber": "#ticket#",

        "status": "#status#",

        "worknote": "#worknote#",

        "clientName": "#clientname#",

        "clientItemNumber": "#clientitenumber#"

    }
```



Request Body

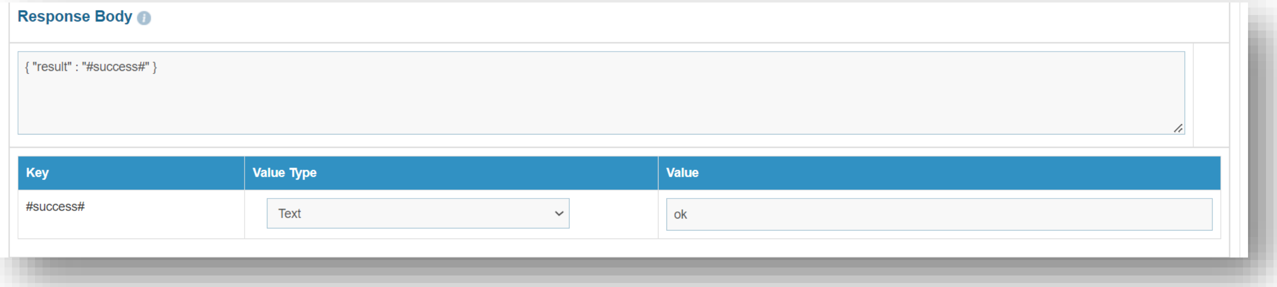
```
    "ticketnumber": "#ticket#",
    "status": "#status#",
    "worknote": "#worknote#",
    "clientName": "#clientname#",
    "clientItemNumber": "#clientitemnumber#"
```

Key
#ticket#
#worknote#
#status#
#clientname#
#clientitemnumber#

Figure 289 – Close Rules Configuration (Request Body)

- **Response Body –** In this section, please enter the response body in JSON format. A sample request is mentioned below:

```
Response Body –

{ "result" : "#success#" }
```



Response Body

```
{ "result" : "#success#" }
```

Key	Value Type	Value
#success#	Text	ok

Figure 290 – Close Rules Configuration (Response Body)

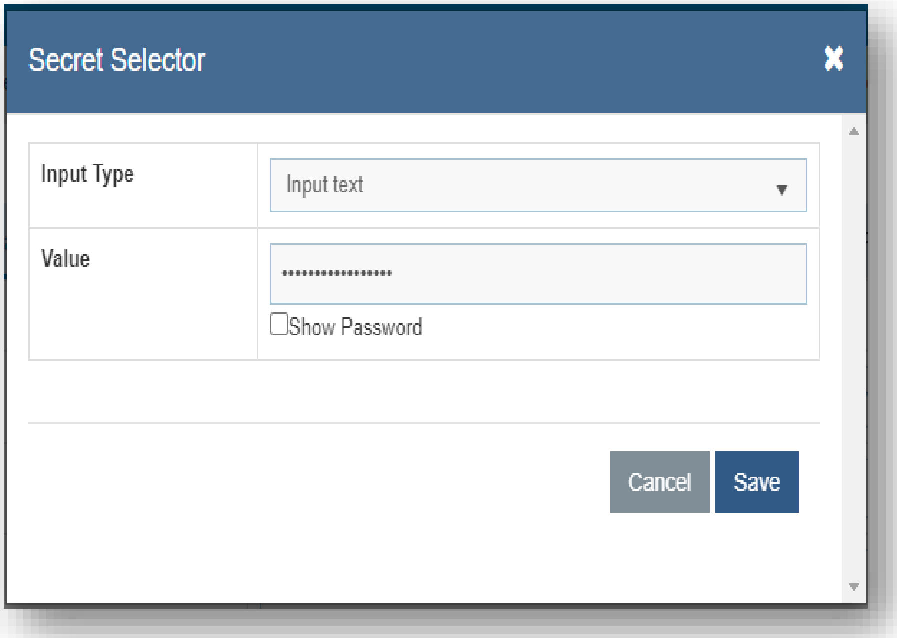
- **Response Key Value** mapping can be done as per the below table.

Table 57 - Sample Response Key Value Mapping

#success#	Text	OK

- Click **Submit** to add the data source.
- On **InProgress Rules Configuration** tab, type in the details as per the requirement.
- In the **Connection Details** section, enter the following details:
  - **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.

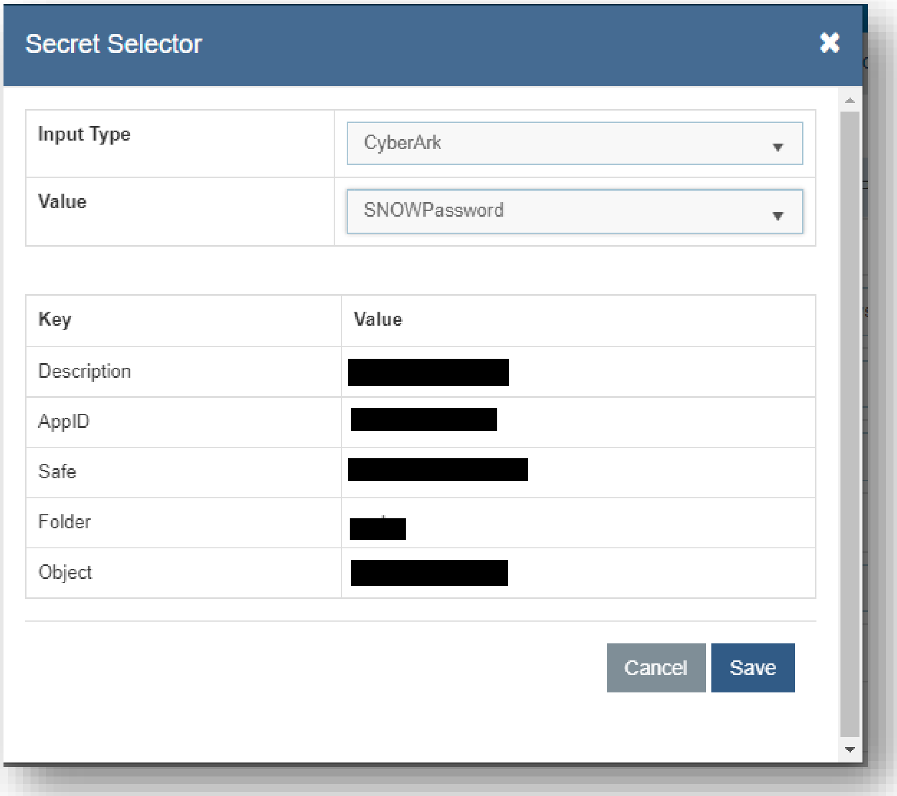
- **Sample URL** - https://inboundBoomiDevCHN1.dryicehcl.com/ws/simple/updateIncidentInSX
- **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously. For e.g., **Basic**.
- **Request Method** – Select Request Method as POST from the drop-down.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- **Password** – For password, click on icon next to it. If the password is available in plaintext, then select Input type as Input Text and enter the password in Value field. Else if it is available in any Key Vault such as CyberArk then select Input Type as CyberArk and then select any of the configured details from the value field.



Secret Selector

Input Type	Input text
Value	•••••••••••••••• ☐ Show Password

Figure 291 – Password in plaintext



Secret Selector

Input Type	CyberArk
Value	SNOWPassword

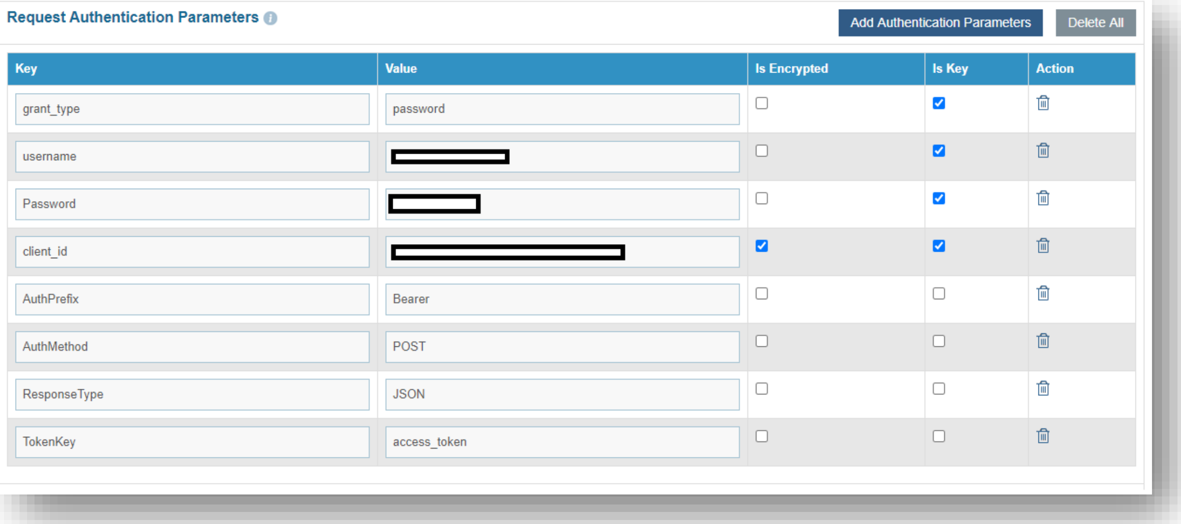
Key	Value
Description	[redacted]
AppID	[redacted]
Safe	[redacted]
Folder	[redacted]
Object	[redacted]

Figure 292 – Password from Key Vault (CyberArk)

- **Request Authentication Parameters -** If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.
- Based on the **Authentication Type**, add the parameters mentioned in the below table

Table 58– Sample Authentication Parameters

Key	Value	Is Encrypted?	Is Key?
grant_type	password	N	Y
username	<username>	N	Y
Password	<password>	Y	Y
client_id	<client_id>	N	Y
AuthPrefix	Bearer	N	N
AuthMethod	POST	N	N
ResponseType	JSON	N	N
TokenKey	access_token	N	N



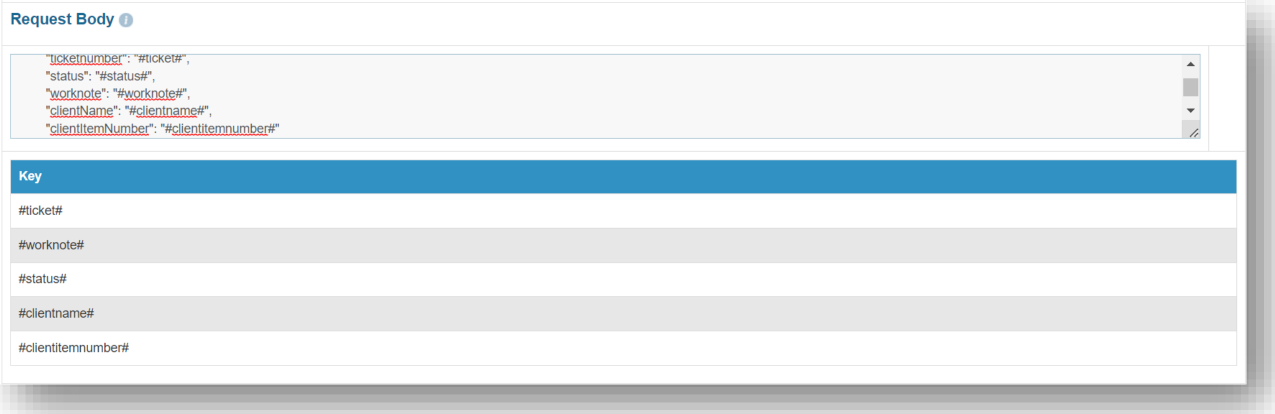
Request Authentication Parameters

Key	Value	Is Encrypted	Is Key	Action
grant_type	password	□	☑	🗑
username	[redacted]	□	☑	🗑
Password	[redacted]	□	☑	🗑
client_id	[redacted]	☑	☑	🗑
AuthPrefix	Bearer	□	□	🗑
AuthMethod	POST	□	□	🗑
ResponseType	JSON	□	□	🗑
TokenKey	access_token	□	□	🗑

Figure 293 – Create Data Source (Request Authentication Parameters)

- **Request Body -** In this section, please enter the request body in JSON format. A sample request is mentioned below:

```
Request Body –
{
        "ticketnumber": "#ticket#",
        "status": "#status#",
        "worknote": "#worknote#",

        "clientName": "#clientname#",

        "clientItemNumber": "#clientitemnumber#"

 }
```



Request Body

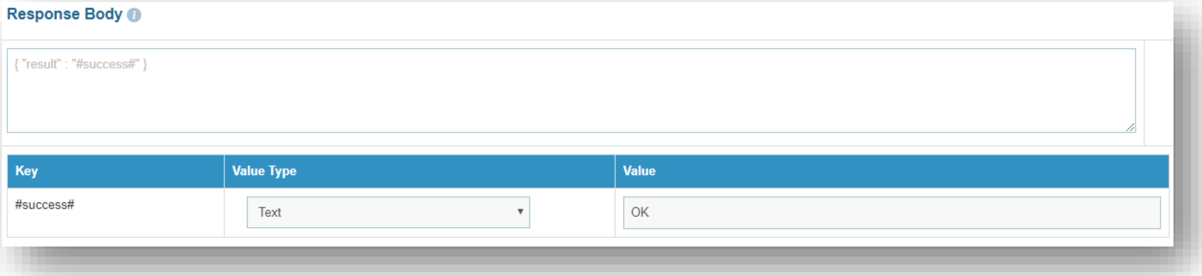
```
     "ticketnumber": "#ticket#",
     "status": "#status#",
     "worknote": "#worknote#",
     "clientName": "#clientname#",
     "clientItemNumber": "#clientitemnumber#"
```

Key
#ticket#
#worknote#
#status#
#clientname#
#clientitemnumber#

Figure 294 - Request body

- **Response Body –** In this section, please enter the response body in JSON format. A sample request is mentioned below:

