



Tot plugin JDBC

Control de versiones	1
Modelo de integración	1
Extracción de metadatos	1
Muestreo de datos	3
Credenciales requeridas	3
Extracción de metadatos	3
Muestreo de datos	3
Despliegue	3
Configuración	4
ImAri disponibles	5
Vendors y driver disponibles	5
Errores comunes	5

Control de versiones

Versión	Fecha de modificación	Responsable	Aprobador	Resumen de cambios
1.0	31/03/2023	Anjana Producto	Anjana Producto	Creación del documento Añadido driver de Snowflake

Modelo de integración

Extracción de metadatos

Se utilizan los métodos que ofrece el driver de JDBC mediante los cuales se accede a la definición de esquemas y tablas.

Extrae los siguientes atributos que deben llamarse igual en la tabla `attribute_definition`, campo `name` para que aparezcan en la plantilla.

- **catalog** con el valor de catalog en la base de datos
- **schema** con el valor de schema en la base de datos
- **physicalName** y **name** con el mismo valor, el nombre de la tabla
- **path** con la concatenación de los valores de catalog, schema and table
- **infrastructure** con el valor seleccionado
- **technology** con el valor seleccionado
- **zone** con el valor seleccionado

También nos enviará los siguientes atributos relativos a los campos del recurso pedido:

- **name** con el valor del campo correspondiente
- **physicalName** con el valor del campo correspondiente
- **defaultValue** con el valor por defecto definido para el campo correspondiente
- **fieldDataType** con el tipo de dato definido para el campo correspondiente
- **length** con el tamaño del campo correspondiente
- **incrementalField** indicando si es un campo incremental
- **position** posición que ocupa el campo correspondiente
- **precision** con el valor de la precisión del campo correspondiente
- **nullable** indicando si el campo correspondiente es nullable
- **pk** indicando si el campo es una pk
- **description** con el valor correspondiente para el campo

El plugin es capaz de realizar la extracción de metadatos de los siguientes tipos de elementos:

- Tabla de base de datos

Muestreo de datos

Utilizando el driver genérico de JDBC de Java se ejecuta una query simple de SELECT para acceder a un número limitado de elementos de la tabla para recuperar una muestra de los datos almacenados. Adicionalmente se sustituyen los valores de los campos sensibles por asteriscos.

Credenciales requeridas

Extracción de metadatos

Usuario con permisos/roles necesarios para hacer SELECT sobre los catálogos, esquemas y tablas que se quieran gobernar.

Muestreo de datos

Usuario con permisos/roles necesarios para hacer SELECT las tablas y el permiso de USAGE para cada uno de los esquemas que se quieran gobernar.

Despliegue

Se ha de seguir el manual genérico del despliegue de plugins.

Doc: Anjana Data x.x - DOC - Tot despliegue de plugins

Configuración

Aquí se incluye el detalle de la configuración específica del plugin, para revisar la configuración común, mirar el documento Anjana Data 4.4 - DS - Configuración técnica de Portal y microservicios

Configuraciones específicas:

- Database url, username and password:
 - url: jdbc:postgresql://<database_host>:<port>/database=<database>
 - user: <user>
 - password: <password>
 - path-separator
 - using-catalogs: false
 - using-schemas: true

Si “using-catalogs” y “using-schemas” están a false, solo muestran el schema por defecto o el elegido en la url de conexión.

Using-schemas lista todas las tablas de todos los esquemas que tenga acceso

Using-catalogs lista todas las tablas de todos los schemas de todas las bases de datos que vea.

Path-separator va a indicar el separador utilizado por parte de Anjana para el path.

El parámetro groupPrefix indica el prefijo que tiene el nombre del grupo.

