

Configuring org.clazzes.login.sql 1.0 to 1.1

Configuration of sql-login-service 1.0 or 1.1

The SQL login service may be configured using the OSGi configuration PID `org.clazzes.login.sql` using the configuration values shown in the table below.

Beginning with the version 1.1.0 (released 2013-02-13), all query strings default to the database structure used by the upcoming [SDS \(SQL Directory Service\)](#) bundle. When using another database structure that does not allow some of the queries, it is important set those configuration values to empty strings; deleting them will not help because default values would kick in right away.

Results of list queries (group memberships, group members) are sorted naturally in the Java layer, so there is no need to use ORDER BY clauses. ORDER BY clauses often provoke temporary tables and filesort, which is quite expensive for queries used quite often.

Application configuration

Key	Description
<code>deactivateUserStatement</code>	<p>Required non-empty for <i>deactivateUser</i> feature.</p> <p>SQL template for a prepared statement to deactivate a user.</p> <p>Default, appropriate for SDS' tables: <code>UPDATE USERS SET PASSWORD='{disabled}' WHERE USERID=?</code></p>
<code>defaultDomain</code>	<p>Optional. Defaults to an empty string.</p> <p>If set to an empty string, the domain parameter of request queries does not get checked but is filled in in returned principals.</p> <p>If set to a non-empty string, requests for other domains are refused.</p> <p>Support for databases maintaining multiple authentication domains might be added in the future, but I do not believe multi-domain setups even exist outside the LDAP/ADS world.</p>
<code>groupsByUserIdQuery</code>	<p>Required non-empty for <i>getGroups</i> feature.</p> <p>SQL template for a prepared statement to query the group IDs and group names of the groups of which the user specified by a <code>userId</code> is a member.</p> <p>Default, appropriate for SDS' tables: <code>SELECT g.GROUPID, g.GROUPNAME FROM GROUPS AS g, USERS AS u, GROUPMEMBERSHIPS AS m WHERE u.USERID='?' AND m.USER_ID = u.ID AND g.ID = m.GROUP_ID</code></p>
<code>defaultPasswordAlgorithm</code>	<p>Optional. Defaults to <code>crypt</code></p> <p>Values supported so far: <code>crypt</code>, <code>ssha1</code>, <code>plain</code>.</p> <p>Password fields may contain:</p> <ul style="list-style-type: none">• the password encrypted using the default password algorithm, or• a LDAP style algorithm prefix and the password encrypted with the algorithm specified in the prefix. Example: <code>{PLAIN}badPassword</code>
<code>setUserPasswordStatement</code>	<p>Required non-empty for <i>changePassword</i> feature.</p> <p>SQL template for a prepared statement to set a new password for the user.</p> <p>Default, appropriate for SDS' tables: <code>UPDATE USERS SET PASSWORD=? WHERE USERID=?</code></p>
<code>userByUserIdQuery</code>	<p>SQL template for a prepared statement to query <code>userId</code>, encrypted password, pretty name and e-mail address of a user specified by a <code>userId</code>.</p> <p>If the pretty name is not part of the database, reuse the <code>userId</code> field.</p> <p>If the e-mail address is not part of the database, use a constant like <code>"</code> or <code>null</code>.</p> <p>Example: <code>SELECT USERID, PASSWORD, USERNAME, EMAIL FROM USERS WHERE USERID=?</code></p>

usersByGroupId Query	<p>Required non-empty for <i>getGroupMembers</i> feature.</p> <p>SQL template for a prepared statement to query the user IDs, user names and e-mail-addresses of the members of the group specified by a groupId.</p> <p>Example: SELECT u.USERID, u.USERNAME, u.EMAIL FROM GROUPS AS g, USERS AS u, GROUPMEMBERSHIPS AS m WHERE g.GROUPID=? AND m.GROUP_ID = g.ID AND u.ID = m.USER_ID</p>
-------------------------	--

Database access configuration

These becomes totally obsolete with version 1.2.

Key	Description
jdbcPassword	Should be self explanatory 😞
jdbcUrl	For examples look at our JDBC Snippets
jdbcUser	Should be self explanatory 😞
validationQuery	For proposals look at our JDBC Snippets