```
Response Body –

{ "result" : "#success#" }
```



Response Body

```
{ "result" : "#success#" }
```

Key	Value Type	Value
#success#	Text	OK

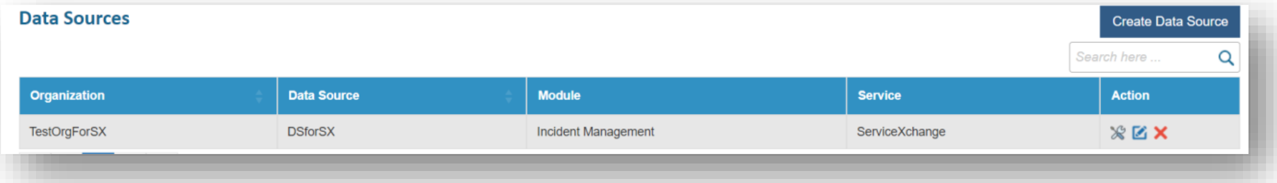
Figure 295 – InProgress Rules Configuration (Response Body)

- **Response Key Value** mapping can be done as per the below table -

Table 59 - Sample Response Key Value Mapping

#success#	Text	OK

- Click **Submit** to add the data source.
- In order to bring the tickets within BigFix Runbook AI scope, a specific queue needs to be configured in the ITSM tool and same has to be configured in BigFix Runbook AI. This is achieved through **Manage the Entry Criteria**. Please perform the below steps:
  - Go to Actions tab and click Manage Data Sources.
  - On the **Data Sources** tab, click 🛠 next to the data source user wants to manage. **Manage Entry Criteria** screen appears.

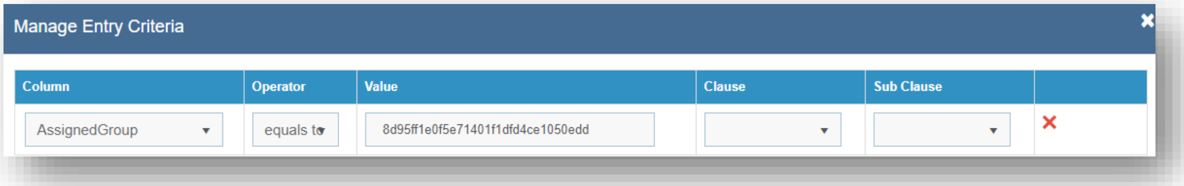


Data Sources

Organization	Data Source	Module	Service	Action
TestOrgForSX	DSforSX	Incident Management	ServiceXchange	🛠 ✎ ✕

Figure 296 – Manage Entry Criteria

  - Select 'AssignedGroup' for the **Column field and** 'equals to' for the **Operator** field.
  - Enter the sys_id of the assignment group in SX in the **Value** field.
  - **Clause** and **Sub-Clause** fields can also be added based on requirement.

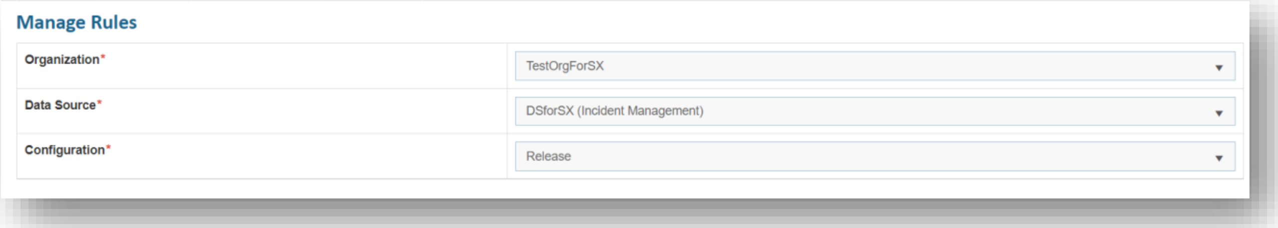


Manage Entry Criteria

Column	Operator	Value	Clause	Sub Clause	
AssignedGroup ▾	equals to ▾	8d95ff1e0f5e71401f1dfd4ce1050edd	▾	▾	✕

Figure 297 – Manage Entry Criteria (cont.)

  - Click **Save**.
- To configure the rules for the data source created earlier, perform the below steps:
  - Go to **Actions Tab → Runbooks** and then click Manage Rules.
  - Select the **Organization** and the data source created from **Data Source** dropdown.

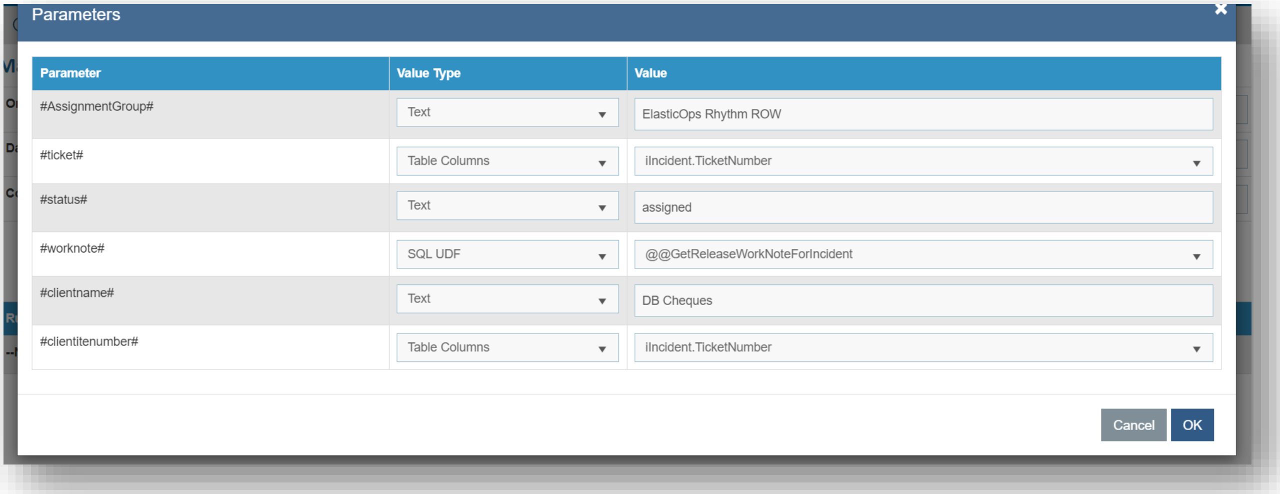


Manage Rules

Organization*	TestOrgForSX
Data Source*	DSforSX (Incident Management)
Configuration*	Release

Figure 298 – Manage Rules

- Click on ⚙ corresponding to **–No Rule—**
- Map the parameter **#Assignmentgroup#** with **ElasticOps Rhythm ROW** as value and value Type is Text.
- Map the parameter **#ticket#** with **iIncident.TicketNumber** as value and value type is Table Columns.
- Map the parameter **#status#** with **Assigned** as value and text as Value Type.
- Map the parameter **#clientname#** with **DB Cheques** as value and text as Value Type.
- Map the parameter **#clientitemnumber#** with **iIncident.TicketNumber** as value and table column as Value Type.
- Map the parameter **#worknote#** with **@@GetReleaseWorkNoteForIncident** as Value and SQL UDF as Value Type.
- Click **OK**.



Parameters

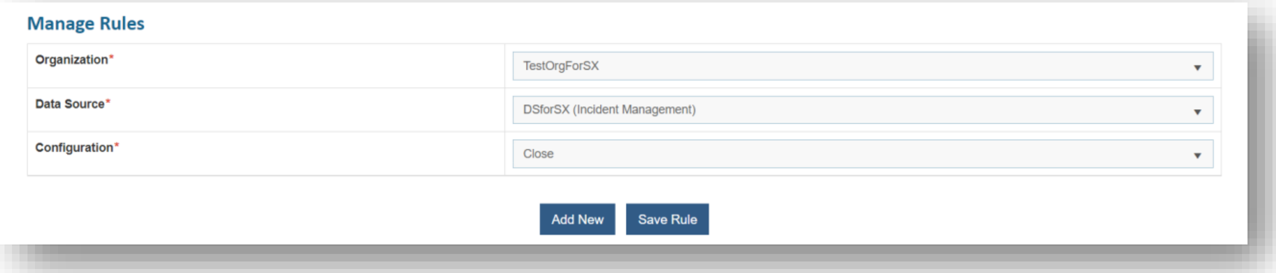
Parameter	Value Type	Value
#AssignmentGroup#	Text	ElasticOps Rhythm ROW
#ticket#	Table Columns	iIncident.TicketNumber
#status#	Text	assigned
#worknote#	SQL UDF	@@GetReleaseWorkNoteForIncident
#clientname#	Text	DB Cheques
#clientitenumber#	Table Columns	iIncident.TicketNumber

Figure 299 – Manage Rules (cont.)

- Click Save Rule.

– To configure the **Close rules** for the data source created earlier, perform the below steps:

- Go to **Actions Tab** and select **Runbooks** and then click **Manage Rules**.
- Select the **Organization** and the data source created from **Data Source** dropdown.

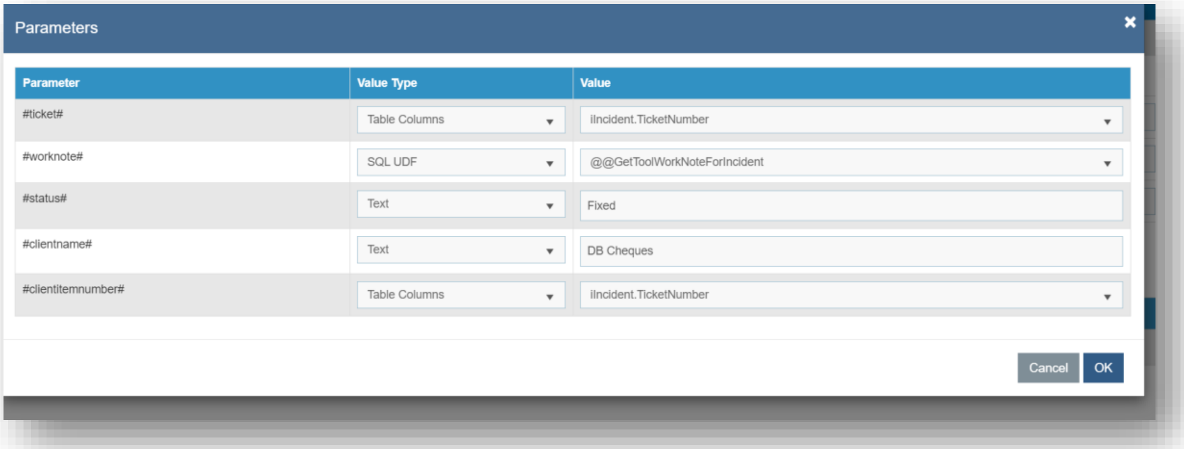


Manage Rules

Organization*	TestOrgForSX ▾
Data Source*	DSforSX (Incident Management) ▾
Configuration*	Close ▾

Figure 300 – Manage Rules (cont.)

- Click on ⚙ corresponding to **–No Rule—**
- Map the parameter **#ticket#** with **iIncident.TicketNumber** as value and value type is Table Columns.
- Map the parameter **#status#** with **Fixed** as value and text as Value Type.
- Map **#worknote#** again to the value type as SQL UDF in which #worknote# was mapped with function **@@GetToolWorkNoteForIncident**.
- Map the parameter **#clientname#** with **DB Cheques** as value and text as Value Type.
- Map the parameter **#clientitemnumber#** with **iIncident.TicketNumber** as value and table column as Value Type



Parameters

Parameter	Value Type	Value
#ticket#	Table Columns ▾	iIncident.TicketNumber ▾
#worknote#	SQL UDF ▾	@@GetToolWorkNoteForIncident ▾
#status#	Text ▾	Fixed
#clientname#	Text ▾	DB Cheques
#clientitemnumber#	Table Columns ▾	iIncident.TicketNumber ▾

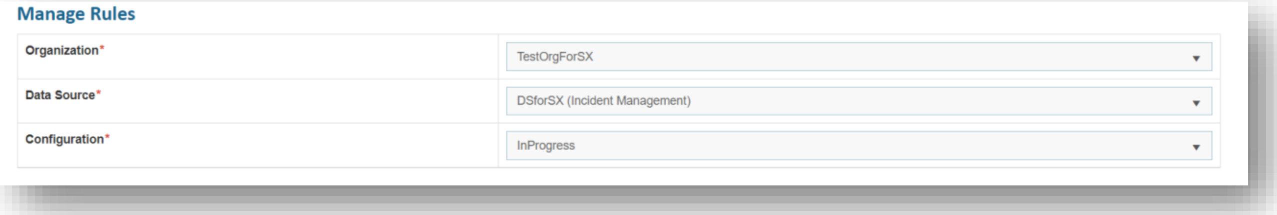
Figure 301 – Manage Rules (cont.)

- Click **OK**.

- Click Save Rule.

– To configure the InProgress rules for the data source created earlier, perform the below steps:

- Go to Actions Tab → Runbooks and then click Manage Rules.
- Select the **Organization** and the data source created from **Data Source** dropdown.

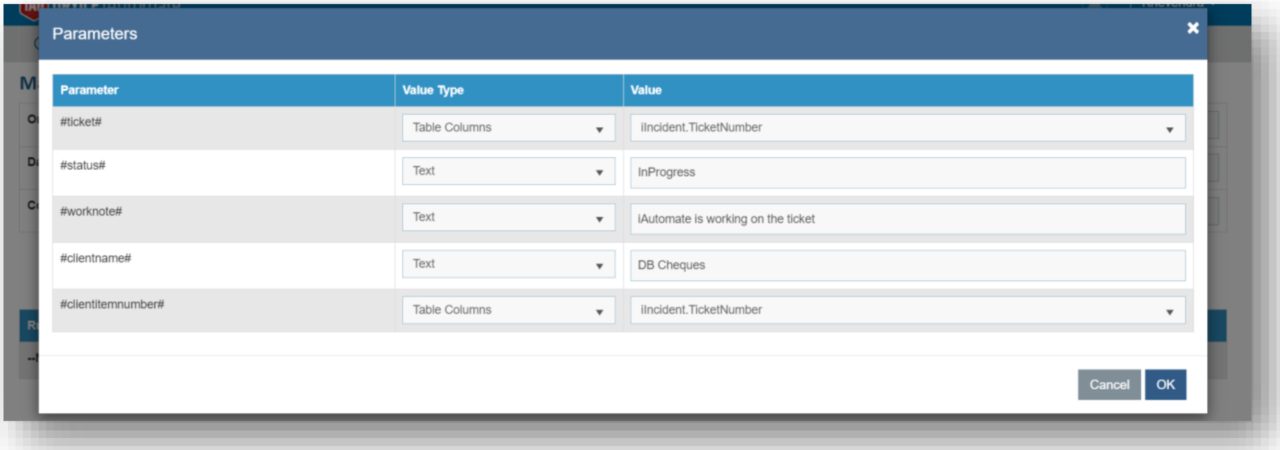


Manage Rules

Organization*	TestOrgForSX
Data Source*	DSforSX (Incident Management)
Configuration*	InProgress

Figure 302 – Manage Release Rules

- Click on ⚙ corresponding to **–No Rule—**
- Map the parameter **#ticket#** with **iIncident.TicketNumber** as value and value type is Table Columns.
- Map the parameter **#status#** with **InProgress** as value and text as Value Type.
- Map the parameter #worknote# with BigFix Runbook AI is working on the ticket as Value and text as Value Type.
- Map the parameter **#clientname#** with **DB Cheques** as value and text as Value Type.
- Map the parameter **#clientitemnumber#** with **iIncident.TicketNumber** as value and table column as Value Type.



Parameters

Parameter	Value Type	Value
#ticket#	Table Columns	iIncident.TicketNumber
#status#	Text	InProgress
#worknote#	Text	iAutomate is working on the ticket
#clientname#	Text	DB Cheques
#clientitemnumber#	Table Columns	iIncident.TicketNumber

Figure 303 – Manage Rules (cont.)

- Click **OK**.
- Click Save Rule.

Integration with Event Management Tools

Any Event Management tool acts as a data source for BigFix Runbook AI from where it pulls the event or Probable Root Cause data and then performs appropriate actions for resolution. Thus, to enable integration with Event Management, it requires for a data source to be created as part of BigFix Runbook AI configuration.

Before proceeding with the configuration related to Data Source creation, user has to ensure that an organization has been configured. If not done already, please refer to the Configuration Guide for the same and create the organization before proceeding ahead.

Please note that for integration with Event Management tool, while creating the organization, user needs to select the Event Management tool from the dropdown.

4.8 Integration with Moogsoft

4.8.1 Incident Management with ITSM (ServiceNow)

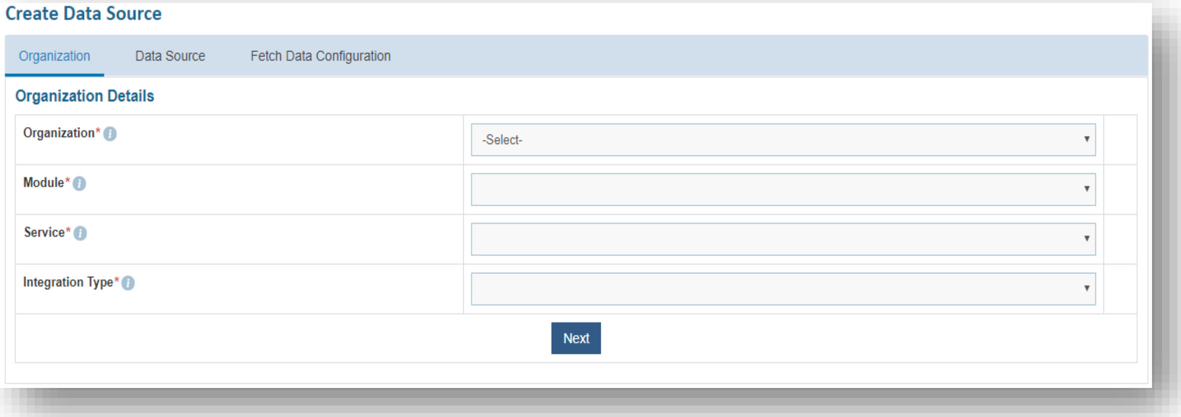
This scenario is applicable when the ITSM tools is available in the client environment and both event management & BigFix Runbook AI is integrated with the ITSM, which acts as a system of record. The event data flows from event management tool to the ITSM leading to a ticket, based on the probable root cause. Upon ticket creation, BigFix Runbook AI picks the ticket from the ITSM tool and performs the appropriate action for resolution.

The user has the option to view the tickets and trigger the resolutions via Moogsoft as well as BigFix Runbook AI console.

To create a data source, perform the following steps:

- On the main menu bar, click **Actions → Manage Data Sources**.
- The **Create Data Source** page appears with the following tabs:
  - Organization

- Data Source
- Fetch Data Configuration
- Release Rules Configuration
- Close Rules Configuration (Optional – applicable only when the ticket closure status update is managed by BigFix Runbook AI directly instead of RBA tool)
- InProgress Rules Configuration (Optional – applicable only when the ticket's in progress status updates is managed by BigFix Runbook AI directly instead of RBA tool)



Create Data Source

Organization Details

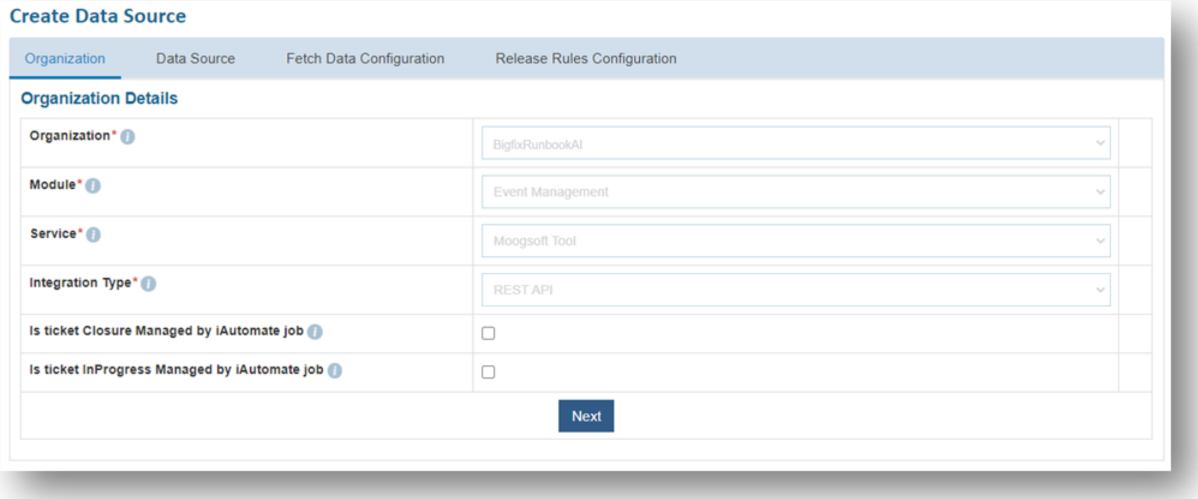
Organization*	-Select-
Module*	
Service*	
Integration Type*	

Figure 304 – Create Data Source

Release Rules Configuration is only applicable for the following **Module** types- **Incident Management, Change Request Task and Service Request Task.** This tab will not be activated for other module types.

- On the **Organization** tab,
    - Select the **Organization Name** from the dropdown.
    - Select the **Module** as **Event Management,** since we are configuring this data source for pulling the event data.
    - Select the **Service** as **Moogsoft Tool** as we are configuring the data source for Moogsoft
    - Select the **Integration Type** as **REST**, since we will be integrating through REST APIs.
    - Check **Is ticket Closure Managed by BigFix Runbook AI job** if you want BigFix Runbook AI to manage the ticket closure updates instead of the RBA tool. In this scenario, an additional tab **Close Rules Configuration** will be activated for providing further details, steps for which are mentioned later.
    - Check "**Is ticket InProgress Managed by BigFix Runbook AI job**" if you want BigFix Runbook AI to manage the ticket's in progress status updates instead of the RBA tool. In this scenario, an additional tab "**InProgress Rules Configuration**" will be activated for providing further details, steps for which are mentioned later.

- Click **Next**.



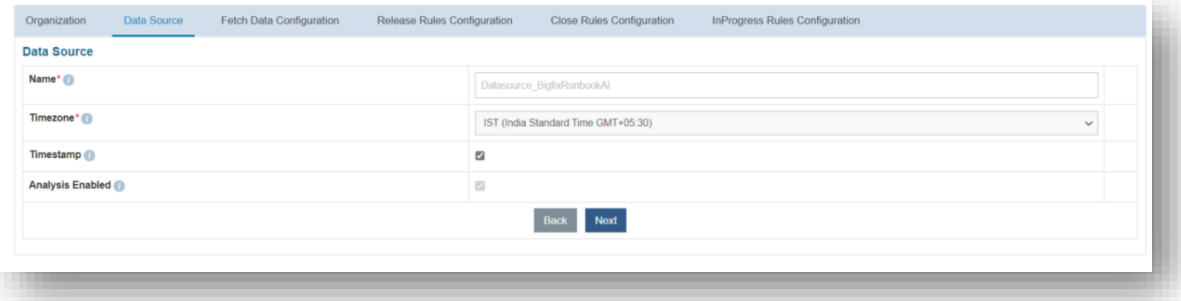
Organization Details

Organization*	BigfixRunbookAI
Module*	Event Management
Service*	Moogsoft Tool
Integration Type*	REST API
Is ticket Closure Managed by iAutomate job	☐
Is ticket InProgress Managed by iAutomate job	☐

Figure 305 – Create Data Source (cont.)

– On the **Data Source** tab,

- Type the new data source in the **Name** field.
- Select the **Timezone** to specify the time zone of the selected data source.
- Select **Timestamp** to view the present data with date and time.
- Select **Analysis Enabled?** if user wants to analyze the data retrieved from the data source.
- Click Next.


Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration
Back
Next

Data Source

Name*	Datasource_BigfixRunbookAI
Timezone*	IST (India Standard Time GMT+05:30)
Timestamp	☑
Analysis Enabled	☑

Figure 306 – Create Data Source (cont.)

– On the **Fetch Data Configuration** tab, type in the details as per the environment.

– In the **Connection Details** section, enter the following details:

- **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.

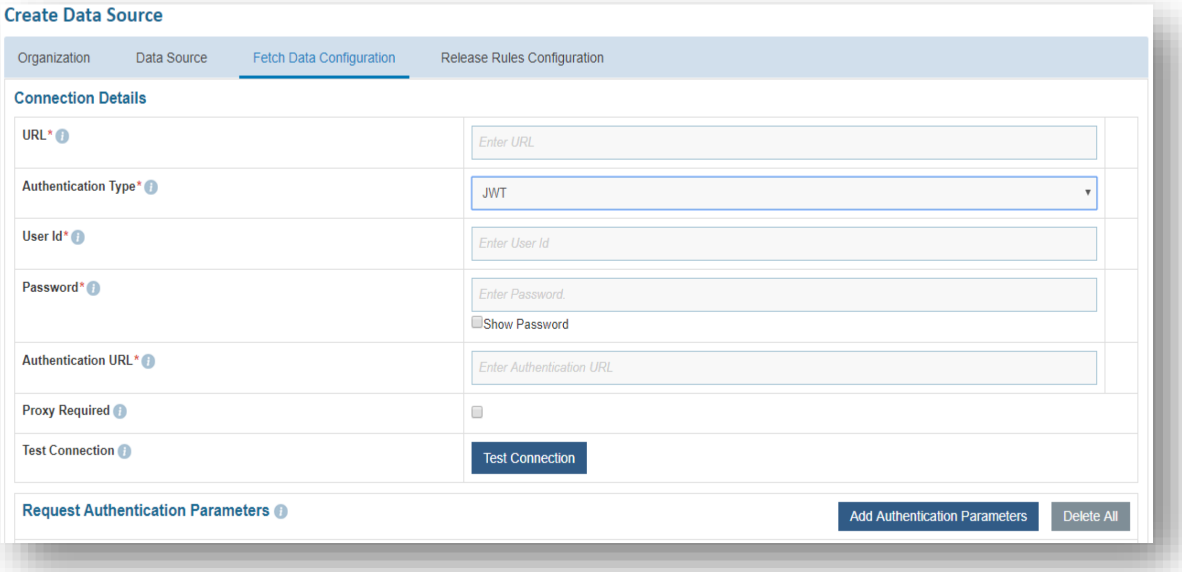
- **Sample URL** - https://URL.service-now.com/api/now/v1/table/incident?sysparm_fields=#Columns#&sysparm_query=sys_updated_on>=#StartDate#^sys_updated_on<=#EndDate#^ORDERBYsys_updated_on
- **Authentication Type** – Select one of the Authentication Types from Basic / Windows, JWT, OAuth 2.0

  Selection of **Basic / Windows** requires you to enter -

  - User Id
  - Password

  Selection of **JWT / OAuth 2.0** requires you to enter -

  - User Id
  - Password
  - Authentication URL
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



Create Data Source

Connection Details

URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	Enter Password □ Show Password
Authentication URL*	Enter Authentication URL
Proxy Required	□
Test Connection	Test Connection

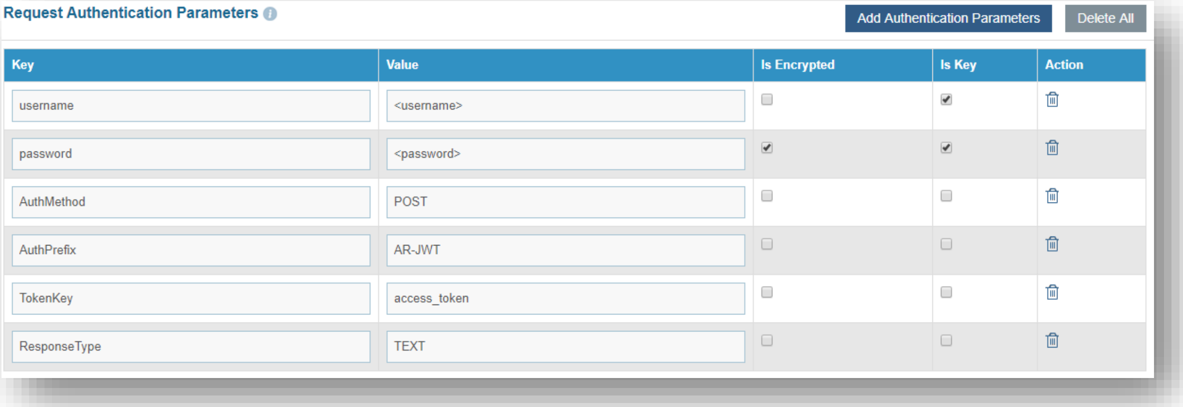
Request Authentication Parameters

Figure 307 – Create Data Source (Connection Details)

- **Request Authentication Parameters** – If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.
- Based on the **Authentication Type**, add the parameters mentioned in the below table.

Table 60– Sample Authentication Parameters

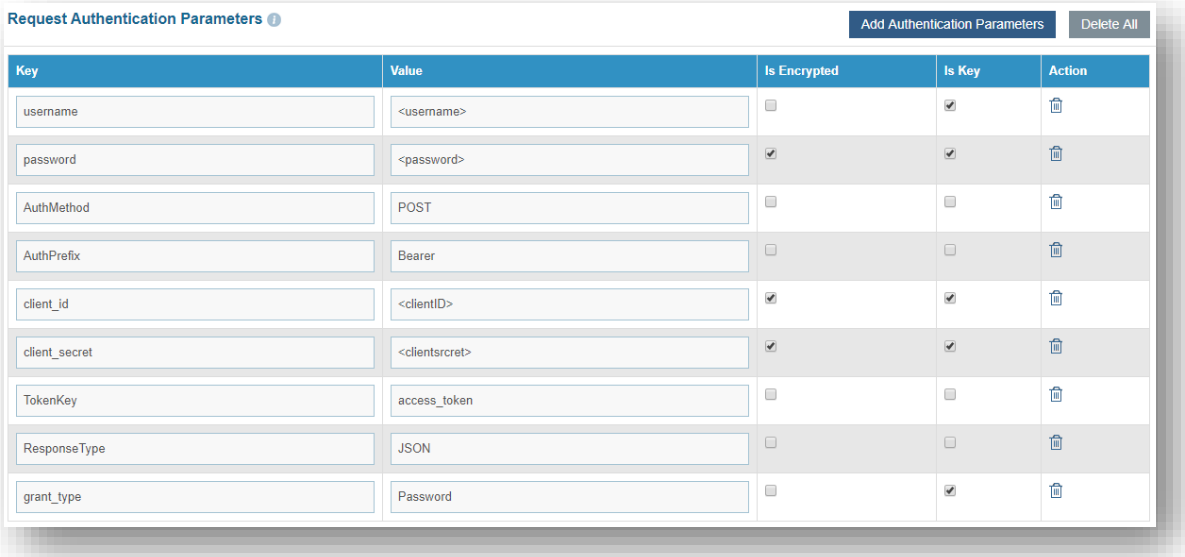
Authentication Type	Key	Value	Is Encrypted?	Is Key?
JWT	username	<username>	NO	YES
JWT	password	<password>	YES	YES
JWT	AuthMethod	POST	NO	NO
JWT	AuthPrefix	AR-JWT	NO	NO
JWT	TokenKey	access_token	NO	NO
JWT	ResponseType	TEXT	NO	NO
OAuth2.0	username	<username	NO	YES
OAuth2.0	password	<password>	YES	YES
OAuth2.0	AuthMethod	POST	NO	NO
OAuth2.0	AuthPrefix	Bearer	NO	NO
OAuth2.0	client_id	<clientID>	YES	YES
OAuth2.0	client_secret	<clientsecret>	YES	YES
OAuth2.0	TokenKey	access_token	NO	NO
OAuth2.0	ResponseType	JSON	NO	NO
OAuth2.0	grant_type	Password	NO	YES



Request Authentication Parameters

Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	AR-JWT	☐	☐	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	TEXT	☐	☐	🗑

Figure 308 – Create Data Source (Request Authentication Parameters for JWT)



Request Authentication Parameters

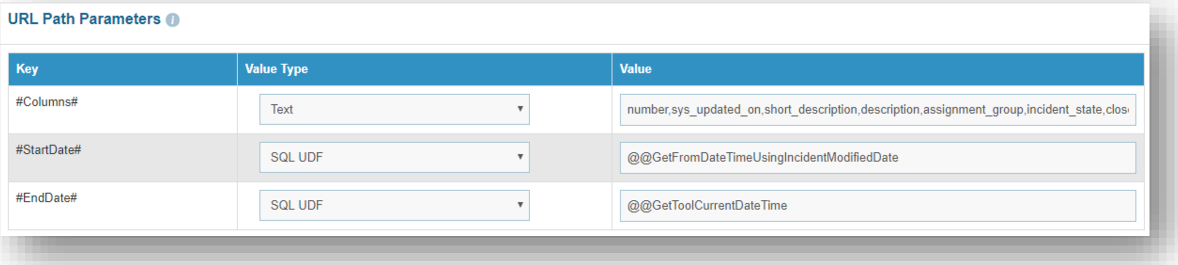
Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	Bearer	☐	☐	🗑
client_id	<clientID>	☑	☑	🗑
client_secret	<clientsrcret>	☑	☑	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	JSON	☐	☐	🗑
grant_type	Password	☐	☑	🗑

Figure 309 – Create Data Source (Request Authentication Parameters for OAuth2.0)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs:

