

# Simplified DAO layer (database) testing

Author: Artūr Girenko

2012-05-09



# Benefits of DAO testing

- Proven correctness of queries, mapping and persistence logic
- Ease of refactoring or optimizing queries
- Working example of DAO API usage
- THE confidence :)



# Simple DAO test is..

- Code of the test is simple to read and understand – it contains nothing else but example of DAO usage
- No repeated complex technical code
- No structural difference (like mandatory inheritance) from other (non DAO) tests
- 5 min to configure DAO testing in project and write a simple test that works



# The stack

- JUnit
- DBUnit
- HSQLDB
- Spring-test (optional)



# Database test flow

- Create database schema (DDL)
- Load a dataset into database
- Run DAO operations
- Compare the dataset in database with expected dataset



# DEMO

2012-05-09



# Why not just DBUnit?

- Lacks some needed methods in `IDatabaseTester`
- No single interface to use – need many DBUnit classes
- `IDatabaseTester` is quite similar to subclassing `TestCase` with `onSetup()`, `onTearDown()` - with no clue what happens there



# Use in-memory HSQLDB

- No need to think about FS at all
- Not state transfer between test runs (no problems with crashed tests and lock files)
- Should be quicker (in theory)





# HSQLDB Database syntax feature

- Useful in native SQL DAOs (JdbcTemplate, MyBatis, plain JDBC) – Native, vendor specific syntax may be used.
- Demo with Oracle style



# Skipping columns during verify

- Solves problems with hard-to-predict auto-generated values
- Demo with creating new objects



# Disable FK's when loading dataset

- Constraints are good, but you will have to think about order in your dataset too much.
- Developer responsible to create valid dataset
- FK's should be enabled during test run



# Using spring test demo

- Usually the case with DAO layer
- Reusing common configs
- Use as much as possible production config
- TestExecutionListener is good for hiding technical stuff



# Transactional tests

- Needed for testing JPA lazy loaded collections
- Forced to use `@Rollback(false)`



# Resume

Creating a facade for hiding testing-related code and reusing it, makes your tests much more simpler



# Q&A

2012-05-09

