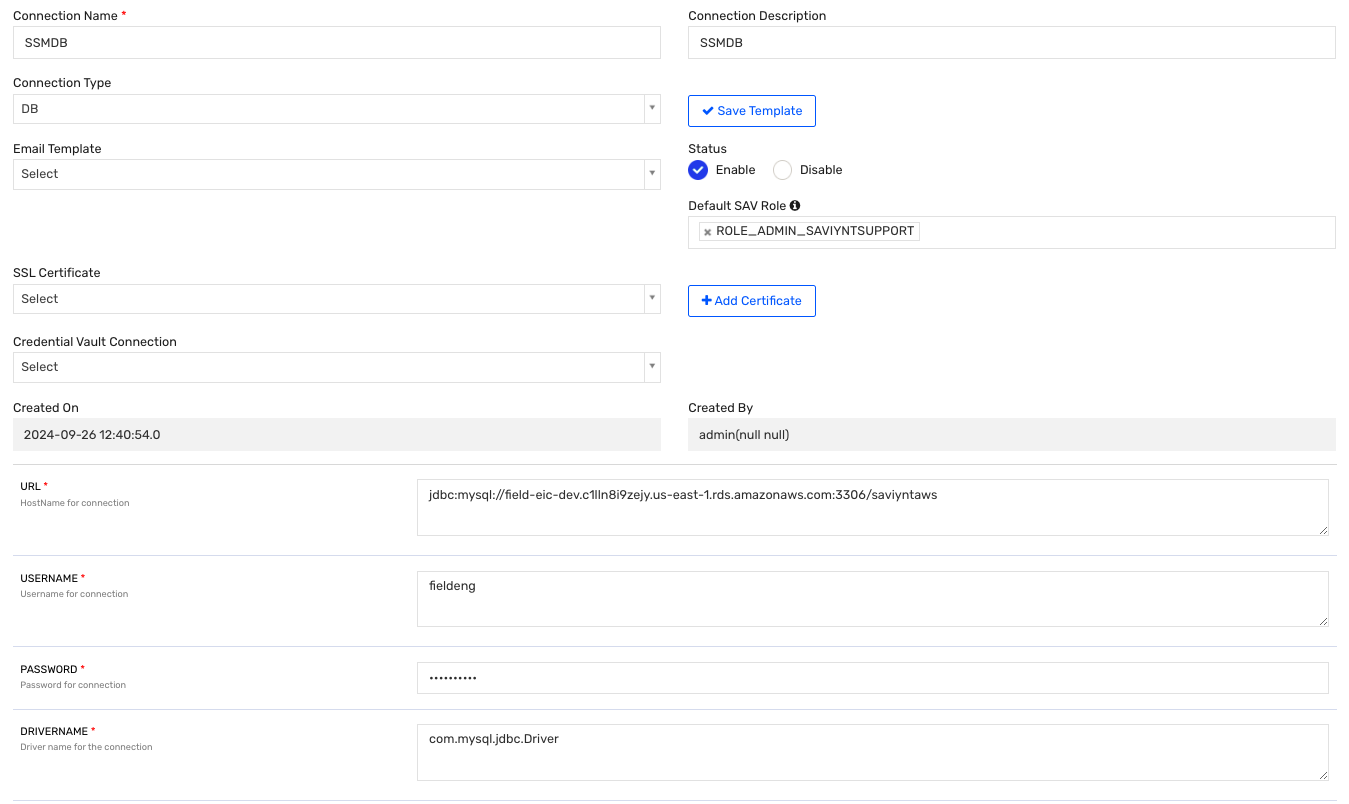
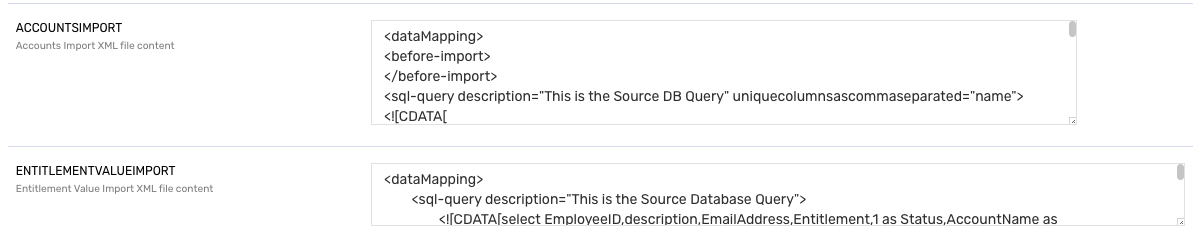
Sample DB connector to pull users from Database

Introduction:  
Saviynt EIC offers OOB capability to integrate DB applications with Saviynt with the help of database connectors. Saviynt Provides OOB templates that can be configured accordingly.  
  