```
Key: #Columns#

ValueType: Text

Value:

number,sys_updated_on,short_description,description,assignment_gro
up,incident_state,closed_at,category,dv_assigned_to,sys_id


Note - These columns are mandatory. User can add more columns if
more data is required to be fetched from ITSM tool.


Key: #StartDate#

ValueType: SQL UDF

VALUE: @@GetFromDateTimeUsingIncidentModifiedDate


Key: #EndDate#
```

```
ValueType: SQL UDF

VALUE: @@GetToolCurrentDateTime
```



URL Path Parameters

Key	Value Type	Value
#Columns#	Text ▾	number,sys_updated_on,short_description,description,assignment_group,incident_state,clos
#StartDate#	SQL UDF ▾	@@GetFromDateTimeUsingIncidentModifiedDate
#EndDate#	SQL UDF ▾	@@GetToolCurrentDateTime

Figure 310– URL Path Parameters

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Response Body –** In this section, please enter the output of URL query for one of the incidents in JSON format. A sample response is mentioned below.

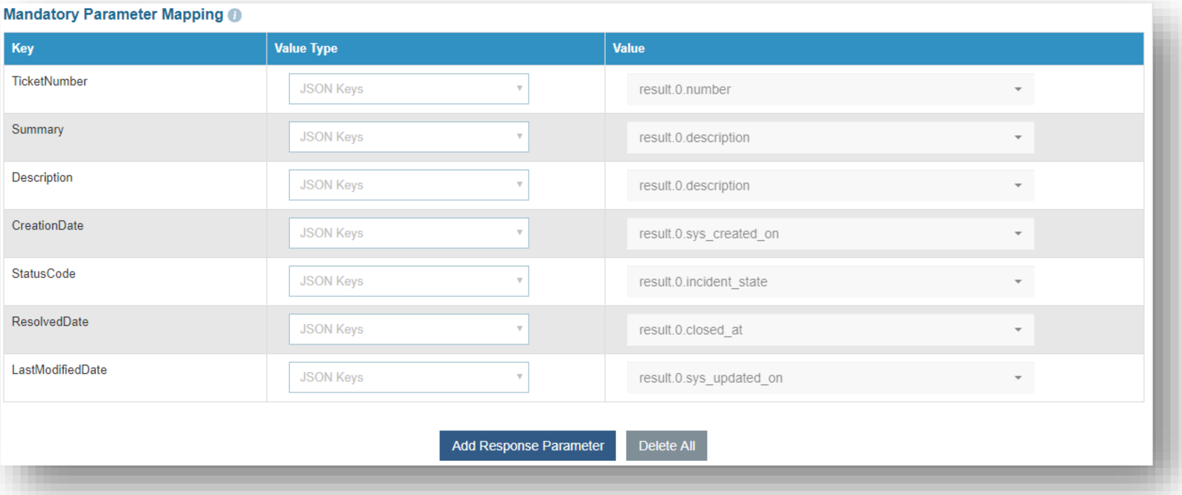
```
Response Body –

{  "result": [{   "number": "INC0079154",   "closed_at": "",
"assignment_group": {    "link": "<https://sample.service-
now.com/api/now/v1/table/sys_user_group/All user group>",
"value": "All user group"   },   "incident_state": "6",
"sys_created_on": "2017-12-22 06:59:03",   "description": "Memory
Utilization:10.0.0.11",  "short_description": "Memory
Utilization:localhost",  "sys_updated_on": "2018-01-02 06:39:56",
"category": "",   "priority": "4",   "sys_id": "123456"  }] }
```

- After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section**.**
- **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below.

Table 61– Sample Mandatory Parameter Mapping

Key	Value Type	Value
TicketNumber	JSON.Keys	result.0.number
Summary	JSON.Keys	result.0.short_description
Description	JSON.Keys	result.0.description
CreationDate	JSON.Keys	result.0.sys_created_on

StatusCode	JSON.Keys	result.0.incident_state
ResolvedDate	JSON.Keys	result.0.closed_at
LastModifiedDate	JSON.Keys	result.0.sys_updated_on



Mandatory Parameter Mapping

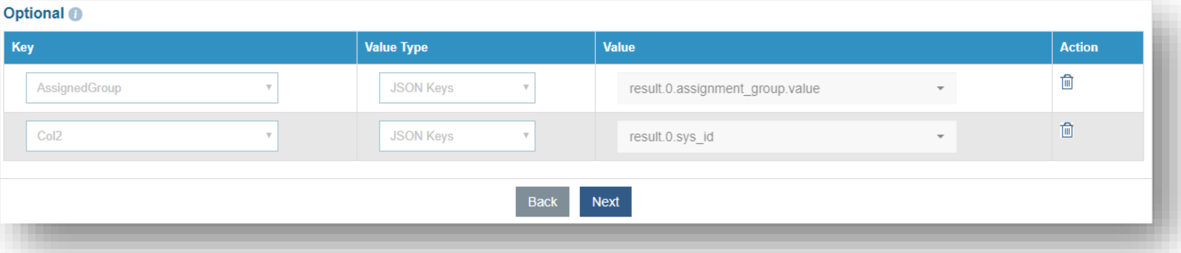
Key	Value Type	Value
TicketNumber	JSON Keys	result.0.number
Summary	JSON Keys	result.0.description
Description	JSON Keys	result.0.description
CreationDate	JSON Keys	result.0.sys_created_on
StatusCode	JSON Keys	result.0.incident_state
ResolvedDate	JSON Keys	result.0.closed_at
LastModifiedDate	JSON Keys	result.0.sys_updated_on

Figure 311 – Mandatory Parameter Mapping

– If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 62– Sample Optional Parameters

Key	Value Type	Value
AssignedGroup	JSON.Keys	result.0.assignment_group.value
Col1	JSON.Keys	result.0.sys_id



Optional

Key	Value Type	Value	Action
AssignedGroup	JSON Keys	result.0.assignment_group.value	🗑
Col2	JSON Keys	result.0.sys_id	🗑

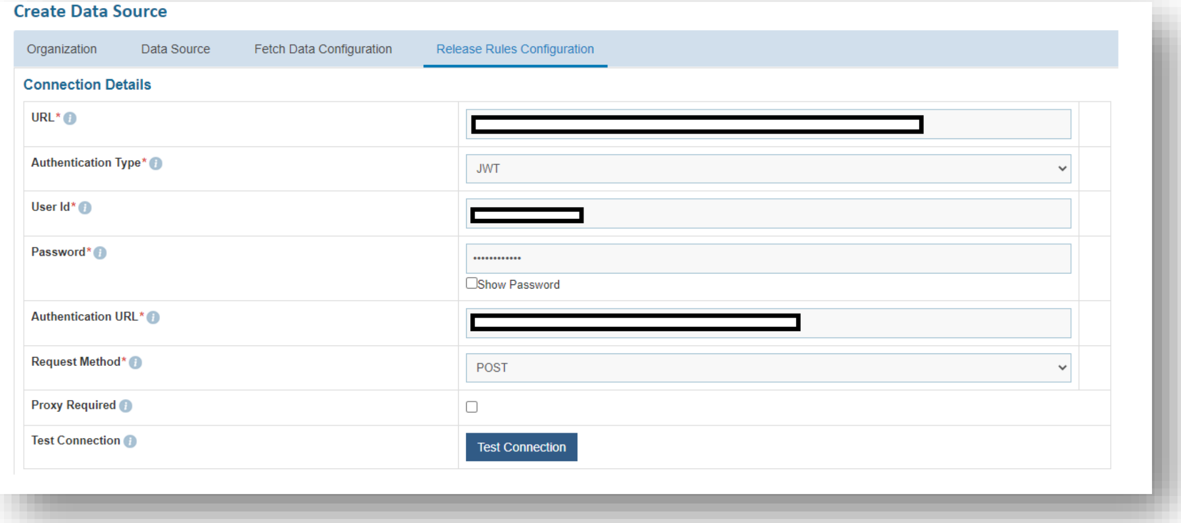
Figure 312 – Optional Parameter Mapping

– Click Next to proceed to Release Rules Configuration.

– On **Release Rules Configuration** tab, type in the details as per the requirement.

– In the **Connection Details** section, enter the following details:

- **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.
- **Sample URL** - https://<url>.service-now.com/api/now/table/incident/#incident#
- **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously.
- **Request Method** – Select Request Method as PUT from the drop-down.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



Create Data Source

Organization | Data Source | Fetch Data Configuration | Release Rules Configuration

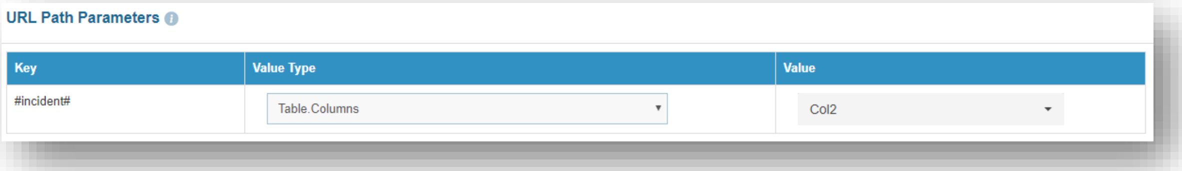
Connection Details

URL*	[redacted]
Authentication Type*	JWT
User Id*	[redacted]
Password*	············ □ Show Password
Authentication URL*	[redacted]
Request Method*	POST
Proxy Required	□
Test Connection	Test Connection

Figure 313 – Release Rules Configuration (Connection Details)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs.

```
Key: #incident#

ValueType: Table Columns

Value:

Select from dropdown that mapped to sys_id from previous screen
"Col2"
```



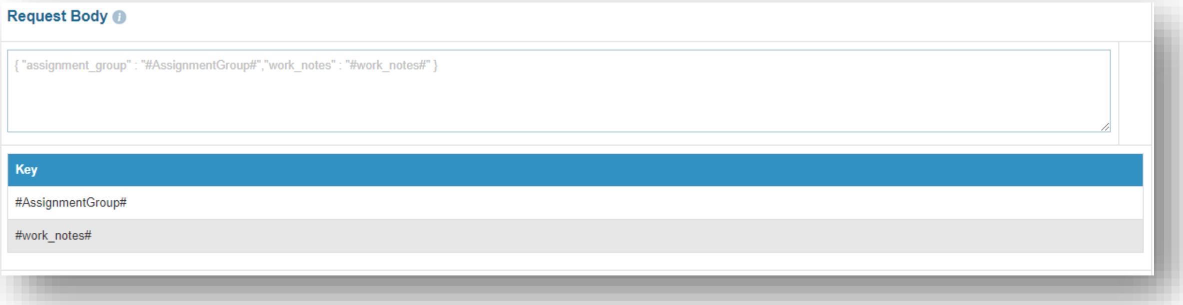
URL Path Parameters

Key	Value Type	Value
#incident#	Table.Columns	Col2

Figure 314 – Release Rules Configuration (URL Path Parameters)

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body –** In this section, please enter the request body in JSON format. A sample request is mentioned below.

```
Request Body –
{ "assignment_group" : "#AssignmentGroup#","work_notes" :
"#work_notes#" }
```



Request Body

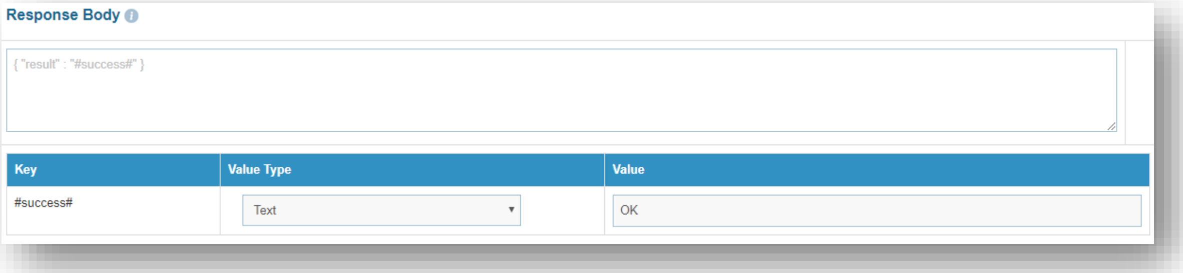
{ "assignment_group" : "#AssignmentGroup#","work_notes" : "#work_notes#" }

Key
#AssignmentGroup#
#work_notes#

Figure 315 – Release Rules Configuration (Request Body)

- **Response Body –** In this section, please enter the response body in JSON format. A sample response is mentioned below.

