

Grouper funnel

We have several information systems from which we need to get information about members of these groups:

- teachers, staff, students,...
- courses students and teachers
- lessons students and teachers
- departments students, teachers, ...
- ...

GrouperLoader doesn't fit for us because some systems (for example AFS groups) doesn't have database interface. Also we want fast tool which will return groups and their members in few seconds and doesn't make significant load to source information systems.

So we develop simple application which provides generic database interface and interface for developing custom classes.

You can download [GrouperFunnel.zip](#) which contains:

- sample config file
- binary distribution with required libraries
- Java sources
- shell script

Sample Spring configuration file.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN" "http://www.springframework.org/dtd/spring-beans.dtd">
<beans>
    <!--definition of datasource for custom Source-->
    <bean id="IdmDS" class="oracle.jdbc.pool.OracleDataSource" destroy-method="close">
        <property name="URL" value="jdbc_url" />
        <property name="user" value="" />
        <property name="password" value="" />
    </bean>

    <!--datasource for generic SQL Source-->
    <bean id="StagDS" class="oracle.jdbc.pool.OracleDataSource" destroy-method="close">
        <property name="URL" value="" />
        <property name="user" value="" />
        <property name="password" value="" />
    </bean>

    <bean id="teachers" class="cz.zcu.civ.feeder.SQLSource">
        <!--used datasource-->
        <property name="dataSource" ref="StagDS" />
        <!--sql to execute-->
        <property name="sql">
            <value>
                <![CDATA[select distinct LOWER(z.prac_zkr) AS prac_zkr,o.USERNAME from
studenti stu,
orion o,
osoby os,
znamky z
where
stu.OSOBIDNO = os.OSOBIDNO
and os.ROD_CISLO = o.ROD_CISLO
and o.USERNAME is not null
and stu.STAV <> 'N'
and z.OS_CISLO = stu.OS_CISLO
ORDER BY o.USERNAME]]>
            </value>
        </property>
        <!--name of column with user ID-->
        <property name="subjectColumn" value="username" />
        <!--name of group in Grouper-->
        <property name="groupPrefix" value="teachers" />
        <!--name of source - used in logging-->
        <property name="name" value="Teachers" />
        <!--parent stem-->
        <property name="stem" value="gen:stag" />
        <!--comma separated list of users ID which has admin privileges-->
        <property name="permissions" value="IDM" />
    </bean>
```

```

        <bean id="distancniStudenti" class="cz.zcu.civ.feeder.SQLSource">
            <property name="dataSource" ref="StagDS" />
            <property name="sql">
                <value>
                    <![CDATA[select o.USERNAME
from
    studenti st,
    studijni_programy sp,
    osoby os,
    orion o
where
    st.STPRIDNO = sp.STPRIDNO
    and sp.FORMA = 'D'
    and st.STAV <>'N'
    and os.OSOBIDNO = st.OSOBIDNO
    and os.ROD_CISLO = o.ROD_CISLO
    and o.USERNAME is not null
ORDER BY o.USERNAME]]>
                </value>
            </property>
            <property name="subjectColumn" value="username" />
            <property name="groupPrefix" value="students-dist" />
            <property name="name" value="Distancni studenti" />
            <property name="stem" value="gen:stag" />
            <property name="permissions" value="IDM" />
        </bean>

        <!--custom Source which uses DB but make -->
        <bean id="idmSkupiny" class="cz.zcu.civ.feeder.IdmSource">
            <property name="dataSource" ref="IdmDS" />
            <property name="name" value="IDM skupiny" />
            <property name="stem" value="gen:cro" />
            <property name="permissions" value="IDM" />
        </bean>

        <!--special kind of source which execute external script and gets data from its standard output.-->
        <!--<bean id="sampleApp" class="cz.zcu.civ.feeder.AppSource">
            <property name="name" value="Sample App" />
            <property name="stem" value="bbb" />
            <property name="command" value="echo testgroup#foo,bar,magic" />
        </bean>
-->

        <bean id="feeder" class="cz.zcu.civ.feeder.Feeder">
            <property name="sources">
                <list>
                    <ref bean="teachers" />
                    <ref bean="distancniStudenti" />
                    <ref bean="idmSkupiny" />
                    <ref bean="sampleApp" />-->
                </list>
            </property>
        </bean>
</beans>

```

Every source is fetched in separate thread. Result is stored in simple csv file. We plan to change it to more verbose XML format.

Currently we have config file with twelve sources. With optimized sql and custom classes completes fetching of all data in several seconds. With [our custom provisioning](#) we are able to accomplish near realtime distribution of changes in central systems to Grouper and consequently in Sun IDM.