spring

HIBERNATE

SUNBEAM

Trainer: Nilesh Ghule

Wake up from Hibernate, Spring up!!!

Course Introduction

- What is prerequisite of the course?
  - Java – OOP, Collections, Reflection, Annotation & Proxy.
  - Development – JUnit & Maven.
  - RDBMS/SQL – JDBC
  - HTML, JS, CSS – Servlets, JSP
- What should I expect from this course?
  - In depth knowledge of Hibernate/JPA (5.4.17), Spring (5.2.7)
  - Introduction to Spring Boot.
  - Hands-on experience in Hibernate and Spring.
- What is NOT covered in this course?
  - Core Java, JVM internals, SQL queries, HTML/JS coding
  - Spring Boot Micro-services and Spring Cloud

Course Introduction

- Course schedule
  - Weekdays: Monday to Friday ✓
  - Time: 8:00 AM to 11:00 AM ✓
- Resources
  - Source codes & slides: gitlab.com repository: https://gitlab.com/nilesh-g/sh-09 ✓
    - Repository will be private after this session. → profile - edit - gitlab
    - Update your gitlab id into student portal: students.sunbeamapps.org 2:00PM
  - Recorded videos will be available under students.sunbeamapps.org
    - Videos are provided only considering network issues / office work hours.
    - Video is accessible from a single device for 7 days (from date of uploading).
- Hands-on
  - Try to code along with trainer during the session.
  - Try assignments (homework) for gaining more confidence.
  - Lab mentors: Mr. Yogesh Kolhe and Mr. Akash Shelke.

Agenda

- Java EE Overview
- Mini-Project Idea
- Object Relational Mapping
- Hibernate Introduction
- Hibernate Architecture
- Hibernate Configuration
- Hibernate3 Bootstrapping
- ORM using Annotation
- ORM using XML file
- CRUD operations & Transactions
- openSession() vs getCurrentSession()

JDBC Quick Revision

- JDBC is specification given by Sun/Oracle.
- Specification interfaces are implemented by driver.
  - Driver, Connection, Statement, ResultSet
- JDBC driver convert Java request to RDBMS understandable form and RDBMS response to Java understandable form.
- JDBC programming steps
  - Add JDBC driver into project CLASSPATH.
  - Load and register JDBC driver.
  - Create JDBC connection.
  - Prepare JDBC statement.
  - Execute query and process result.
  - Close all.

JDBC Quick Revision

- Transaction is a set of DML queries executed as a single unit. If any query fails, other queries are discarded.
- Transaction is feature of RDBMS.
- RDBMS commands: START TRANSACTION, SAVEPOINT, COMMIT, ROLLBACK.
- It follows ACID properties: Atomicity, Consistency, Isolation and Durability.
- JDBC functions ↗ default true
  - ✓ con.setAutocommit(false);
  - ✓ con.commit();
  - ✓ con.rollback();

```
try {
    con.setAutocommit(false);        begin tx
    // exec DML statements  ☰
    con.commit();    commit
} catch(Exception e) {
    con.rollback();    rollback
}
```

Java EE Overview

javase

tomcat ⟷ Hibernate + Spring → tomcat 9.x

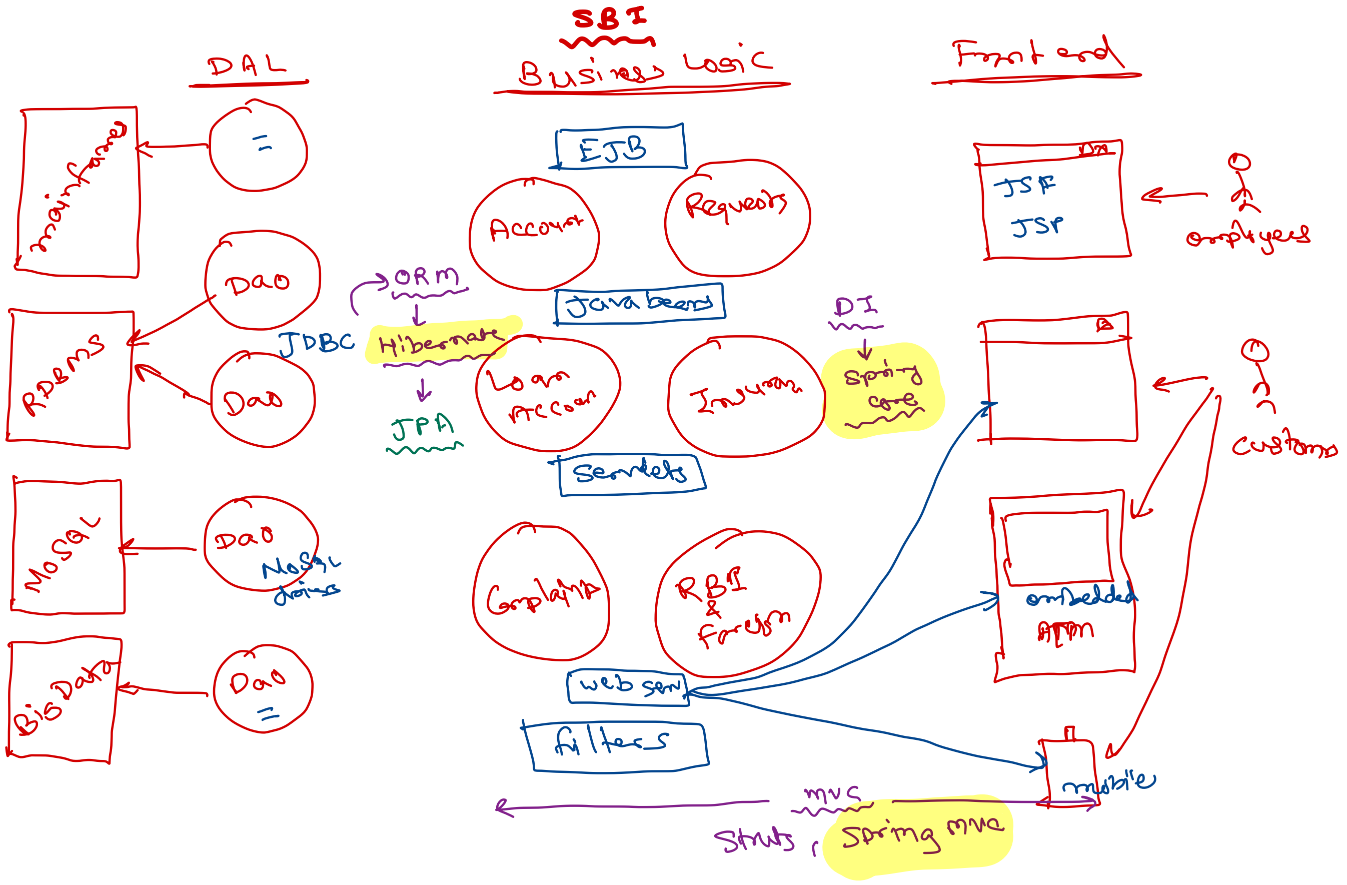
- Java EE is specification by Sun/Oracle.
- Java EE 8 specification includes
  - Servlet 4.0, WebSocket 1.1.
  - JSP 2.3, EL 3.0, JPA 2.2.
  - JSF 2.3, CDI 1.0.
  - Common Annotations 1.3.
  - JMS 2.1, EJB 3.2, JTA 1.2.