```
Response Body –
{ "result" : "#success#" }
```



Response Body

{ "result" : "#success#" }

Key	Value Type	Value
#success#	Text	OK

Figure 316 – Release Rules Configuration (Response Body)

– **Response Key Value** mapping can be done as per the below table.

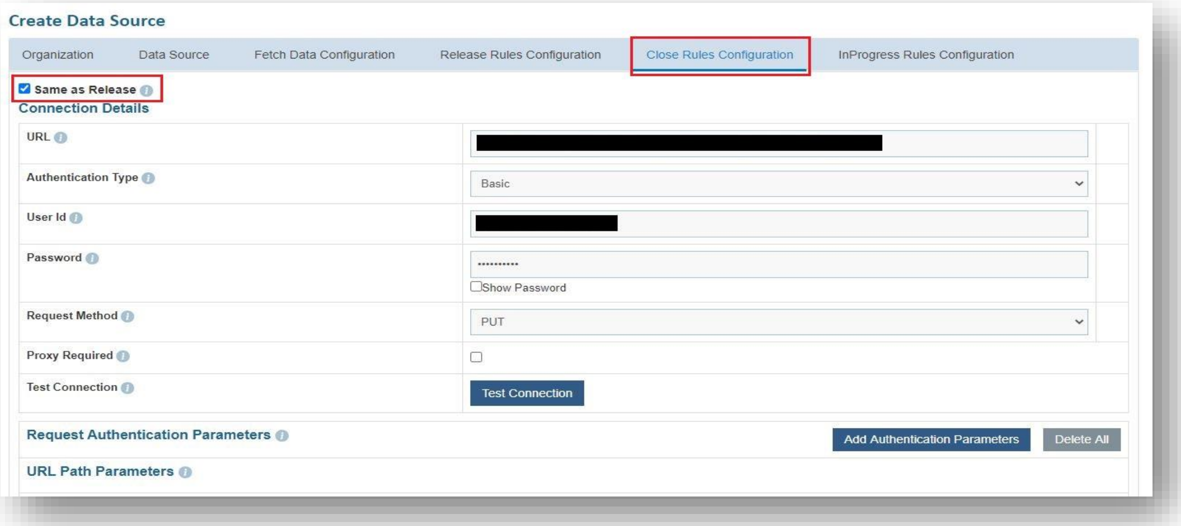
Table 63– Sample Response Key Value Mapping

#success#	Text	OK

– On **Close Rules Configuration** tab, type in the details as per the requirement. Check **Same as Release** if similar configurations as mentioned in "Release Rules Configuration" are required, else proceed ahead

– In the **Connection Details** section, enter the following details:

- **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.
- **Sample URL** - https://<url>.service-now.com/api/now/table/incident/#incident#
- **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously.
- Request Method – Select Request Method as PUT from the drop-down.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



Create Data Source

Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

☑ Same as Release

Connection Details

URL	[redacted]
Authentication Type	Basic
User Id	[redacted]
Password	•••••••••• □ Show Password
Request Method	PUT
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

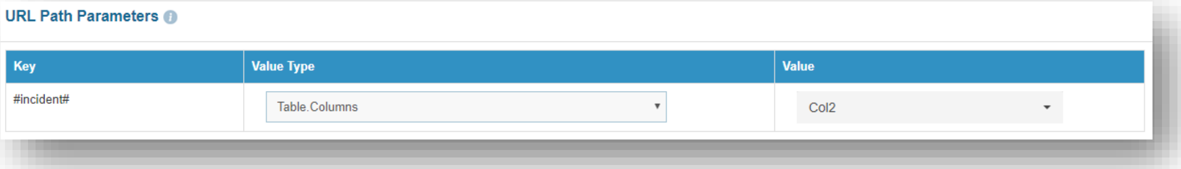
Add Authentication Parameters
Delete All

URL Path Parameters

Figure 317 – Close Rules Configuration (Connection Details)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs.

```
Key: #incident#
ValueType: Table Columns
Value:
Select from dropdown that mapped to sys_id from previous screen
"Col2"
```



URL Path Parameters

Key	Value Type	Value
#incident#	Table.Columns	Col2

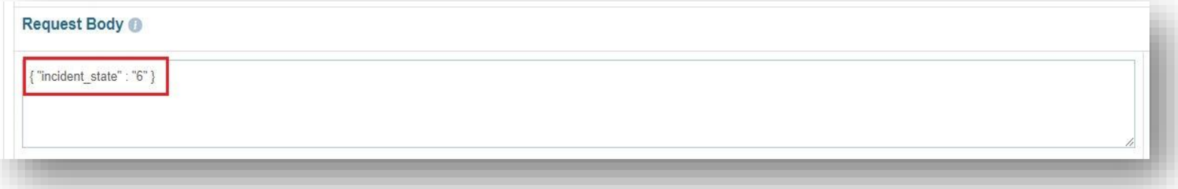
Figure 318 – Close Rules Configuration (URL Path Parameters)

– **Request Header Parameters –** Please enter the request header parameters as required.

– **Request Body –** In this section, please enter the request body in JSON format. A sample request is mentioned below.

```
Request Body –
```

```
{ "incident_state" : "6"} If you also want to add worknotes while
Close ticket, use json {"incident_state":"6", "work_notes":
"#Notes#"}
```

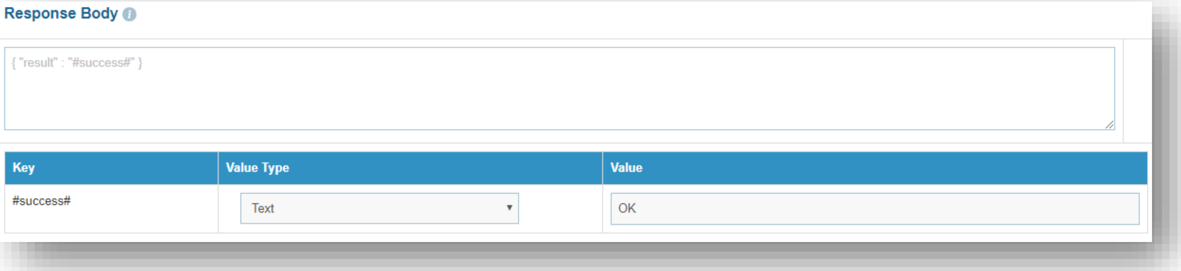

Request Body

{ "incident_state" : "6" }

Figure 319 – Close Rules Configuration (Request Body)

- **Response Body –** In this section, please enter the response body in JSON format. A sample response is mentioned below.

```
Response Body –

{ "result" : "#success#" }
```


Response Body

{ "result" : "#success#" }

Key	Value Type	Value
#success#	Text	OK

Figure 320 – Close Rules Configuration (Response Body)

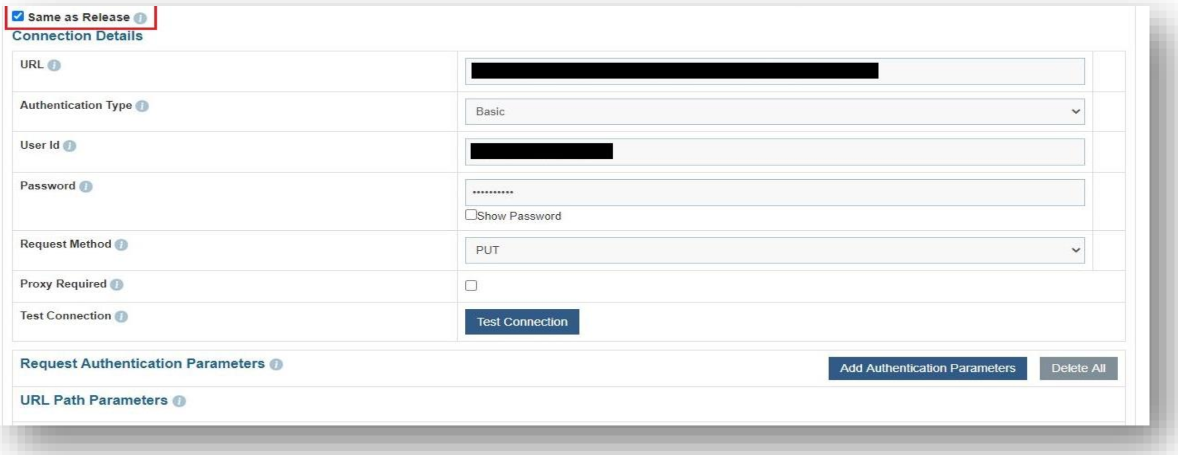
- **Response Key Value** mapping can be done as per the below table.

Table 64– Sample Response Key Value Mapping

#success#	Text	OK

- On **InProgress Rules Configuration** tab, type in the details as per the requirement. Check **Same as Release** if similar configurations as mentioned in “Release Rules Configuration” are required, else proceed ahead.
- In the **Connection Details** section, enter the following details:

- **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.
- **Sample URL** - https://<url>.service-now.com/api/now/table/incident/#incident#
- **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously.
- Request Method – Select Request Method as PUT from the drop-down.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



☑ Same as Release

Connection Details

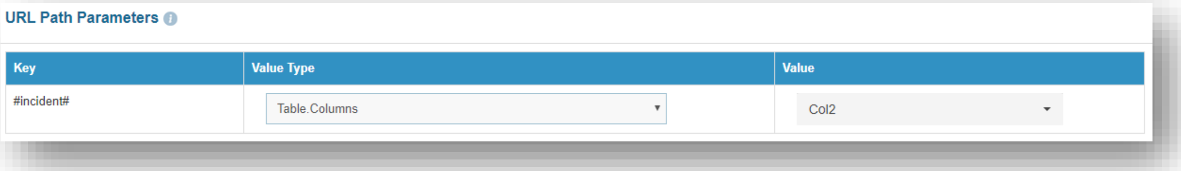
URL	[redacted]
Authentication Type	Basic
User Id	[redacted]
Password	•••••••••• □ Show Password
Request Method	PUT
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters
Add Authentication Parameters
Delete All
URL Path Parameters

Figure 321 – InProgress Rules Configuration (Connection Details)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs.

```
Key: #incident#

ValueType: Table Columns

Value:

Select from dropdown that mapped to sys_id from previous screen
"Col2"
```



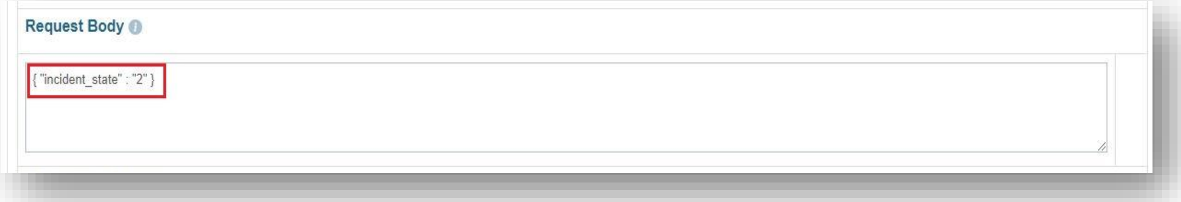
URL Path Parameters

Key	Value Type	Value
#incident#	Table.Columns	Col2

Figure 322 – InProgress Rules Configuration (URL Path Parameters)

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body –** In this section, please enter the request body in JSON format. A sample request is mentioned below.

```
Request Body –

{"incident_state" : "2"} If you also want to add worknotes while
inprogress ticket, use json {"incident_state":"2", "work_notes":
"#Notes#"}
```



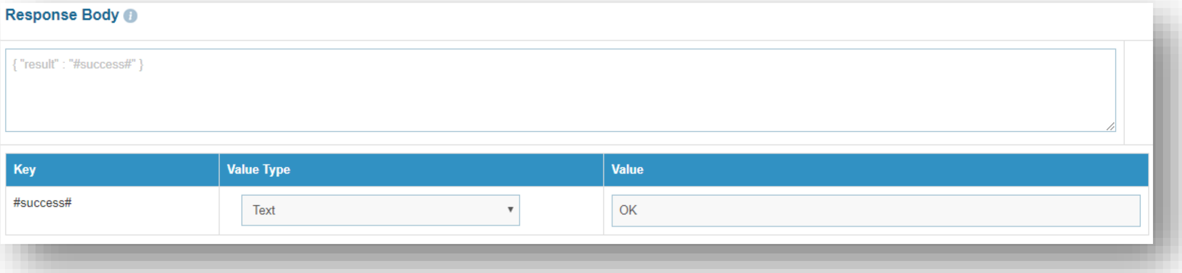
Request Body

```
{ "incident_state" : "2" }
```

Figure 323 – InProgress Rules Configuration (Request Body)

- **Response Body –** In this section, please enter the response body in JSON format. A sample response is mentioned below.

