

OrientDB Connector

According to the [OrientDB's website](#):

OrientDB is a new Open Source NoSQL DBMS born with the best features of all the others. It's written in Java and it's amazing fast: can store up to 150,000 records per second on common hardware. Even if it's Document based database the relationships are managed as in Graph Databases with direct connections among records. You can traverse entire or part of trees and graphs of records in few milliseconds. Supports schema-less, schema-full and schema-mixed modes. Has a strong security profiling system based on user and roles and support the SQL between the query languages. Thank to the SQL layer it's straightforward to use it for people skilled in Relational world.

Download

The Gitorious repository is hosted [here](#). You can clone it with

```
git clone git://gitorious.org/shibboleth-idp-orientdb-connector/shibboleth-idp-orientdb-connector.git
```

Installation

1. Compile the source with:

```
mvn clean package
```

2. Extract all the jars archived in the `lib` folder of the newly created `target/orientdb-connector-<version>-bin.zip` into the Shibboleth IdP source lib directory.
3. Re-install the IdP using the install script to build and deploy a new idp.war

Configuration

Enable the connector in `$IDP_CONFIG_DIR/attribute-resolver.xml` by doing the following:

1. Add those namespace and xsd schema to the root element:

```
xmlns:ino="http://inocybe.ca/NS/orientdb-connector"  
http://inocybe.ca/NS/orientdb-connector classpath:/schema/orientdb-connector.xsd
```

to make it look something like:

```

<AttributeResolver xmlns="urn:mace:shibboleth:2.0:resolver" xmlns:resolver="urn:mace:shibboleth:2.0:
resolver"
                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:pc="urn:mace:shibboleth:
2.0:resolver:pc"
                xmlns:ad="urn:mace:shibboleth:2.0:resolver:ad" xmlns:dc="urn:mace:shibboleth:2.0:
resolver:dc"
                xmlns:enc="urn:mace:shibboleth:2.0:attribute:encoder" xmlns:sec="urn:mace:shibboleth:
2.0:security"
                xmlns:ino="http://inocybe.ca/NS/orientdb-connector"
                xsi:schemaLocation="urn:mace:shibboleth:2.0:resolver classpath:/schema/shibboleth-2.0-
attribute-resolver.xsd
                                urn:mace:shibboleth:2.0:resolver:pc classpath:/schema/shibboleth-
2.0-attribute-resolver-pc.xsd
                                urn:mace:shibboleth:2.0:resolver:ad classpath:/schema/shibboleth-
2.0-attribute-resolver-ad.xsd
                                urn:mace:shibboleth:2.0:resolver:dc classpath:/schema/shibboleth-
2.0-attribute-resolver-dc.xsd
                                urn:mace:shibboleth:2.0:attribute:encoder classpath:/schema
/shibboleth-2.0-attribute-encoder.xsd
                                urn:mace:shibboleth:2.0:security classpath:/schema/shibboleth-2.0-
security.xsd
                                http://inocybe.ca/NS/orientdb-connector classpath:/schema
/orientdb-connector.xsd">

```

2. Configure the connector. Its attributes are:

- orientDbHost:** The host where the OrientDB is running
- orientDbPort:** The port on the host where the OrientDB is running
- orientDbName:** The database containing the users' table
- username:** The username to use to connect to the database
- password:** The password to use to connect to the database

The connector also has a mandatory `QueryTemplate` element to specify how to query the database to retrieve attributes and optional `AttributeMap` elements.

Here are some example configurations:

Example 1 (without attribute mapping)

```

<resolver:DataConnector id="myOrient" xsi:type="ino:OrientDbDataConnector"
                        orientDbHost="myorienthost.com"
                        orientDbPort="2424"
                        orientDbName="database"
                        username="admin"
                        password="password">
    <ino:QueryTemplate>
        <![CDATA[
            select from users where uid = '$requestContext.principalName'
        ]]>
    </ino:QueryTemplate>
</resolver:DataConnector>

```

Example 2 (with attribute mapping)

```

<resolver:DataConnector id="myOrient" xsi:type="ino:OrientDbDataConnector"
                        orientDbHost="myorienthost.com"
                        orientDbPort="2424"
                        orientDbName="database"
                        username="admin"
                        password="password">

  <ino:QueryTemplate>
    <![CDATA[
      select from users where uid = '$requestContext.principalName'
    ]]>
  </ino:QueryTemplate>

  <ino:AttributeMap orientKey="email" attributeID="mail" />
  <ino:AttributeMap orientKey="telephone" attributeID="homePhone" />
</resolver:DataConnector>

```

3. Configure all the `<resolver:AttributeDefinition ...>` elements to make them use the OrientDB connector by making sure their `<resolver:Dependency ...>` element point to the correct connector. For instance:

```

<resolver:AttributeDefinition id="uid" xsi:type="Simple" xmlns="urn:mace:shibboleth:2.0:resolver:ad"
                             sourceAttributeID="uid">
  <resolver:Dependency ref="myOrient" />

  <resolver:AttributeEncoder xsi:type="SAML1String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
                             name="urn:mace:dir:attribute-def:uid" />

  <resolver:AttributeEncoder xsi:type="SAML2String" xmlns="urn:mace:shibboleth:2.0:attribute:encoder"
                             name="urn:oid:0.9.2342.19200300.100.1.1" friendlyName="uid" />
</resolver:AttributeDefinition>

```

Logging

To enable logging, add the following in `$IDP_CONFIG_DIR/logging.xml`

```

<logger name="ca.inocybe.shibboleth" level="INFO" />

```

Bugs, questions and feedback

Contact jonathan.tellier@gmail.com