  
How it works:  
  
Saviynt offers a connector module which can be configured to make a connection with the application.  
  
  
Prerequisites:   
  
Network connectivity should be established before we can begin operations like recon and provisioning. The network needs to be whitelisted, and firewall should allow the network to flow from Saviynt to EIC and vice versa.  
  
Integration Steps:  
  
Create an DB connection. Let’s look at the integration steps .



1)Provide a Connection Name and Description.  
2) Select the Connection Type as DB, it will load the template below according to AD connection.  
3) Give the URL: This is the hostname of the DB to which we are trying to connect. Provide schema also  
4) Give the username and password.  
5) Provide the drivername accordingly to the database to which we are connecting. com.mysql.jdbc.Driver in case of mysql



6) Provide The user Import:   
  
example:  
<dataMapping>

<before-import>

</before-import>

<sql-query description="This is the Source DB Query" uniquecolumnsascommaseparated="username">

<![CDATA[

select username,(CASE WHEN statuskey IS NULL THEN '1' ELSE NULL END) as manager from users where statuskey = 1;

]]>

</sql-query>

<importsettings>

<zeroDayProvisioning>false</zeroDayProvisioning>

<userNotInFileAction>NOACTION</userNotInFileAction>

<checkRules>true</checkRules>

<buildUserMap>false</buildUserMap>

<userReconcillationField>username</userReconcillationField>

</importsettings>

<mapper description="This is the mapping field for SAviynt Field name" dateformat="date" >

<mapfield saviyntproperty="username" sourceproperty="username" type="character"></mapfield>

<mapfield saviyntproperty="manager" sourceproperty="manager" type="character"></mapfield>

</mapper>

<after-import description="SQL">

</after-import>

</dataMapping>

* **Data Mapping Structure**:
  + <dataMapping> defines the entire import process.
* **Before Import**:
  + <before-import> is an optional section for tasks or configurations to execute before the main import. It’s empty in this configuration.
* **SQL Query**:
  + <sql-query> defines the SQL statement used to pull data from the source database.
  + **Attributes**:
    - description: Provides a description of the query.
    - uniquecolumnsascommaseparated: Specifies username as the unique column for this query.
  + **Query Body**:
    - The SQL query retrieves the username and manager status from the users table.
    - manager is derived using a CASE statement: it sets manager to '1' if statuskey is NULL, otherwise manager will be NULL.
    - The WHERE clause filters records where statuskey is equal to 1.
* **Import Settings**:
  + <importsettings> configures various settings for the import.
  + **Attributes**:
    - zeroDayProvisioning: Set to false, meaning zero-day provisioning isn’t enabled.
    - userNotInFileAction: NOACTION implies no action will be taken if a user in Saviynt is missing in the source data.
    - checkRules: true indicates that Saviynt will evaluate any configured rules.
    - buildUserMap: false means a user map won’t be built during this import.
    - userReconcillationField: username is used to reconcile users.
* **Mapper**:
  + <mapper> maps fields from the source data to corresponding Saviynt properties.
  + **Attributes**:
    - description: Provides details on the field mappings.
    - dateformat: Specifies the date format used (though no dates are mapped in this example).
  + **Map Field Elements**:
    - <mapfield> entries specify mappings:
      * username from the source is mapped to Saviynt’s username.
      * manager from the source query is mapped to Saviynt’s manager property.
* **After Import**:
  + <after-import> allows defining post-import actions, though it’s only labeled here with "SQL" and contains no specific actions.

This configuration facilitates importing user information, such as usernames and manager status, and includes reconciliation, mapping, and rules-checking settings that control how the data is processed and integrated in Saviynt.

**Troubleshooting:  
The most common errors include syntax error in the query and data mismatch.  
  
Resolution:   
Please run the query from the xml in the target database to validate the result set.**