```
Response Body –

{ "result" : "#success#" }
```



Response Body

{ "result" : "#success#" }

Key	Value Type	Value
#success#	Text	OK

Figure 324 – InProgress Rules Configuration (Response Body)

– **Response Key Value** mapping can be done as per the below table.

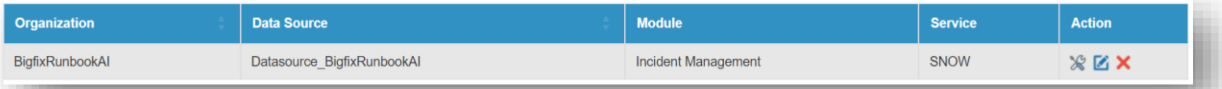
Table 65– Sample Response Key Value Mapping

#success#	Text	OK

– Click **Submit** to add the data source.

– In order to bring the tickets within BigFix Runbook AI scope, a specific queue needs to be configured in the ITSM tool and same has to be configured in BigFix Runbook AI. This is achieved through **Manage the Entry Criteria**. Please perform the below steps –

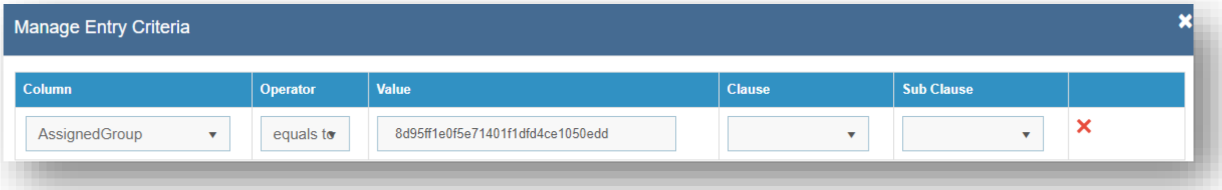
- Go to Actions tab and click Manage Data Sources.
- On the **Data Sources** tab, click ⚒ next to the data source user wants to manage. **Manage Entry Criteria** screen appears.



Organization	Data Source	Module	Service	Action
BigfixRunbookAI	Datasource_BigfixRunbookAI	Incident Management	SNOW	⚒ ✎ ✕

Figure 325 – Manage Entry Criteria

- Select 'AssignedGroup' for the **Column** field and 'equals to' for the **Operator**.
- Enter the sys_id of the assignment group in ServiceNow in the **Value** field.
- **Clause** and **Sub-Clause** fields can also be added based on requirement.



Manage Entry Criteria

Column	Operator	Value	Clause	Sub Clause	
AssignedGroup	equals to	8d95ff1e0f5e71401f1dfd4ce1050edd			✕

Figure 326 – Manage Entry Criteria (cont.)

- Click **Save**.

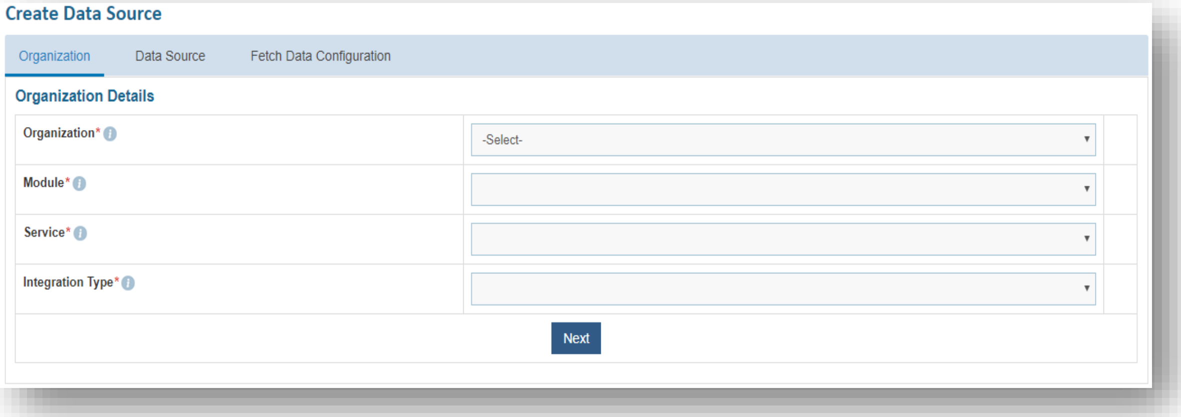
4.8.2 Incident Management without ITSM (ServiceNow)

This scenario is applicable when the ITSM tools is not available in the client environment and event management tool and BigFix Runbook AI are tightly integrated directly. The event data or the probable root cause identified flows to BigFix Runbook AI which then performs the appropriate action for resolution.

The user has the option to view the events and trigger the resolutions via Moogsoft as well as BigFix Runbook AI console.

To create a data source, perform the following steps:

- On the main menu bar, click **Actions Tab → Manage Data Source**.
- The **Create Data Source** page appears with the following tabs:
  - Organization
  - Data Source
  - Fetch Data Configuration
  - Release Rules Configuration
  - Close Rules Configuration (Optional – applicable only when the issue closure status update is managed by BigFix Runbook AI directly instead of RBA tool)
  - InProgress Rules Configuration (Optional – applicable only when the issue's in progress status updates is managed by BigFix Runbook AI directly instead of RBA tool)



Create Data Source

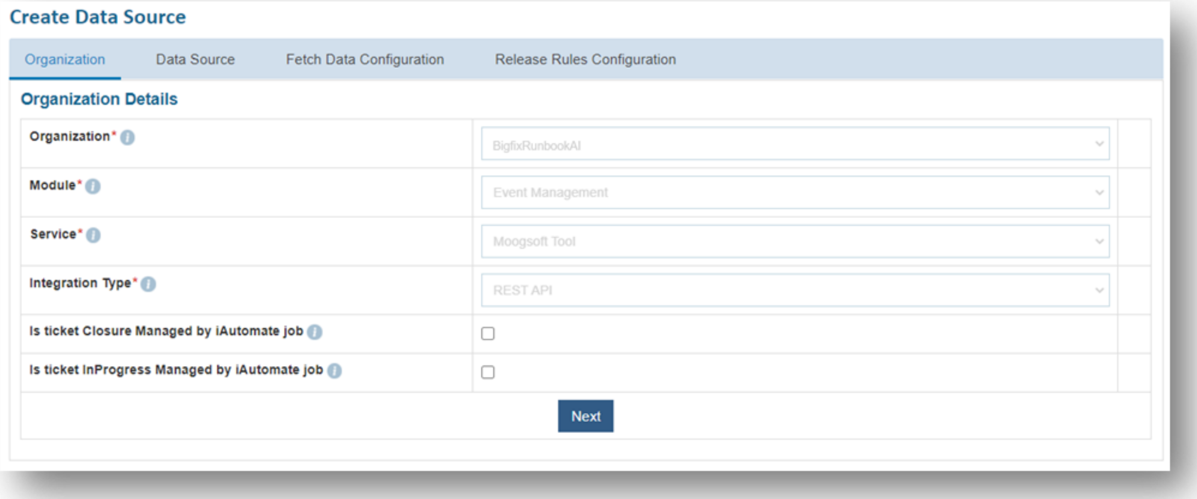
Organization Details

Organization*	-Select-
Module*	
Service*	
Integration Type*	

Figure 327 – Create Data Source

Release Rules Configuration is only applicable for the following **Module** types- **Incident Management, Change Request Task and Service Request Task.** This tab will not be activated for other module types.

- On the **Organization** tab,
  - Select the **Organization Name** from the dropdown.
  - Select the **Module** as **Event Management,** since we are configuring this data source for pulling the event data.
  - Select the **Service** as **Moogsoft Tool** as we are configuring the data source for Moogsoft
  - Select the **Integration Type** as **REST**, since we will be integrating through REST APIs.
  - Check **Is ticket Closure Managed by BigFix Runbook AI job** if you want BigFix Runbook AI to manage the ticket closure updates instead of the RBA tool. In this scenario, an additional tab **Close Rules Configuration** will be activated for providing further details, steps for which are mentioned later.
  - Check "**Is ticket InProgress Managed by BigFix Runbook AI job**" if you want BigFix Runbook AI to manage the ticket's in progress status updates instead of the RBA tool. In this scenario, an additional tab "**InProgress Rules Configuration**" will be activated for providing further details, steps for which are mentioned later.
  - Click **Next**.



Create Data Source

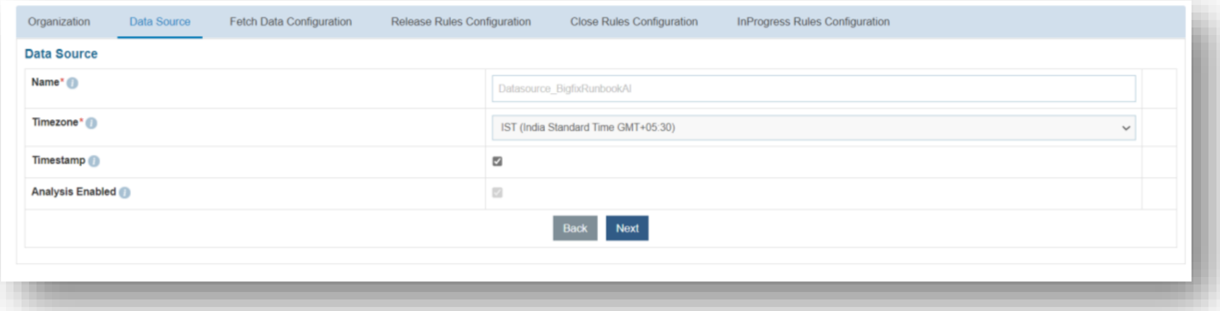
Organization Details

Organization*	BigfixRunbookAI
Module*	Event Management
Service*	Moogsoft Tool
Integration Type*	REST API
Is ticket Closure Managed by iAutomate job	□
Is ticket InProgress Managed by iAutomate job	□

Figure 328 – Create Data Source (cont.)

– On the **Data Source** tab,

- Type the new data source in the **Name** field.
- Select the **Timezone** to specify the time zone of the selected data source.
- Select **Timestamp** to view the present data with date and time.
- Select **Analysis Enabled** if user wants to analyze the data retrieved from the data source.
- Click **Next**.



Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

Data Source

Name*	Datasource_BigfixRunbookAI
Timezone*	IST (India Standard Time GMT+05:30)
Timestamp	☑
Analysis Enabled	☑

Figure 329 – Create Data Source (cont.)

– On the **Fetch Data Configuration** tab, type in the details as per the environment.

– In the **Connection Details** section, enter the following details:

- **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.

- **Sample URL** - http://<IP>:<PORT>/iAutomateAPI/Request/GetIncidentTicketData/11?start_date>=#startdate#&end_date<=#enddate#
- **Authentication Type** – Select one of the Authentication Types from Basic / Windows, JWT, OAuth 2.0

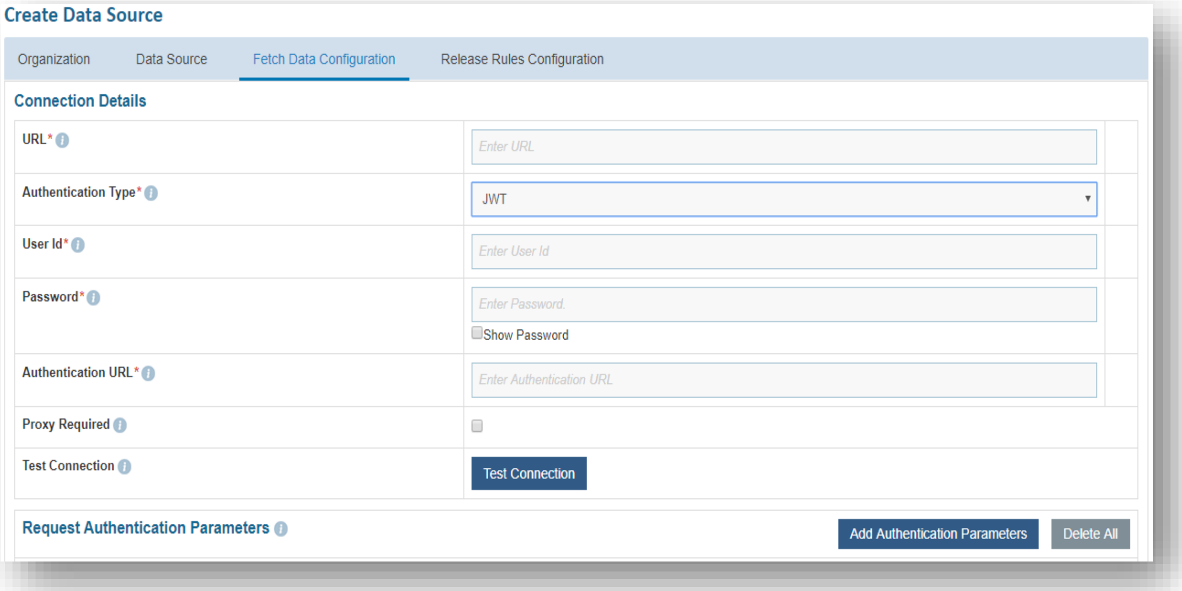
  Selection of **Basic / Windows** requires you to enter -

  - User Id
  - Password

  Selection of **JWT / OAuth 2.0** requires you to enter -

  - User Id
  - Password
  - Authentication URL

- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



Create Data Source

Connection Details

URL*	Enter URL
Authentication Type*	JWT
User Id*	Enter User Id
Password*	Enter Password. □ Show Password
Authentication URL*	Enter Authentication URL
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

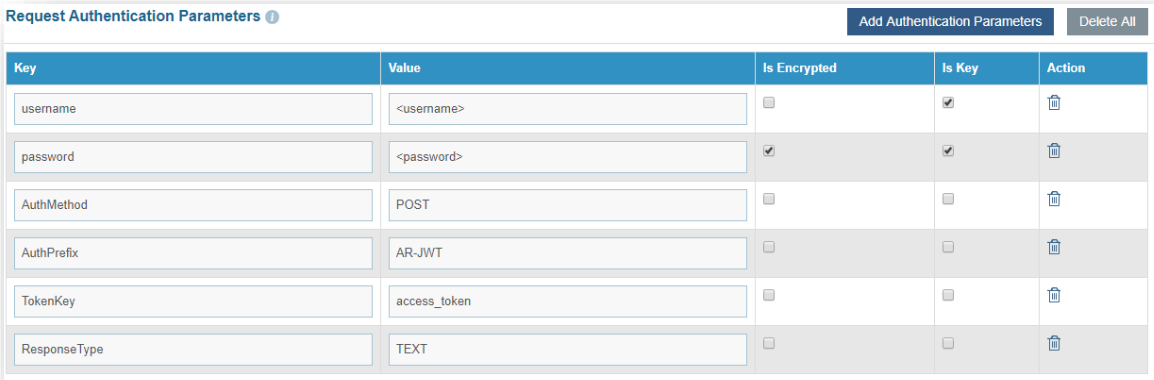
Figure 330 – Create Data Source (Connection Details)

– **Request Authentication Parameters** – If the user has additional parameters, click Add Authentication Parameters under the Request Authentication Parameters tab.

– Based on the **Authentication Type**, add the parameters mentioned in the below table.

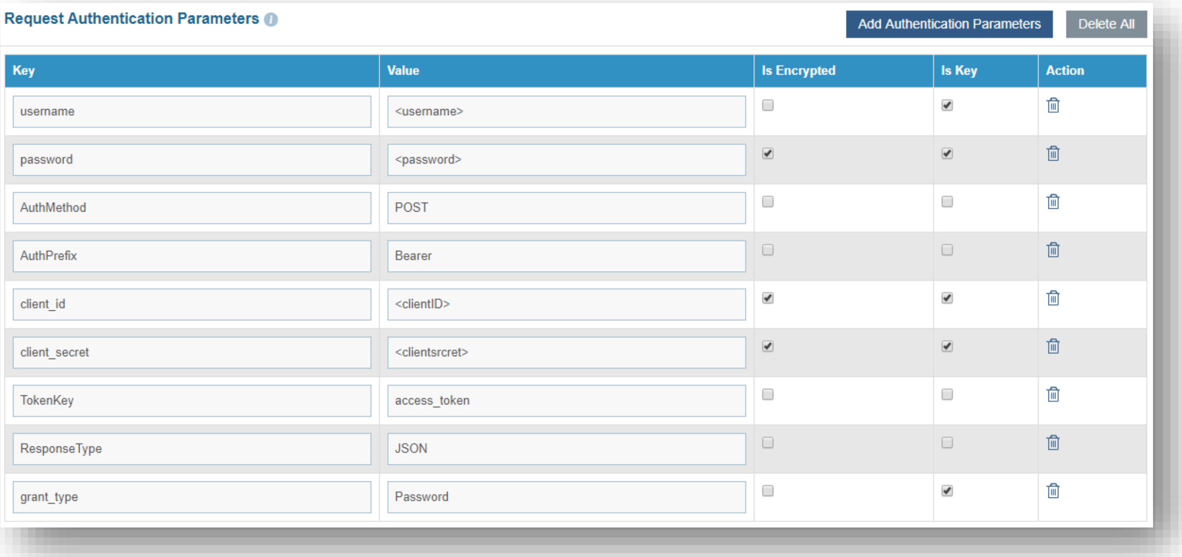
Table 66– Sample Authentication Parameters

Authentication Type	Key	Value	Is Encrypted?	Is Key?
JWT	username	<username>	NO	YES
JWT	password	<password>	YES	YES
JWT	AuthMethod	POST	NO	NO
JWT	AuthPrefix	AR-JWT	NO	NO
JWT	TokenKey	access_token	NO	NO
JWT	ResponseType	TEXT	NO	NO
OAuth2.0	username	<username	NO	YES
OAuth2.0	password	<password>	YES	YES
OAuth2.0	AuthMethod	POST	NO	NO
OAuth2.0	AuthPrefix	Bearer	NO	NO
OAuth2.0	client_id	<clientID>	YES	YES
OAuth2.0	client_secret	<clientsecret>	YES	YES
OAuth2.0	TokenKey	access_token	NO	NO
OAuth2.0	ResponseType	JSON	NO	NO
OAuth2.0	grant_type	Password	NO	YES


Request Authentication Parameters
Add Authentication Parameters
Delete All

Key	Value	Is Encrypted	Is Key	Action
username	<username>	☐	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	☐	☐	🗑
AuthPrefix	AR-JWT	☐	☐	🗑
TokenKey	access_token	☐	☐	🗑
ResponseType	TEXT	☐	☐	🗑

Figure 331 – Create Data Source (Request Authentication Parameters for JWT)



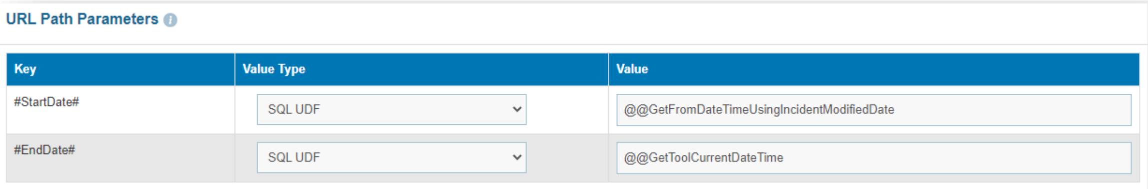
Request Authentication Parameters

Key	Value	Is Encrypted	Is Key	Action
username	<username>	□	☑	🗑
password	<password>	☑	☑	🗑
AuthMethod	POST	□	□	🗑
AuthPrefix	Bearer	□	□	🗑
client_id	<clientID>	☑	☑	🗑
client_secret	<clientsrcret>	☑	☑	🗑
TokenKey	access_token	□	□	🗑
ResponseType	JSON	□	□	🗑
grant_type	Password	□	☑	🗑

Figure 332 – Create Data Source (Request Authentication Parameters for OAuth2.0)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs.

```
Key: #StartDate#
ValueType: SQL UDF
VALUE: @@GetFromDateTimeUsingIncidentModifiedDate


Key: #EndDate#
ValueType: SQL UDF
VALUE: @@GetToolCurrentDateTime
```



URL Path Parameters

Key	Value Type	Value
#StartDate#	SQL UDF	@@GetFromDateTimeUsingIncidentModifiedDate
#EndDate#	SQL UDF	@@GetToolCurrentDateTime

Figure 333– URL Path Parameters

– **Request Header Parameters –** Please enter the request header parameters as required.

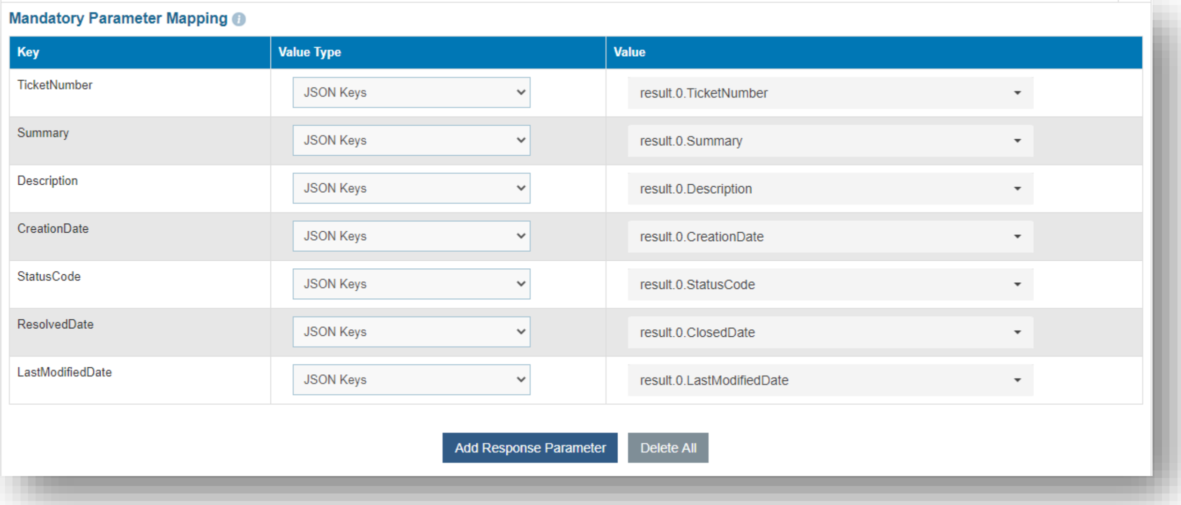
- **Response Body –** In this section, please enter the output of URL query for one of the incidents in JSON format. A sample response is mentioned below.

```
Response Body –

{ "result": [ { "TicketNumber": "1006976", "Summary": "Restart
Spooler service on target server ", "Description": "Restart
Spooler service on target server", "AssignedGroup":
"945e4f5b7ba0108fd5bfce83af21369", "StatusCode": "1",
"CreationDate": "2020-05-04 10:40:30.000", "LastModifiedDate":
"2020-05-04 04:41:50.000", "ClosedDate": "2020-05-06
10:41:53.000", "sys_id": "945e9006d4b89a98fe7574c1cc284", "Col1":
"", "Col2": "", "Col3": "", "Col4": "", "Col5": "",
"iAutomate_CreatedDateInGMT": "2020-05-04 05:25:36.350",
"iAutomate_UpdatedDateInGMT": "2020-05-04 05:25:36.350" } ] }
```

- After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section.
- **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below.

Table 67– Sample Mandatory Parameter Mapping

Key	Value Type	Value
TicketNumber	JSON.Keys	result.0.TicketNumber
Summary	JSON.Keys	result.0.Summary
Description	JSON.Keys	result.0.Description
CreationDate	JSON.Keys	result.0.CreationDate
StatusCode	JSON.Keys	result.0.StatusCode
ResolvedDate	JSON.Keys	result.0.ClosedDate
LastModifiedDate	JSON.Keys	result.0.LastModifiedDate



Mandatory Parameter Mapping

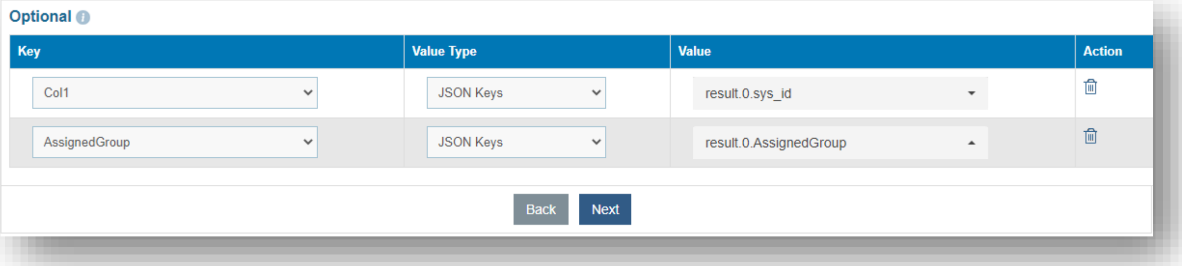
Key	Value Type	Value
TicketNumber	JSON Keys	result.0.TicketNumber
Summary	JSON Keys	result.0.Summary
Description	JSON Keys	result.0.Description
CreationDate	JSON Keys	result.0.CreationDate
StatusCode	JSON Keys	result.0.StatusCode
ResolvedDate	JSON Keys	result.0.ClosedDate
LastModifiedDate	JSON Keys	result.0.LastModifiedDate

Figure 334 – Mandatory Parameter Mapping

- If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 68– Sample Optional Parameters

Key	Value Type	Value
AssignedGroup	JSON.Keys	result.0. AssignedGroup
Col1	JSON.Keys	result.0.sys_id



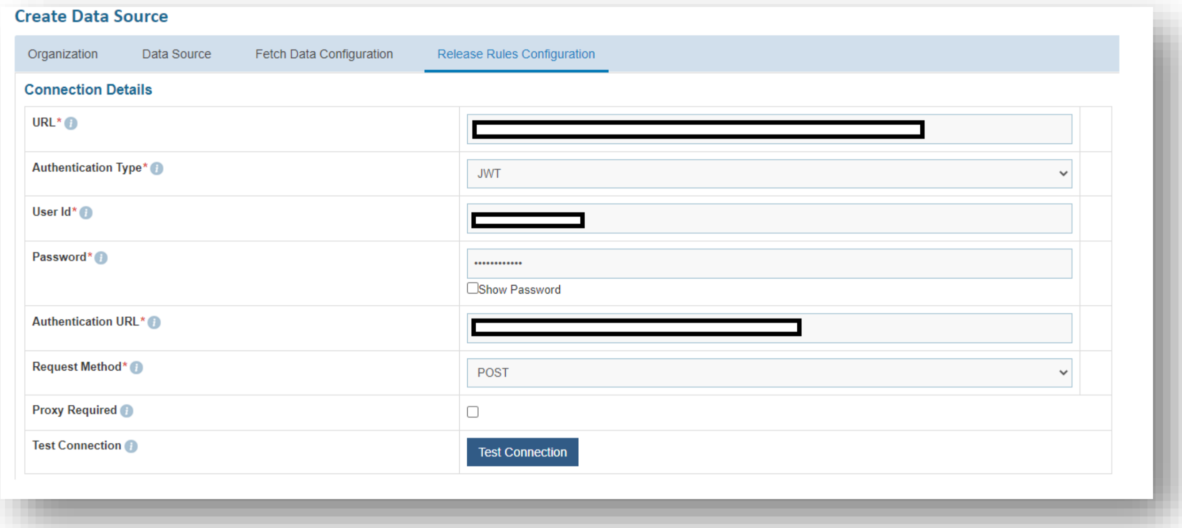
Optional

Key	Value Type	Value	Action
Col1	JSON Keys	result.0.sys_id	🗑
AssignedGroup	JSON Keys	result.0.AssignedGroup	🗑

Figure 335 – Optional Parameter Mapping

- Click Next to proceed to Release Rules Configuration.
- On **Release Rules Configuration** tab, type in the details as per the requirement.
- In the **Connection Details** section, enter the following details:

- **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.
- **Sample URL** - https://<URL>/graze/v1/#value#
- **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously.
- **Request Method** – Select Request Method as PUT from the drop-down.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



Create Data Source

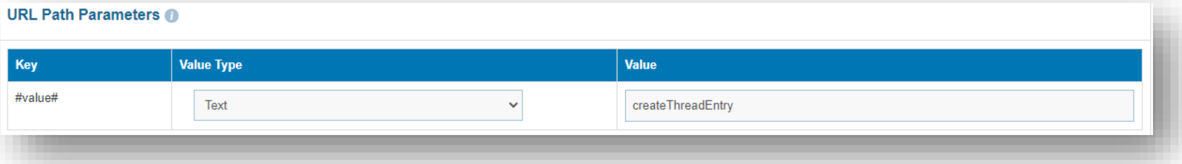
Connection Details

URL*	[redacted]
Authentication Type*	JWT
User Id*	[redacted]
Password*	•••••••••••• □Show Password
Authentication URL*	[redacted]
Request Method*	POST
Proxy Required	□
Test Connection	Test Connection

Figure 336 – Release Rules Configuration (Connection Details)

– **URL Path Parameters –** Based on the URL entered earlier, please map the values to the URL Path Parameters. E.g., for the URL entered earlier, please populate the below inputs.

```
Key: #value#

ValueType: Text

Value: createThreadEntry
```



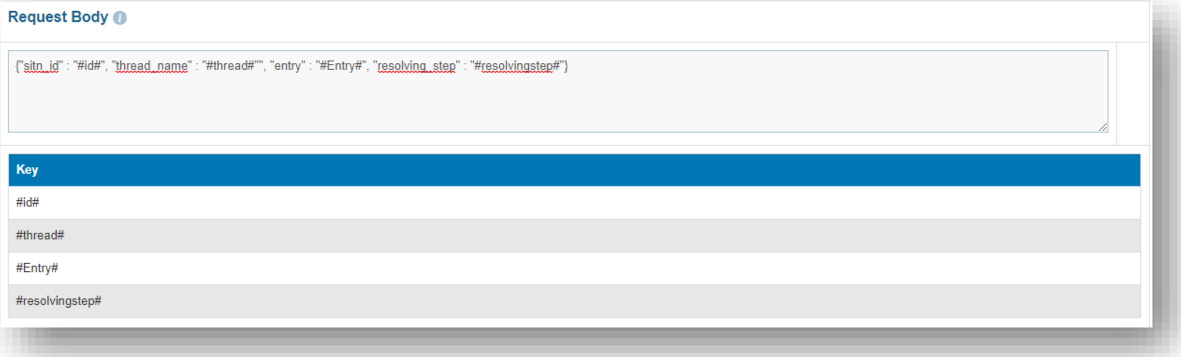
URL Path Parameters

Key	Value Type	Value
#value#	Text	createThreadEntry

Figure 337 – Release Rules Configuration (URL Path Parameters)

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body –** In this section, please enter the request body in JSON format. A sample request is mentioned below.

