

Swivel Core V4 Database Menu

Contents

- 1 General
- 2 Shipping
- 3 Internal
- 4 Connection Pool

General

swivelsecure

Status

Log Viewer

Server

Policy

Logging

Messaging

Database

General

Connection Pool

Appliance Database

Mode

Repository

RADIUS

Migration

Appliance

OATH

Config Sync

Reporting

User Administration

Save Configuration

Upload Email Images

Administration Guide

Logout

Database: Appliance Database

Case sensitive usernames: No

Databases:

Shipping

Internal

Appliance Database

JDBC

MS SQL Server

MySQL 5

Identifier: MySQL 5

Class: com.swiveltechnologies.pinsafe.server.user.database.MySQL5D

Driver: org.mariadb.jdbc.Driver

URL: jdbc:mysql://localhost/pinsafe

Username: pinsafe

Password:

Delete

Oracle 10g

New Entry

Apply Reset

This screen allows you to determine where the Swivel data is to be stored. You can use the in-built Swivel database or any compatible external database. If you are unsure about the compatibility of a database that you wish to use, contact Swivel Secure for more information. There are two special cases for the database that is the shipping database and the internal database; these are described below. At the time of launch Swivel was compatible with MS-SQL, MySQL, PostGres and Oracle databases. If you purchased Swivel as a HA pair (Active-Active) the MySQL database will be pre-configured with the required settings and the drivers will be already loaded

- **Database:** Select the database you wish to use from this drop down list. If you add a new database to this screen it will appear on this list as an option.
- **Case Sensitive Usernames:** If you set this to Yes, then to successfully authenticate the username provided as part of the credentials must match the case of the username as stored in the user-repository

Databases:

These fields specify the various possible databases that are available. Some flavours of database are supported "out of the box" others can be added.

To configure a database you need to specify the following parameters

- **Identifier:** Friendly name to uniquely identify the database
- **Class:** Name of the Java class that implements connectivity to the repository, usually pre-filled
- **Driver:** This is the java driver that Swivel will use to connect with the database. This needs to be copied to the apache-tomcat directory where it can be accessed by Swivel, eg under the WEB-INF\lib directory. You may need to restart tomcat to allow Swivel to load the driver
- **URL:** The url of the database. The exact format of the url may depend on the specific database, but a typical example would be jdbc:mysql://sql.server.local:3306/pinsafe

The /pinsafe being the name of the database. This database needs to be created by the DBA prior to configuring PINsafe.

- **Username/Password:** These are the credentials Swivel will use to authenticate against the database

Shipping

This is a special case of the database; this is the database with which Swivel ships. The database is not a real database but holds one single account with the username admin and a PIN of 1234. This account is not editable so setting the database to shipping will always mean you can log on with these credentials

Internal

The internal database is a built in database configured and ready to use. No configuration of the database is required. The only disadvantage of the database is that the data is internal to the Swivel server and cannot be accessed by other applications or Swivel servers.

Connection Pool

Status

Log Viewer

▸ Server

▸ Policy

▸ Logging

▸ Messaging

▾ Database

> General

> **Connection Pool**

> Appliance Database

▸ Mode

▸ Repository

▸ RADIUS

▸ Migration

▸ Appliance

▸ OATH

▸ Config Sync

▸ Reporting

User Administration

Save Configuration

Upload Email Images

Administration Guide

Logout

Database / Connection Pool ?

Configure a connection pool for the database in use. A connection pool will speed up operation and

Enable Connection Pool:

No ▾

Maximum Pool size:

5

Connection test statement:

Connection test timeout (s):

5

Test on borrow:

No ▾

Test while idle:

No ▾

Connection test interval (s):

300

Apply

Reset

- **Enable Connection Pool:** Allows to enable/disable the connection pool
- **Maximum Pool size:** Specifies the maximum number of connections to be made to the database
- **Connection test statement:** This is a SQL statement which is sent to the database server at intervals to keep the connection alive. It therefore needs to be a statement that takes little processing time. "SELECT 1" is used by SQL Servers. When a test statement is set either Test on borrow or Test while idle should be set.
- **Test on borrow:** If set to Yes, then the test statement will only be sent when a connection is requested. If there has not been a connection in a while, then it may time out and produce an error log message. This is not required if Test on idle is set to Yes. Test on idle is more efficient, but you then have the possibility that the connection fails between the last time it was tested and the time it's actually used.
- **Test while idle:** If set to Yes, then the statement will be sent at regular intervals (as set by the Connection test interval) for all connections that are not currently active. This should mean that connections will never time out (unless you set the test interval too long), but will result in more traffic. This is not required if Test on borrow is set to Yes. Test on borrow has more overhead, because you execute an extra SQL command every time, rather than at the set time.
- **Connection test interval (s):** How often to run test, default 300 seconds