



Squill: Not Another ORM



Jevgeni Kabanov

Lead Rebel
ZeroTurnaround

Squill co-founder
<http://squill.dev.java.net>

What is Squill?

- **An embedded Java DSL for expressing SQL queries**
- Typesafe: no strings are used anywhere, everything is generic
- Schema-driven: database metadata used to generate classes describing the tables and matching data objects
- Transparent: Selection and joins are explicit, no hidden queries ever generated
- Lightweight ORM: CRUD, selecting whole data objects and relations are easy
- Specific: SQL/database extensions are supported

What Squill is Not

- **Not an ORM replacement**
- Data objects cannot be used for rich domain design
- Field mapping is not supported (data objects are one-to-one with the database)
- Relationship navigation after the query is not supported
- Object-with-relations selection is not supported
- Caching, versioning and advanced entity management has to be done externally
- Not portable by default (but you can choose to be)

Why Squill?

1. To replace usual SQL

- Every bigger project has a ton of SQL written to improve/guarantee performance
- Squill has same expressivity, but is shorter and typesafe

2. To use if ORM is too heavy for you

- ORM abstracts from the database, which sometimes can compromise performance (selection, joins, hidden queries)
- Squill can be more productive thanks to type safety and autocompletion

SQL Example (4 errors)

```
StringBuffer sql = new StringBuffer();
sql.append("SELECT o.sum, (SELECT
first_name, last_name");
sql.append("                FROM person p");
sql.append("                WHERE o.person_id=p.id) AS
client");
sql.append(" FROM order o");
sql.append("WHERE o.id = "+orderId);
sql.append(" AND o.status_code IN (?, ?)");
PreparedStatement stmt =
conn.prepareStatement(sql.toString());
stmt.setString(1, "PAID");

//...
```


Squill Example

```
PersonTable p = new PersonTable();
List<Tuple3<String, Integer, Date>> rows =
    squill
        .from(p)
        .where(gt(p.height, 170))
        .selectList(p.name, p.height, p.birthday)

for (Tuple3<String, Integer, Date> row : rows) {
    String name = row.v1;
    Integer height = row.v2;
    Date birthday = row.v3;
    System.out.println(
        name + " " + height + " " + birthday);
}
```

DEMO

Let's port Spring PetClinic to
Squill



Thanks for your attention!

<http://squill.dev.java.net>

Come see “Zero Turnaround in Java”
on Thursday at 17:50 in Room 7
or visit the ZeroTurnaround booth