```
Request Body –

{"sitn_id" : "#id#", "thread_name" : "#thread#"", "entry" :
"#Entry#", "resolving_step" : "#resolvingstep#"}
```



Request Body

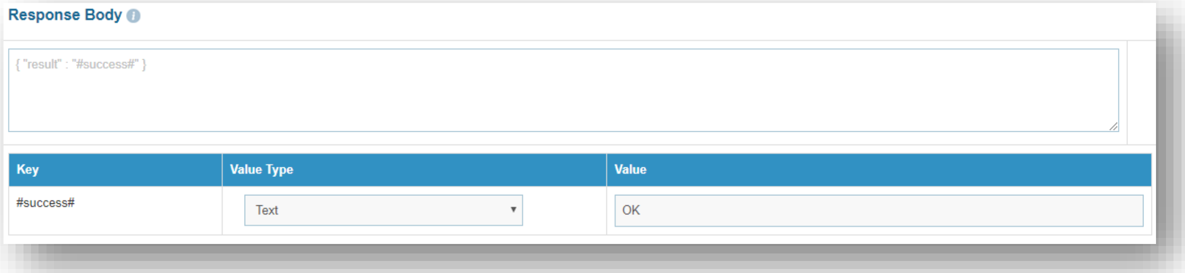
```
{"sitn_id" : "#id#", "thread_name" : "#thread#"", "entry" : "#Entry#", "resolving_step" : "#resolvingstep#"}
```

Key
#id#
#thread#
#Entry#
#resolvingstep#

Figure 338 – Release Rules Configuration (Request Body)

- **Response Body –** In this section, please enter the response body in JSON format. A sample response is mentioned below.

```
Response Body –

{"result":"#success#"}
```



Response Body

{ "result" : "#success#" }

Key	Value Type	Value
#success#	Text	OK

Figure 339 – Release Rules Configuration (Response Body)

– **Response Key Value** mapping can be done as per the below table.

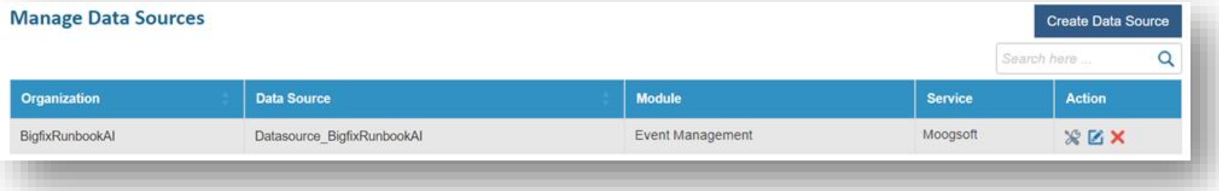
Table 69– Sample Response Key Value Mapping

#success#	Text	OK

– Click **Submit** to add the data source.

– In order to bring the tickets within BigFix Runbook AI scope, a specific queue needs to be configured in the Event Management tool and same has to be configured in BigFix Runbook AI. This is achieved through **Manage the Entry Criteria**. Please perform the below steps:

- Go to Actions tab and click Manage Data Sources.
- On the **Data Sources** tab, click ⚒ next to the data source user wants to manage. **Manage Entry Criteria** screen appears.

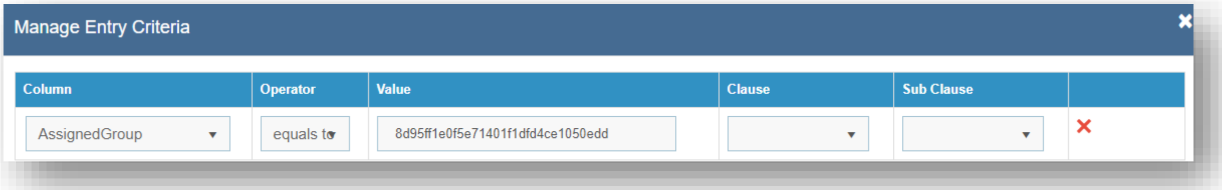


Manage Data Sources

Organization	Data Source	Module	Service	Action
BigfixRunbookAI	Datasource_BigfixRunbookAI	Event Management	Moogsoft	⚒ ✎ ✕

Figure 340 – Manage Entry Criteria

- Select ‘AssignedGroup’ for the **Column** field and ‘equals to’ for the **Operator** field.
- Enter the sys_id of the assignment group in Moogsoft in the **Value** field.
- **Clause** and **Sub-Clause** fields can also be added based on requirement.



Manage Entry Criteria

Column	Operator	Value	Clause	Sub Clause	
AssignedGroup	equals to	8d95ff1e0f5e71401f1dfd4ce1050edd			✕

Figure 341 – Manage Entry Criteria (cont.)

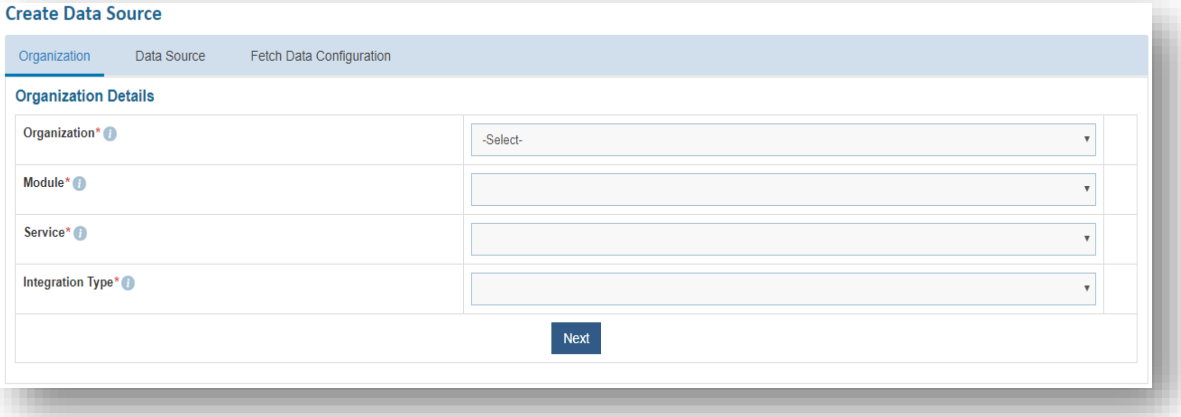
- Click **Save**.

4.9 Integration with Zenoss

This scenario is applicable when the ITSM tools is not available in the client environment and event management tool and BigFix Runbook AI are tightly integrated directly. The event data or the probable root cause identified flows to BigFix Runbook AI which then performs the appropriate action for resolution.

To create a data source, perform the following steps:

- On the main menu bar, click **Actions tab → Manage Data Source**.
- The **Create Data Source** page appears with the following tabs:
  - Organization
  - Data Source
  - Fetch Data Configuration
  - Release Rules Configuration
  - Close Rules Configuration (Optional – applicable only when the issue closure status update is managed by BigFix Runbook AI directly instead of RBA tool)
  - InProgress Rules Configuration (Optional – applicable only when the issue's in progress status updates is managed by BigFix Runbook AI directly instead of RBA tool)



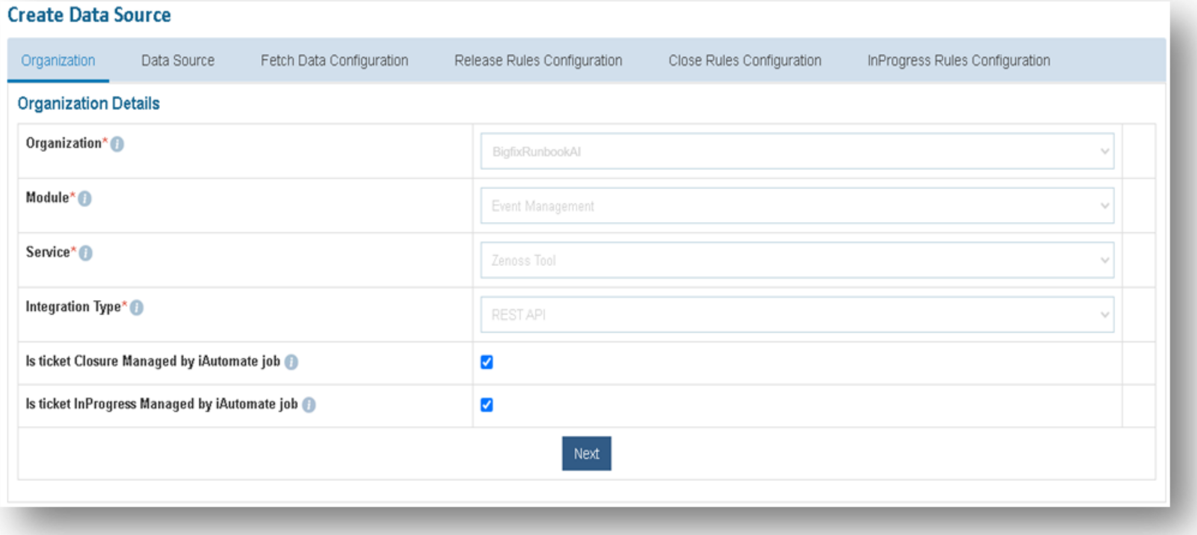
Create Data Source

Organization Details

Organization*	-Select-
Module*	
Service*	
Integration Type*	

Figure 342 – Create Data Source

- On the **Organization** tab,
  - Select the **Organization Name** from the dropdown.
  - Select the **Module** as **Event Management,** since we are configuring this data source for pulling the event data.
  - Select the **Service** as **Zenoss Tool** as we are configuring the data source for Zenoss
  - Select the **Integration Type** as **REST**, since we will be integrating through REST APIs.
  - Check **Is ticket Closure Managed by BigFix Runbook AI job** if you want BigFix Runbook AI to manage the issue closure updates instead of the RBA tool. In this scenario, an additional tab **Close Rules Configuration** will be activated for providing further details, steps for which are mentioned later.
  - Check "**Is ticket InProgress Managed by BigFix Runbook AI job**" if you want BigFix Runbook AI to manage the issue's in progress status updates instead of the RBA tool. In this scenario, an additional tab "**InProgress Rules Configuration**" will be activated for providing further details, steps for which are mentioned later.
  - Click **Next**.



Create Data Source

Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

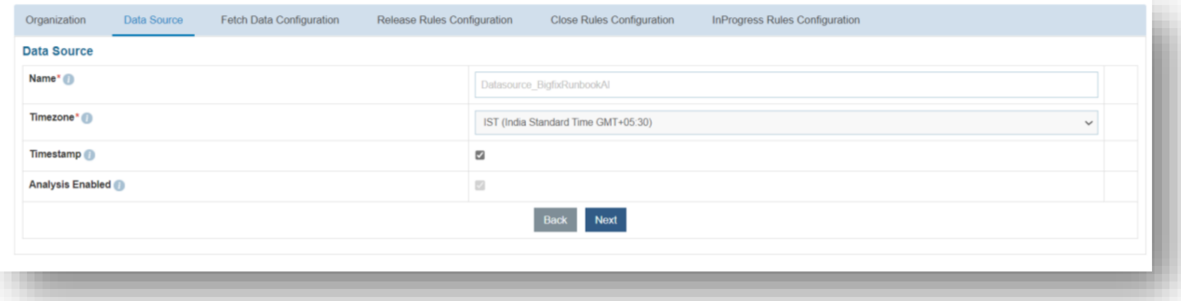
Organization Details

Organization*	BigfixRunbookAI
Module*	Event Management
Service*	Zenoss Tool
Integration Type*	REST API
Is ticket Closure Managed by iAutomate job	☑
Is ticket InProgress Managed by iAutomate job	☑

Figure 343 – Create Data Source (cont.)

– On the **Data Source** tab:

- Type the new data source in the **Name** field.
- Select the **Timezone** to specify the time zone of the selected data source.
- Select **Timestamp** to view the present data with date and time.
- Select **Analysis Enabled** if user wants to analyze the data retrieved from the data source.
- Click Next.



Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

Data Source

Name*	Datasource_BigfixRunbookAI
Timezone*	IST (India Standard Time GMT+05:30)
Timestamp	☑
Analysis Enabled	☑

Figure 344 – Create Data Source (cont.)

– On the **Fetch Data Configuration** tab, type in the details as per the environment.

– In the **Connection Details** section, enter the following details:

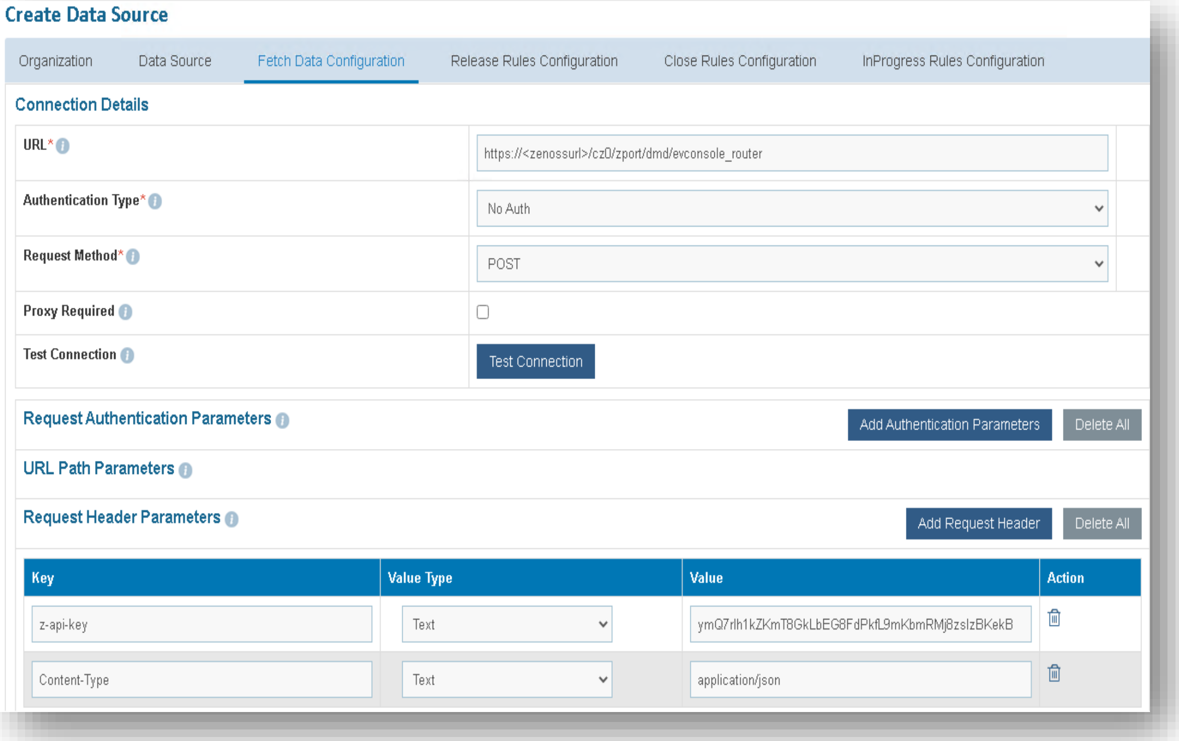
- **URL –** Type the URL which contains the placeholders that display the parameters based on the applied clause such as the number of records to be fetched, query type, date on which the data is fetched, and the order by and so on. It is dependent on the URL or API provided by the tool.

- **Sample URL** - https://<zenossURL>/cz0/zport/dmd/evconsole_router
- **Authentication Type** - Select one of the Authentication Types from NoAuth / Basic / Windows

  Selection of **Basic / Windows** requires you to enter -

  - User Id
  - Password

- **Request Method -** Select Request Method as **POST** from the drop-down.
- **Proxy Required -** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



Connection Details

URL*	https://<zenossurl>/cz0/zport/dmd/evconsole_router
Authentication Type*	No Auth
Request Method*	POST
Proxy Required	□
Test Connection	Test Connection

Request Authentication Parameters

URL Path Parameters

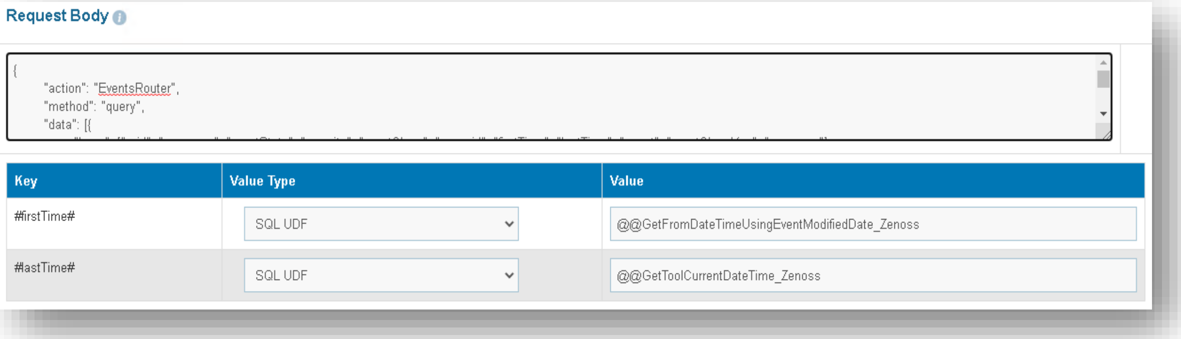
Request Header Parameters

Key	Value Type	Value	Action
z-api-key	Text	ymQ7rlh1kZKmT8GkLbEG8FdPkfL9mKbmRMj8zslzBKekB	🗑
Content-Type	Text	application/json	🗑

Figure 345 – Create Data Source (Connection Details)

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body -** As request method selected earlier is **POST**, please enter the body of URL. A sample response is mentioned below.

