

Hibernate

The Ubiquitous ORM

Michał Żmuda

Piotr Turek

Why Hibernate?

Hibernate is the choice for you because:

- Is a bridge between OO World and Relational DBs
- Is everywhere
- Is robust and performant
- Is used in enterprise programming
- Better know it before going to a job interview

Object-Relational Mapping

Solves the problem of translating *object* definitions to database *tables* and vice versa

Why are Relational DBs so ubiquitous?

- Efficient operations on large amounts of data
- Well defined transaction mechanisms
- ACID
- Complex queries
- Relations between tables
- *Huge legacy*

Why are Objects the way to go?

- Resemble the way we reason about world
- Most enterprise languages and frameworks are Object-Oriented
- Systems easier to model using OO paradigm

What Hibernate is

- Multilanguage ORM framework
- Maps POJO to Relations
- DB-independent - you provide driver only
- Introduces own query language HQL
- Simplifies reading/writing objects from db
- Supports inheritance (!)
- mutual compatibility with JPA

Environment

- java 1.7 SE
- postgres
- maven
- maven hibernate dependency
- basic Spring

thats all !!!

How to make magic?

- mapping definition either by
 - annotating
 - xml configuration
- database connection
- *SessionFactory*

How to get SessionFactory?

- Programmatically

```
Configuration configuration = new Configuration();  
configuration.configure();  
  
ServiceRegistry serviceRegistry =  
    new ServiceRegistryBuilder().  
        applySettings(configuration.getProperties()).  
        buildServiceRegistry();  
  
SessionFactory sessionFactory =  
    configuration.buildSessionFactory(serviceRegistry);
```

or...

How to get SessionFactory?

- Declaratively

```
<bean id="sessionFactory" class="org.springframework.orm.hibernate4.LocalSessionFactoryBean">
  <property name="dataSource" ref="dataSource"/>
  <property name="packagesToScan" value="pl.agh.turek.bazy.hibernate"/>
  <property name="hibernateProperties">
    <props>
      <prop key="hibernate.dialect">org.hibernate.dialect.PostgreSQL82Dialect</prop>
      <prop key="hibernate.show_sql">true</prop>
      <prop key="hibernate.hbm2ddl.auto">validate</prop>
      <prop key="hibernate.cache.use_second_level_cache">true</prop>
      <prop key="hibernate.cache.provider_class">org.hibernate.testing.cache.CachingRegionFactory</prop>
      <prop key="hibernate.cache.region.factory_class">org.hibernate.testing.cache.CachingRegionFactory</prop>
    </props>
  </property>
</bean>
```

Entry level course agenda

1. creating hibernate & spring maven-based project
2. connecting to the db
3. schema<->model generation
4. overview of object states
5. fetching types
6. accessing SessionFactory
7. read/write methods
8. association types (one-to-many, many-to-many ...)
9. extending model

Advanced course agenda

1. transactions
 - a. ACID
 - b. propagation levels
2. review of JPA compatibility
3. common problems
 - a. eager fetching performance
 - b. lazy init
 - c. caching
4. good practices