Config file:

```
server:
  port: 15001

totplugin:
  location: http://totpluginjdbcserver:15001/plugin/jdbc/api/v1
  server:
    url: http://totserver:15000/tot/
  connection:
    vendor: POSTGRESQL
    driver: org.postgresql.Driver
```

```
url: jdbc:postgresql://rdbservice:5432/anjana
user: anjana
password: ****
path-separator: "/"
using-catalogs: false
using-schemas: true

aris:
- ari: "anja:totplugin:sample:/jdbc/Postgresql/postgresql/"
- ari: "anja:totplugin:extract:/jdbc/Postgresql/postgresql/"
groupPrefix: Dsa_
```

ImAri disponibles

- Azure
- Ldap

Vendors y driver disponibles

Para cada vendor se detalla el driver a configurar en la propiedad totplugin.connection.driver, siendo el vendor el que se tiene que configurar en totplugin.conneccion.driver

- ORACLE
 - oracle.jdbc.driver.OracleDriver
 - jdbc:oracle:thin:@<database_host>:<port>/<service>
- HIVE
 - Driver de Apache → org.apache.hive.jdbc.HiveDriver (v2.3.9)
 - Driver de Cloudera → com.cloudera.hive.jdbc.HS2Driver (v2.6.21.1025)
 - jdbc:hive2://<database_host>:<port>/<database>
- MYSQL
 - com.mysql.cj.jdbc.Driver
 - Para el driver de MySQL los schemas son catalogs y viceversa, es decir, para crear schemas hay que configurar la propiedad de using-catalogs a true y la de using-schemas a false.
 - jdbc:mysql://<database_host>:<port>/<database>
- POSTGRESQL
 - org.postgresql.Driver
 - jdbc:postgresql://<database_host>:<port>/database=<database>
- DB2
 - com.ibm.db2.jcc.DB2Driver
 - jdbc:db2://<database_host>:<port>/<database>
- MSSQLSERVER
 - com.microsoft.sqlserver.jdbc.SQLServerDriver
 - jdbc:sqlserver://<database_host><instance_name>:<port>
- H2
 - org.h2.Driver

- jdbc:h2:mem:<database>
- MARIADB
 - org.mariadb.jdbc.Driver
 - jdbc:mariadb://<database_host>:<port>/<database>
- SNOWFLAKE
 - net.snowflake.client.jdbc.SnowflakeDriver
 - jdbc:snowflake://<database_host>:<port>/?db=<database>&warehouse=<warehouse>

Errores comunes

- Si, al ejecutar la extracción de metadato conectando con una base de datos de Oracle sale el siguiente error:

```
1] oracle.simplefan.FanManager : attempt to configure ONS in FanManager failed with oracle.ons.NoServersAvailable: Server time out
1] c.a.t.p.js.ExtractMetadataService : Exception on extractMetadata
java.lang.IllegalArgumentException: ONS configuration failed
    at oracle.simplefan.impl.FanManager.configure(FanManager.java:108) ~[simplefan-21.3.0.0.jar!/:21.3.0.0.0]
    at oracle.jdbc.driver.HAManager.configure(HAManager.java:146) ~[ojdbc8-12.2.0.1.jar!/:12.2.0.1.0]
    at oracle.jdbc.driver.HAManager.<init>(HAManager.java:122) ~[ojdbc8-12.2.0.1.jar!/:12.2.0.1.0]
    at oracle.jdbc.driver.HAManager.getInstance(HAManager.java:126) ~[ojdbc8-12.2.0.1.jar!/:12.2.0.1.0]
    at oracle.jdbc.driver.HAManager.enableHAIfNecessary(HAManager.java:725) ~[ojdbc8-12.2.0.1.jar!/:12.2.0.1.0]
    at oracle.jdbc.driver.PhysicalConnection.connect(PhysicalConnection.java:713) ~[ojdbc8-12.2.0.1.jar!/:12.2.0.1.0]
    at oracle.jdbc.driver.T4CDriverExtension.getConnection(T4CDriverExtension.java:39) ~[ojdbc8-12.2.0.1.jar!/:12.2.0.1.0]
    at oracle.jdbc.driver.OracleDriver.connect(OracleDriver.java:691) ~[ojdbc8-12.2.0.1.jar!/:12.2.0.1.0]
    at java.sql.DriverManager.getConnection(DriverManager.java:664) ~[na:1.8.0_352]
```

Es necesario incluir en el comando de ejecución del descriptor de servicio del plugin “-Doracle.jdbc.fanEnabled=false” (info del problema: https://support.oracle.com/cloud/faces/DocumentDisplay?_afLoop=190836230347481&_afWindowMode=0&id=2616175.1&_adf.ctrl-state=qk122q2vr_4)

- En caso de que las tablas de Hive estén encriptadas, se recupera este log de error

```
No Route to Host from client.domain/127.0.0.1 to client.domain:8020
failed on socket timeout exception: java.net.NoRouteToHostException
```