```
Request Body –

{"action": "EventsRouter","method": "query",

"data": [{

"keys": ["evid", "summary", "eventState", "severity",
"eventClass", "ownerid", "firstTime", "lastTime", "count",
"eventClassKey", "message"],

"params": {

"eventState": [0, 1], "severity": [5],

"excludeNonActionables": false,

"firstTime": "#firstTime# TO #lastTime#","eventClass": []},

"limit": 200,

"sort": "firstTime",

"dir": "ASC",

"start": 0,

"uid": "/cz0/zport/dmd"

}],

"type": "rpc",

"tid": 2

}
```



Request Body

```
{
        "action": "EventsRouter",
        "method": "query",
        "data": [{
```

Key	Value Type	Value
#firstTime#	SQL UDF	@@GetFromDateTimeUsingEventModifiedDate_Zenoss
#astTime#	SQL UDF	@@GetToolCurrentDateTime_Zenoss

Figure 346 – Create Data Source (Connection Details)

- **Response Body –** In this section, please enter the output of URL query for one of the incidents in JSON format. A sample response is mentioned below.

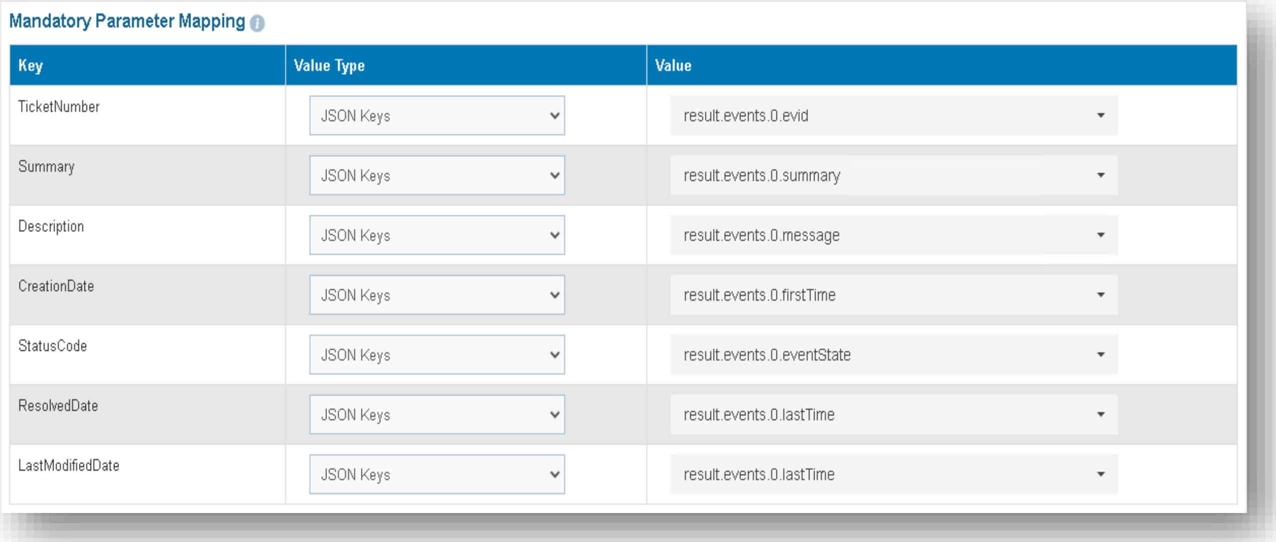
```
Response Body –
{
"result": {
        "totalCount": 1,
        "events": [
            {
                "count": 1,
                "firstTime": 1600874287.072,
                "severity": 5,
                "evid": "0242ac11-000c-b913-11ea-fdaffba5ea6f",
                "eventClassKey": "",
                "summary": "10.1.140.244 | manageIP:
10.1.140.244",
                "eventState": "New",
                "ownerid": null,
                "eventClass": {
                    "text": "/App",
                    "uid": "/zport/dmd/Events/App"
                },
                "lastTime": 1600874287.072,
                "message": "10.1.140.244"
            }
        ],
        "success": true,
        "asof": 1601266658.118566
        }
}
```

– After entering the response, click **Extract Keys** to add the parameters in the **Mandatory Parameter Mapping** section**.**

– **Mandatory Parameter Mapping –** Please map the mandatory parameters to the respective values as mentioned in the screenshot below:

Table 70– Sample Mandatory Parameter Mapping

Key	Value Type	Value
TicketNumber	JSON.Keys	result.0.evid
Summary	JSON.Keys	result.events.0.summary
Description	JSON.Keys	result.events.0.message
CreationDate	JSON.Keys	result.events.0.firstTime
StatusCode	JSON.Keys	result.events.0.eventState
ResolvedDate	JSON.Keys	result.events.0.lastTime
LastModifiedDate	JSON.Keys	result.events.0.lastTime



Mandatory Parameter Mapping

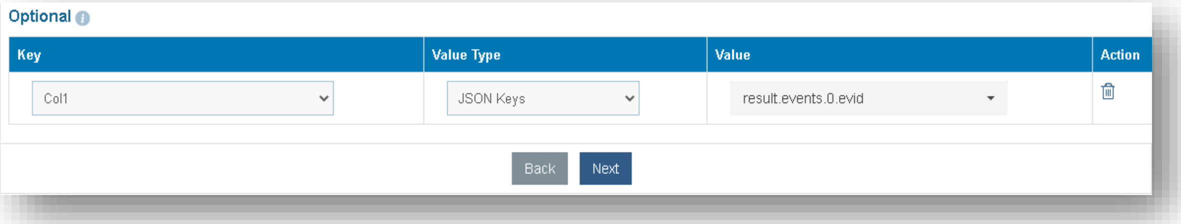
Key	Value Type	Value
TicketNumber	JSON Keys	result.events.0.evid
Summary	JSON Keys	result.events.0.summary
Description	JSON Keys	result.events.0.message
CreationDate	JSON Keys	result.events.0.firstTime
StatusCode	JSON Keys	result.events.0.eventState
ResolvedDate	JSON Keys	result.events.0.lastTime
LastModifiedDate	JSON Keys	result.events.0.lastTime

Figure 347 – Mandatory Parameter Mapping

– If you need to add **Optional** parameters, click **Add Response Parameter** to add more parameters. For our purpose, we will be adding a couple of extra parameters, as mentioned below, as we need them in the later section.

Table 71– Sample Optional Parameters

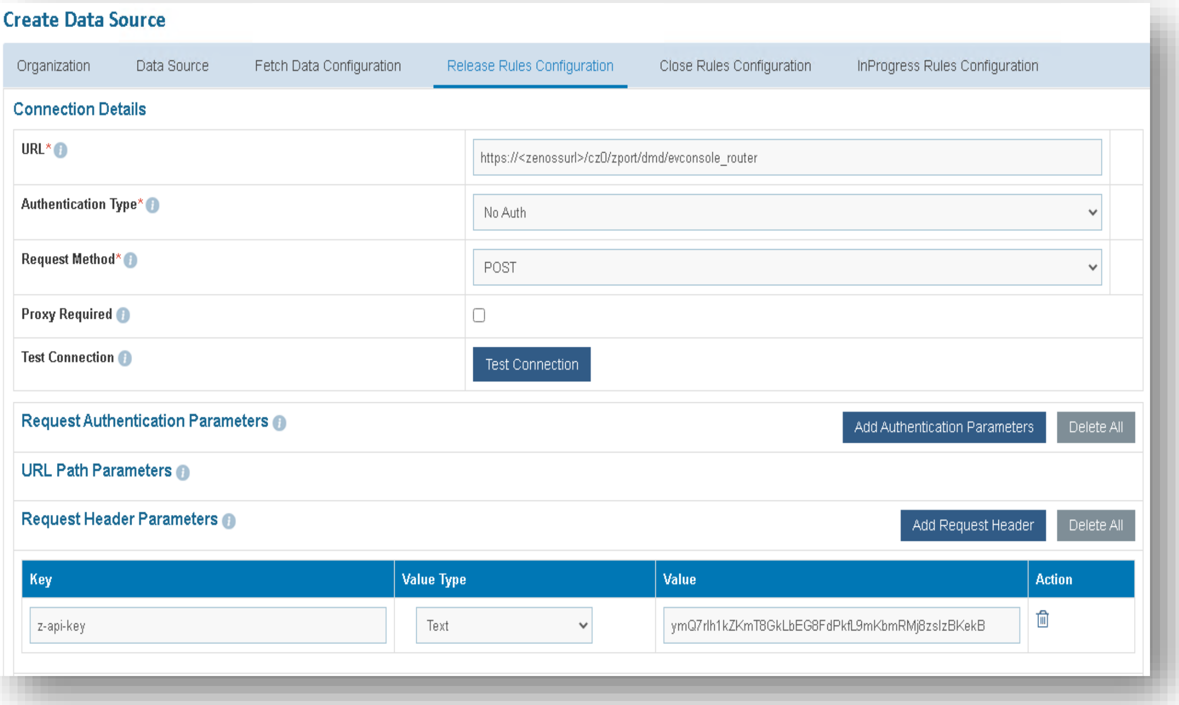
Key	Value Type	Value
Col1	JSON.Keys	result.0.evid



Key	Value Type	Value	Action
Col1	JSON Keys	result.events.0.evid	🗑

Figure 348 – Optional Parameter Mapping

- Click Next to proceed to Release Rules Configuration.
- On **Release Rules Configuration** tab, type in the details as per the requirement.
- In the **Connection Details** section, enter the following details:
  - **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.
  - **Sample URL** - https://<zenossurl>/cz0/zport/dmd/evconsole_router
  - **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously.
  - **Request Method** – Select Request Method as POST from the drop-down.
  - **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
  - Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.



Create Data Source
Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration
Connection Details

URL*	https://<zenossurl>/cz0/zport/dmd/evconsole_router
Authentication Type*	No Auth
Request Method*	POST
Proxy Required	□
Test Connection	Test Connection

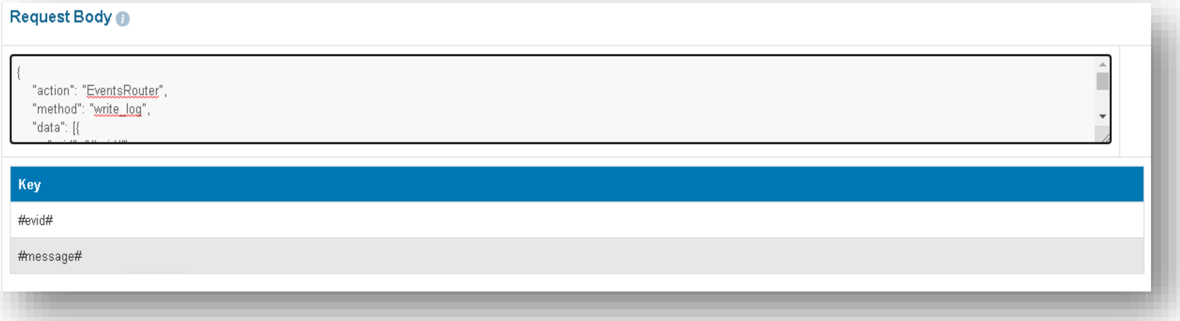
Request Authentication Parameters
Add Authentication Parameters
Delete All
URL Path Parameters
Request Header Parameters
Add Request Header
Delete All

Key	Value Type	Value	Action
z-api-key	Text	ymQ7rlh1kZKmT8GkLbEG8FdPkfL9mKbmRMj8zslzBKekB	🗑

Figure 349 – Release Rules Configuration (Connection Details)

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body –** In this section, please enter the request body in JSON format. A sample request is mentioned below.

```
Request Body –
{
    "action": "EventsRouter",
    "method": "write_log",
    "data": [{
        "evid": "#evid#",
        "message": "#message#"
    }],"tid":2
}
```



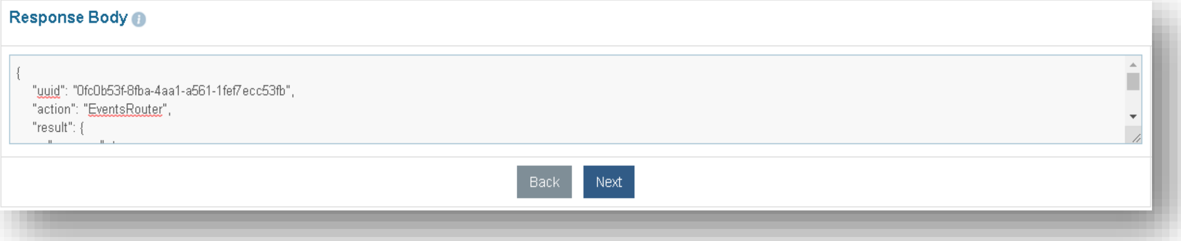
```
{
    "action": "EventsRouter",
    "method": "write_log",
    "data": [{
```

Key
#evid#
#message#

Figure 350 – Release Rules Configuration (Request Body)

– **Response Body –** In this section, please enter the response body in JSON format. A sample response is mentioned below.

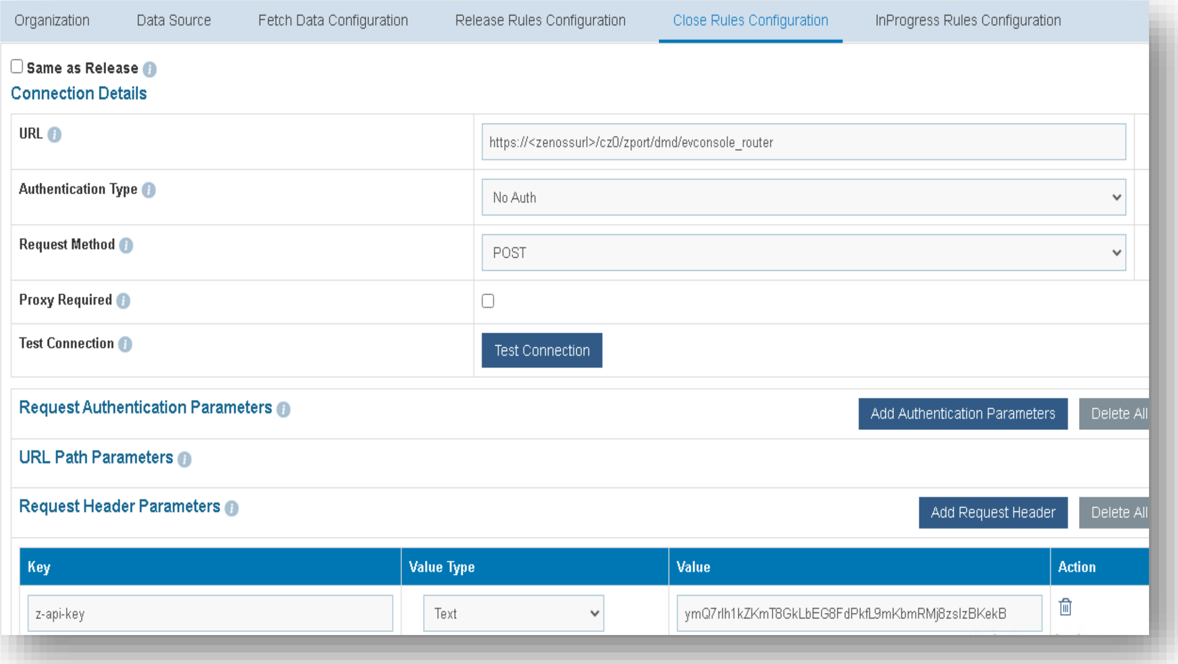
```
Response Body –
{
    "uuid": "0fc0b53f-8fba-4aa1-a561-1fef7ecc53fb",
    "action": "EventsRouter",
    "result": {
        "success": true
    },
    "tid": 2,
    "type": "rpc",
    "method": "write_log"
}
```



```
{
    "uuid": "0fc0b53f-8fba-4aa1-a561-1fef7ecc53fb",
    "action": "EventsRouter",
    "result": {
```

Figure 351 – Release Rules Configuration (Response Body)

- On **Close Rules Configuration** tab, type in the details as per the requirement. Check **Same as Release** if similar configurations as mentioned in "Release Rules Configuration" are required, else proceed ahead.
- In the **Connection Details** section, enter the following details:
  - **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.
  - **Sample URL** - https://<zenossurl>/cz0/zport/dmd/evconsole_router
  - **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously.
  - Request Method – Select Request Method as POST from the drop-down.
  - **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
  - Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.


Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

□ Same as Release

Connection Details

URL	https://<zenossurl>/cz0/zport/dmd/evconsole_router
Authentication Type	No Auth
Request Method	POST
Proxy Required	□
Test Connection	Test Connection

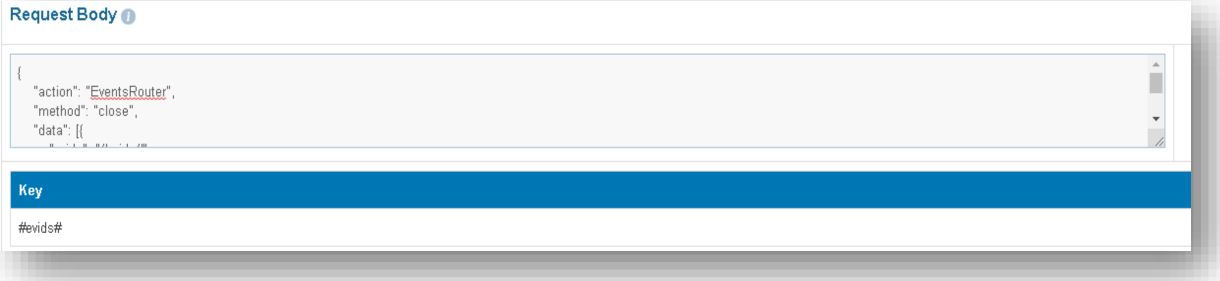
Request Authentication Parameters
Add Authentication Parameters
Delete All
URL Path Parameters
Request Header Parameters
Add Request Header
Delete All

Key	Value Type	Value	Action
z-api-key	Text	ymQ7rlh1kZKmT8GkLbEG8FdPkfL9mKbmRMj8zslzBKekB	🗑

Figure 352 – Release Rules Configuration (Connection Details)

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body –** In this section, please enter the request body in JSON format. A sample request is mentioned below.

```
Request Body –

{

    "action": "EventsRouter",

    "method": "close",

    "data": [{

        "evids": "#evids#"

        }],"tid":2

}
```



Request Body

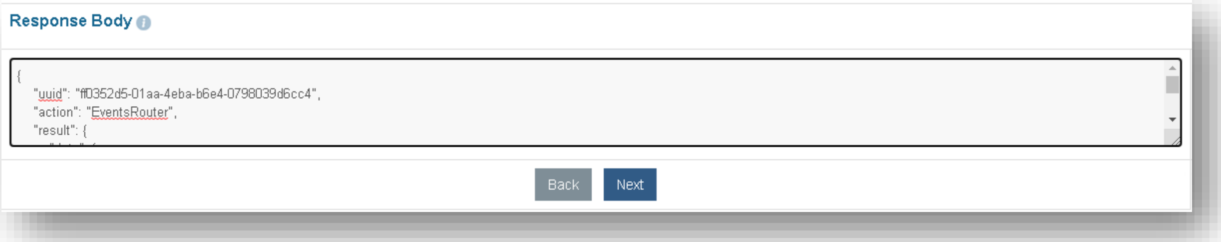
```
{
  "action": "EventsRouter",
  "method": "close",
  "data": [{
```

Key
#evids#

Figure 353 – Release Rules Configuration (Request Body)

– **Response Body –** In this section, please enter the response body in JSON format. A sample response is mentioned below.

```
Response Body –

{

    "uuid": "ff0352d5-01aa-4eba-b6e4-0798039d6cc4",

    "action": "EventsRouter",

    "result": {

        "data": {

            "updated": 31,

            "total": 3670

        },

        "success": true
    },
    "tid": 2,
    "type": "rpc",
    "method": "acknowledge"
}
```



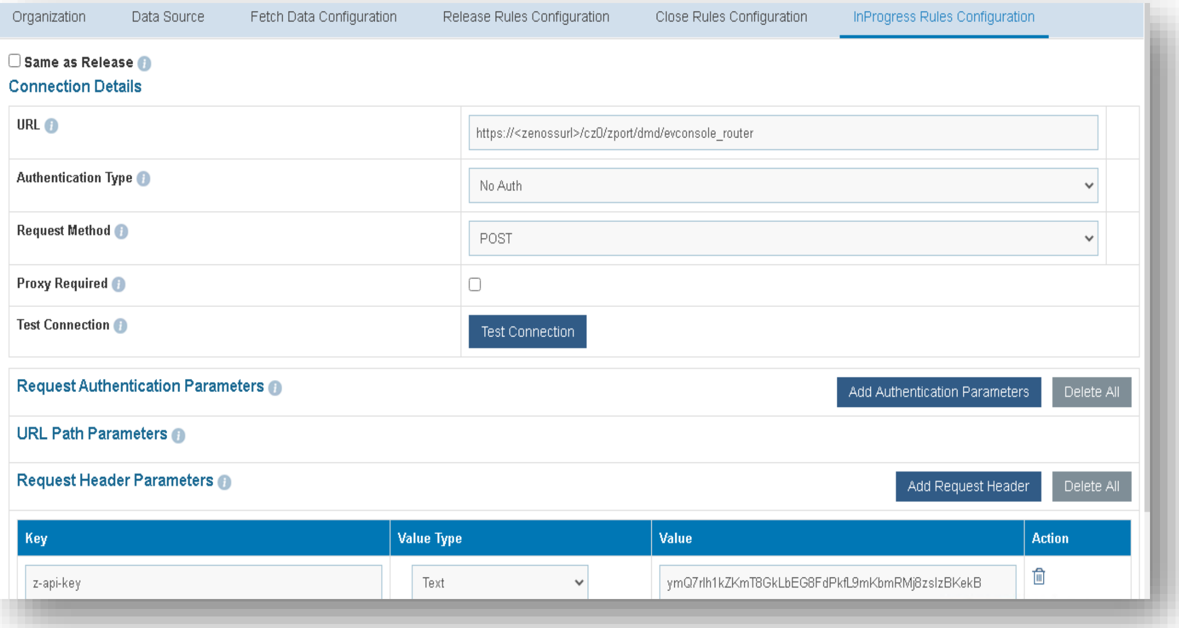
```
{
    "uuid": "ff0352d5-01aa-4eba-b6e4-0798039d6cc4",
    "action": "EventsRouter",
    "result": {
```

Figure 354 – Release Rules Configuration (Response Body)

– On **InProgress Rules Configuration** tab, type in the details as per the requirement. Check **Same as Release** if similar configurations as mentioned in "Release Rules Configuration" are required, else proceed ahead.

– In the **Connection Details** section, enter the following details:

- **URL –** Type the URL of the selected service type to release the ticket. It contains the placeholders that display the parameters based on the applied clause and is dependent on the URL or API provided by the tool.
- **Sample URL** - https://<zenossurl>/cz0/zport/dmd/evconsole_router
- **Authentication Type –** Please enter the information in line with the Authentication type configured for fetching data configuration previously.
- **Request Method** – Select Request Method as POST from the drop-down.
- **Proxy Required –** Check **Proxy Required**, if the environment needs access to content from data sources outside the firewall.
- Click on **Test Connection** to check accessibility of URL from service. Testing the connection is not mandatory, you can still create Data source.


Organization
Data Source
Fetch Data Configuration
Release Rules Configuration
Close Rules Configuration
InProgress Rules Configuration

☐ Same as Release

Connection Details

URL	https://<zenossurl>/cz0/zport/dmd/evconsole_router
Authentication Type	No Auth
Request Method	POST
Proxy Required	☐
Test Connection	Test Connection

Request Authentication Parameters

Add Authentication Parameters
Delete All

URL Path Parameters

Request Header Parameters

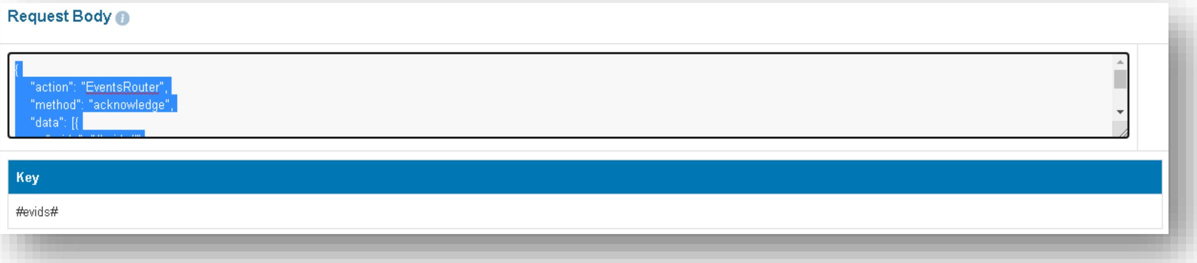
Add Request Header
Delete All

Key	Value Type	Value	Action
z-api-key	Text	ymQ7rlh1kZKmT8GkLbEG8FdPkfL9mKbmRMj8zslzBKekB	🗑

Figure 355 – Release Rules Configuration (Connection Details)

- **Request Header Parameters –** Please enter the request header parameters as required.
- **Request Body –** In this section, please enter the request body in JSON format. A sample request is mentioned below.

