Prof. DI Dr. Erich Gams

Datenbankverbindung mit JNDI, Connection pooling

JDBC

informationssysteme htl-wels

Übersicht ☛ Was lernen wir?

- Verbindung über JNDI
- Beispiele
- Aufgabe

DataSource

- Configuration data in a centralized repository

```
Context ctx=new InitialContext();
DataSource ds=(DataSource) ctx.lookup(“jdbc/database“)
Connection con=ds.getConnection(“username“,“password“);
```

DataSource

- DataSource-Object is integrated in the container (e.g. Tomcat)
- SimpleJNDI: name service that serves data centralized and does not require a server (https://code.google.com/archive/p/osjava/downloads?page=3)
- simple-jndi-0.11.4.1.jar

DataSource

- src/jndi-properties

  java.naming.factory.initial=org.osjava.sj.SimpleContextFactory

  org.osjava.sj.root=config/

- config Folder -> database.properties

  type=javax.sql.DataSource

  driver=org.hsqldb.jdbcDriver

  url=jdbc:hsqldb:file:TestDB;shutdown=true

  user=sa

  password=

DataSource - Zugriff

```
DataSource ds=(DataSource) new InitialContext().lookup(“TestDB“);

con=ds.getConnection();
```

Connection Pooling

- Opening a connection to a database is a time-consuming process -> especially for short queries.
- Consequently, it makes sense to reuse Connection objects in applications that connect repeatedly to the same database.
- *Connection pooling* means that connections are reused rather than created each time a connection is requested.

Connection Pooling Tasks

› A connection pool class should be able to perform the following tasks:

1. Preallocate the connections.
2. Manage available connections.
3. Allocate new connections.
4. Wait for a connection to become available.
5. Close connections when required

JDBC Connection Pooling Frameworks

- Apache Commons DBCP
  - commons-dbcp2 package
  - commons-pool2 package
  - database.properties: pool=true
- HikariCP
- C3PO

Apache Commons DBCP

```
public class DBCPDataSource {

    private static BasicDataSource ds = new BasicDataSource();

    static {
        ds.setUrl("jdbc:h2:mem:test");
        ds.setUsername("user");
        ds.setPassword("password");
        ds.setMinIdle(5);
        ds.setMaxIdle(10);
        ds.setMaxOpenPreparedStatements(100);
    }

    public static Connection getConnection() throws SQLException {
        return ds.getConnection();
    }

    private DBCPDataSource(){ }
}
```

Apache Commons DBCP

- Get the Connection:

```
Connection con = DBCPDataSource.getConnection();
```

Auf los geht‘s los ;-)

Aufgabe

- database: song administration
  - songs: songtitle, artist, year, genre
  - album_songs: albumtitle, songstitle, tracknumber
  - album: albumtitle, artist, label
- This data should be outsorced. (Class: Properties)

```
Driver=org.hsqldb.jdbcDriver
url=jdbc:hsqldb:file:.\hsqldb\data/;shutdown=true
User=SA
Password=
```

Sources

- https://www.baeldung.com/java-connection-pooling
- https://examples.javacodegeeks.com/core-java/apache/commons/dbcp/dbcp-connection-pooling-example/