```
Request Body –
{
    "action": "EventsRouter",
    "method": "acknowledge",
    "data": [{
        "evids": "#evids#"
    }],"tid":2
}
```



Request Body

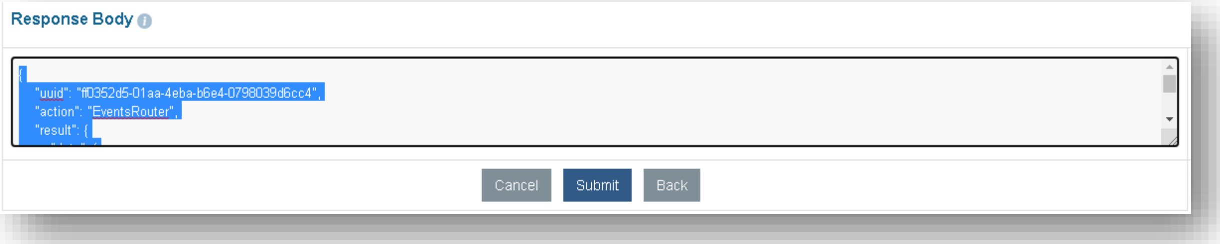
```
{
    "action": "EventsRouter",
    "method": "acknowledge",
    "data": [{
```

Key
#evids#

Figure 356 – Release Rules Configuration (Request Body)

- **Response Body –** In this section, please enter the response body in JSON format. A sample response is mentioned below.

```
Response Body –
{
    "uuid": "ff0352d5-01aa-4eba-b6e4-0798039d6cc4",
    "action": "EventsRouter",
    "result": {
        "data": {
            "updated": 31,
            "total": 3670
        },
        "success": true
    },
    "tid": 2,
    "type": "rpc",
    "method": "acknowledge"
}
```



Response Body

```
{
    "uuid": "ff0352d5-01aa-4eba-b6e4-0798039d6cc4",
    "action": "EventsRouter",
    "result": {
```

Figure 357 – Release Rules Configuration (Response Body)

- Click **Submit** to add the data source.

5 Integration with RBA / Orchestrator Tools

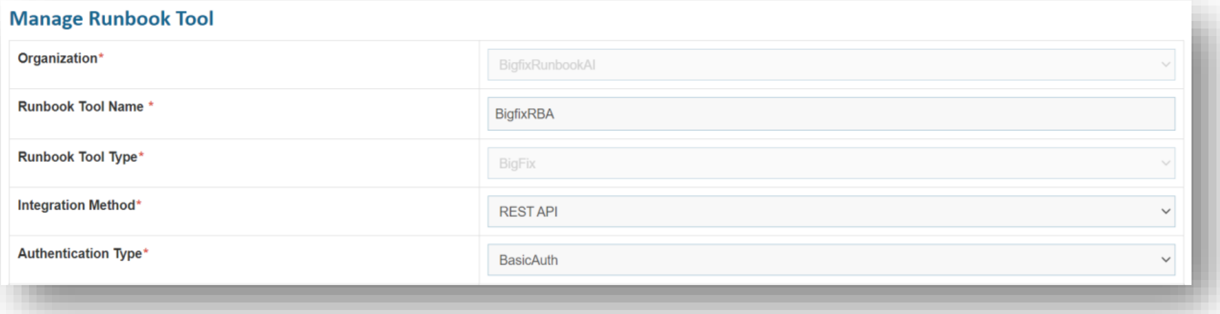
BigFix Runbook AI leverages the services of a Runbook Automation (RBA) / Orchestrator tool to perform actions as defined in the runbooks a.k.a. workflows. Thus, to enable integration with RBA tool, you need to onboard a runbook automation tool through configuration.

Before proceeding with the configuration related to Data Source creation, user has to ensure that an organization has been configured. If not done already, please refer to the Configuration Guide for the same and create the organization before proceeding ahead.

5.1 Integration with BigFiX

To manage / onboard BigFix as the RBA tool, perform the following steps:

- On the main menu bar, click **Runbooks**, and then click **Manage Runbook Tool**. The **Manage Runbook Tool** appears.
- Click **Add New** to add a new tool or click ✎ to edit an existing runbook automation tool.
- Select organization for which you need to create runbook tool in the **Organization Name** field.
- Type the runbook tool name in the **Runbook Tool Name** field.
- Select **BigFix** from the **Runbook Tool Type** drop-down.
- Select **REST** as the integration method for BigFix for the **Integration Method** field.



Manage Runbook Tool

Organization*	BigfixRunbookAI
Runbook Tool Name *	BigfixRBA
Runbook Tool Type*	BigFix
Integration Method*	REST API
Authentication Type*	BasicAuth

Figure 358 - Manage Runbook Tool (cont.)

- Select one of the Authentication Type from BasicAuth.
  - Selection of from **BasicAuth** requires you to enter –
    - User Id

  - Password

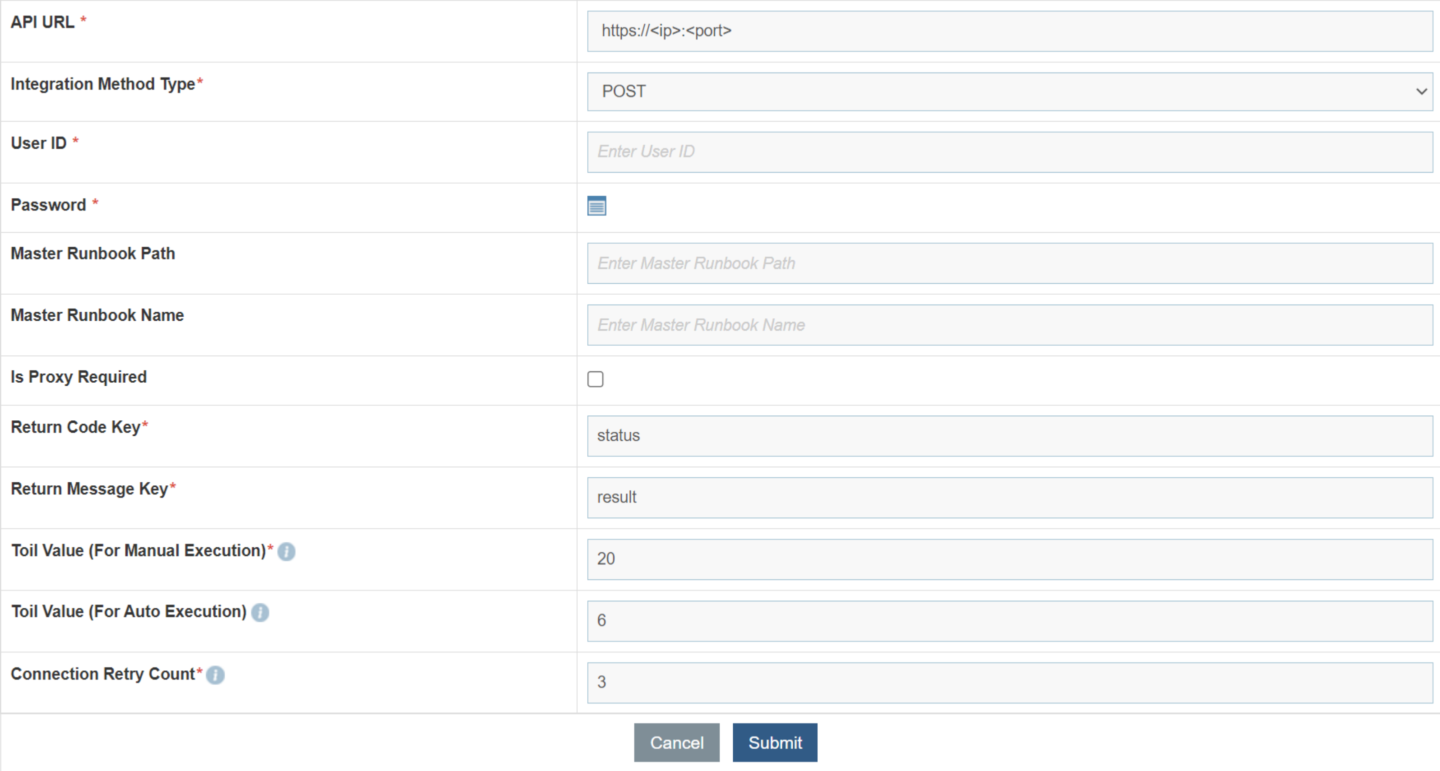
- Type the URL in the **API URL**. field.
- **Sample URL** - *https://<ip>:<port>*
- Select the Integration Method Type as POST
- Type the username and password in the **User ID** and **Password** field to get access to API web services.

API URL, User ID, and Password are dependent on the selected integration method

- Specify the path to get the consolidated scripts for the execution of runbooks in the **Master Runbook Path** field. This will be provided by respective **Runbook Tool** teams if they have a master runbook.

This is not a mandatory field. Users can change and run these scripts any time.

- Select **Proxy Required**, if the environment needs access to content from servers outside a firewall.
- Type the return code key in the **Return Code Key** field to identify the success or failure of runbook execution. E.g., status
- Type the return message key in the **Return Message Key** field to display the success or failure of runbook execution. E.g., result



API URL *	https://<ip>:<port>
Integration Method Type*	POST
User ID *	Enter User ID
Password *	
Master Runbook Path	Enter Master Runbook Path
Master Runbook Name	Enter Master Runbook Name
Is Proxy Required	☐
Return Code Key*	status
Return Message Key*	result
Toil Value (For Manual Execution)*	20
Toil Value (For Auto Execution)	6
Connection Retry Count*	3

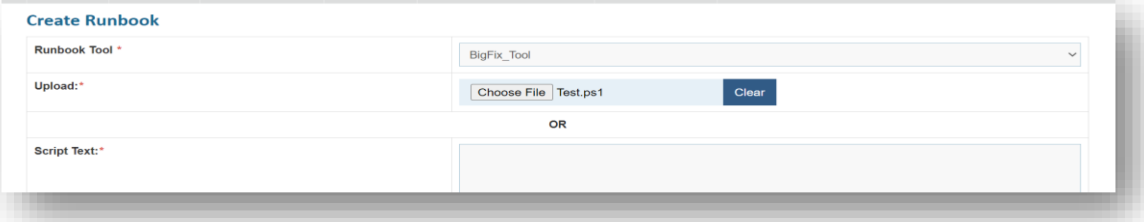
Figure 359 - Manage Runbook Tool (cont.)

- Click **Submit** / **Update** for adding a new tool or making changes to an existing tool. An appropriate success message will be displayed.

5.1.1 Integration with Bigfix Master Fixlet

To create Bigfix master runbook, perform the following steps:

- On the main menu bar, click **Runbooks**, then click **Create Runbook**. The **Create Runbook** page appears.
- Select **Runbook Tool**, the tool against which master runbook has to be created.
- Either **Upload** or type **Script Text**, file has to be uploaded which are of extensions .ps1/.bat/.py/.sh.

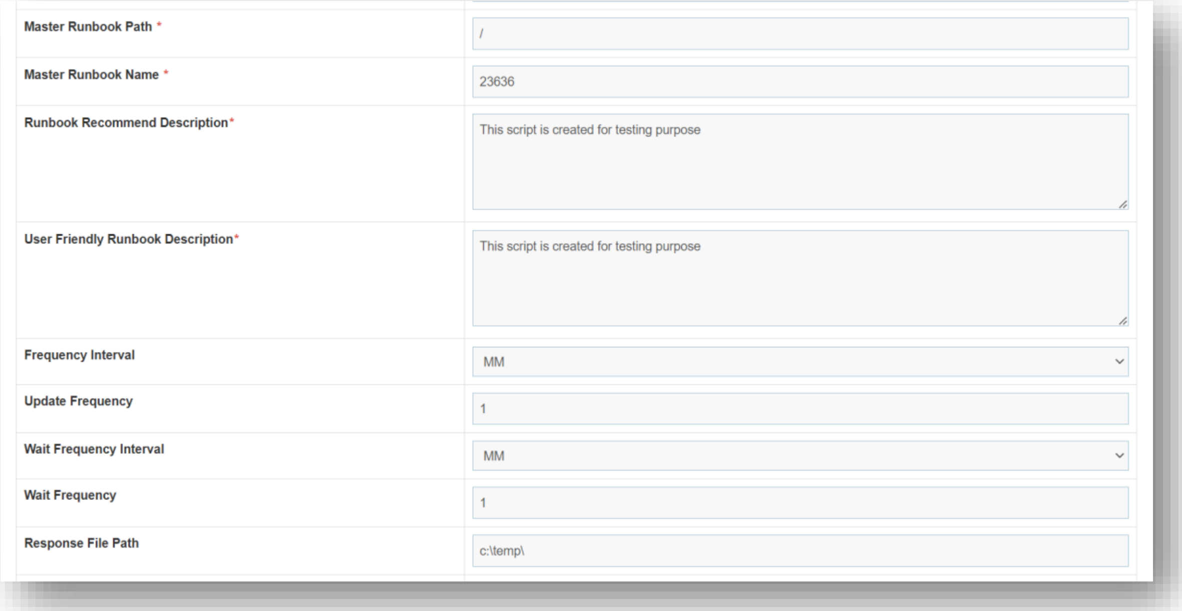


Create Runbook

Runbook Tool *	BigFix_Tool
Upload:*	Choose File Test.ps1 Clear
	OR
Script Text:*	

Figure 360 - Create Runbook

- Type the name of the runbook in **Runbook Name** field.
- Add runbook path in the field **Master Runbook Path**. Although in case of bigfix, this can be given any value, since bigfix integration is independent of runbook path.
- Type the value of master fixlet ID in the field **Master Runbook Name**.
- Add the path of 'error_folder' in the field **Response File Path**. While creation of Bigfix Master Runbook, this field is mandatory.

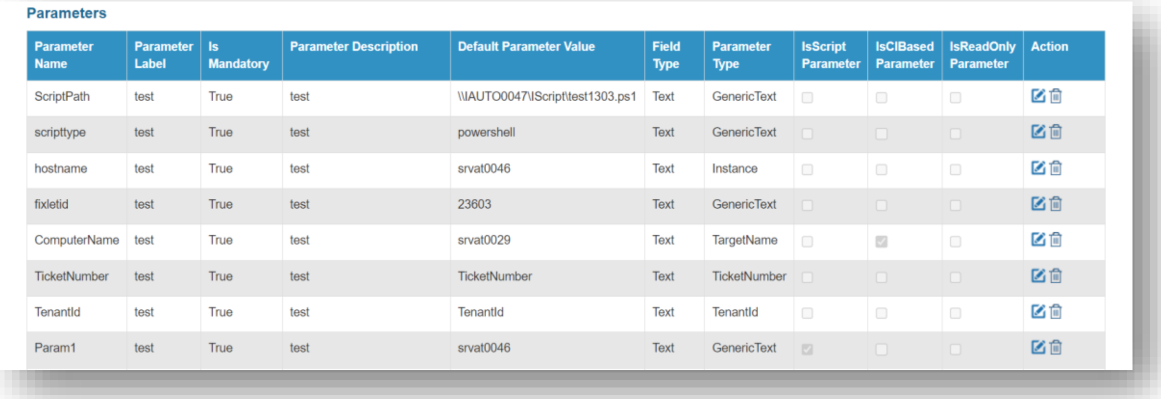


Master Runbook Path *	/
Master Runbook Name *	23636
Runbook Recommend Description*	This script is created for testing purpose
User Friendly Runbook Description*	This script is created for testing purpose
Frequency Interval	MM
Update Frequency	1
Wait Frequency Interval	MM
Wait Frequency	1
Response File Path	c:\temp\

Figure 361 - Create Runbook Contd..

- Add the following Parameter Names in the parameter grid:
- **ScriptPath** – The default parameter value consists of the shared path.
- **ScriptType** – The default parameter value consists of the type of the script uploaded.
- **Hostname** – The default parameter value consists of the target server on which script is getting executed.

- **Fixletid** - The default parameter value consists of the value of the ID of child fixlet executed.
- **Computername** – The default parameter consists of the value of the master server or the root server.
- **TicketNumber** – The default parameter consists of the static value 'TicketNumber' and it is mapped with TicketNumber in Parameter Type .
- **TenantID** – The default parameter consists of the static value 'TenantID' and it is mapped with TenantID in Parameter Type.
- **Param1** – The default parameter consists of the parameter value user wants to add in. If user wants to add multiple parameters, those are also added in the similar manner like param1. Furthermore, it needs to be checked in for 'IsScript Parameter'.



Parameters

Parameter Name	Parameter Label	Is Mandatory	Parameter Description	Default Parameter Value	Field Type	Parameter Type	IsScript Parameter	IsCIBased Parameter	IsReadOnly Parameter	Action
ScriptPath	test	True	test	\\IAUTO0047\IScript\test1303.ps1	Text	GenericText	☐	☐	☐	✎ 🗑
scripttype	test	True	test	powershell	Text	GenericText	☐	☐	☐	✎ 🗑
hostname	test	True	test	srvat0046	Text	Instance	☐	☐	☐	✎ 🗑
fixletid	test	True	test	23603	Text	GenericText	☐	☐	☐	✎ 🗑
ComputerName	test	True	test	srvat0029	Text	TargetName	☐	☑	☐	✎ 🗑
TicketNumber	test	True	test	TicketNumber	Text	TicketNumber	☐	☐	☐	✎ 🗑
TenantId	test	True	test	TenantId	Text	TenantId	☐	☐	☐	✎ 🗑
Param1	test	True	test	srvat0046	Text	GenericText	☑	☐	☐	✎ 🗑

Figure 362 - Parameter grid in Create Runbook.

- Select 'Save' button after adding all the details for the master runbook.
- Note: The master runbook created on 'Create Runbook' will be visible in Manage Runbooks. (On main menu, go to Runbooks and select manage runbooks.)

6 Appendix

Table 72 List of Abbreviations

Abbreviation	Expansion
AD	Active Directory
AI	Artificial Intelligence
ITOPS	IT Operations
ITSMS	IT Service Management System

KEDB	Known Error Database
SNOW	ServiceNow