→ web servers

+ ejb

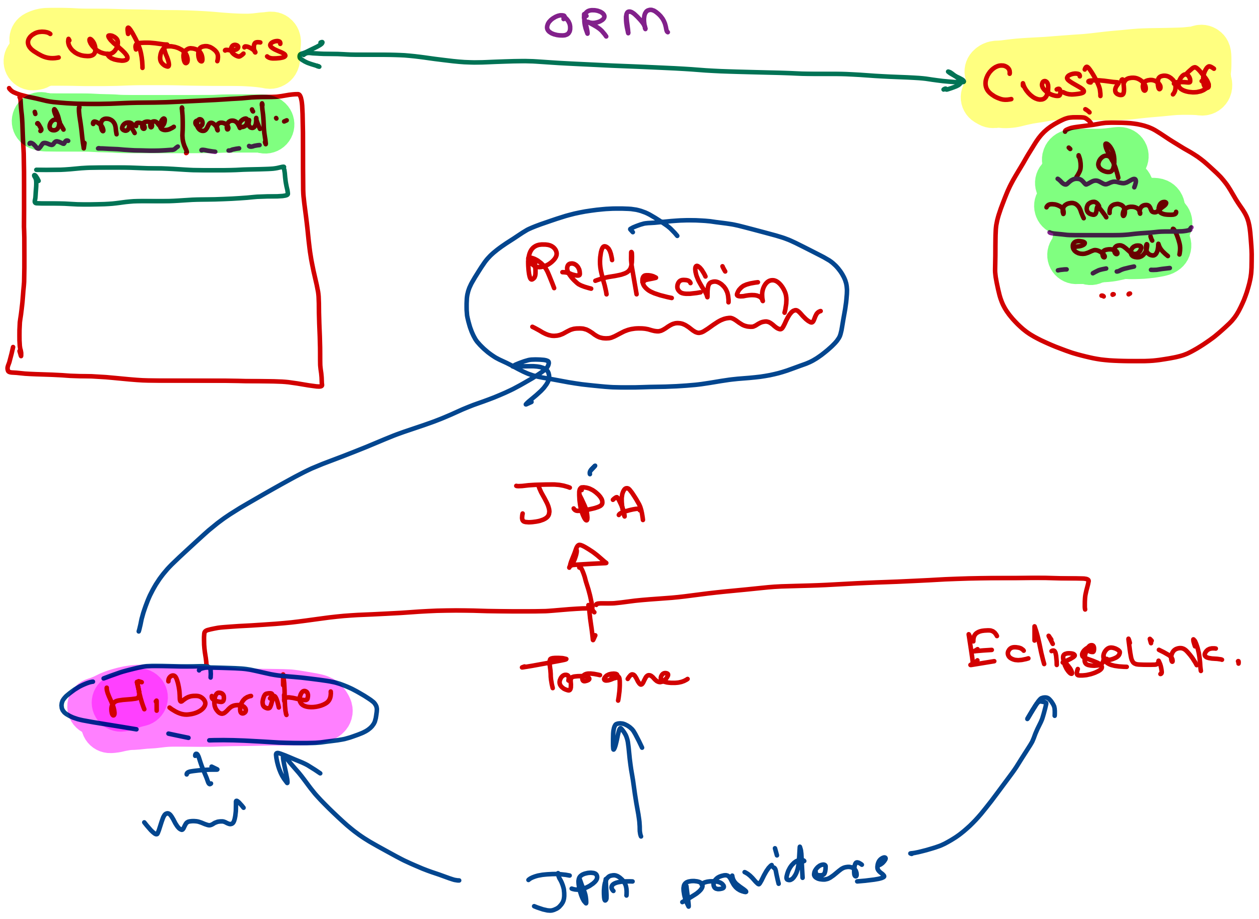
= → appln servers

- Java EE specification is implemented by Java web servers and application servers like Tomcat, JBoss, WebSphere, etc.

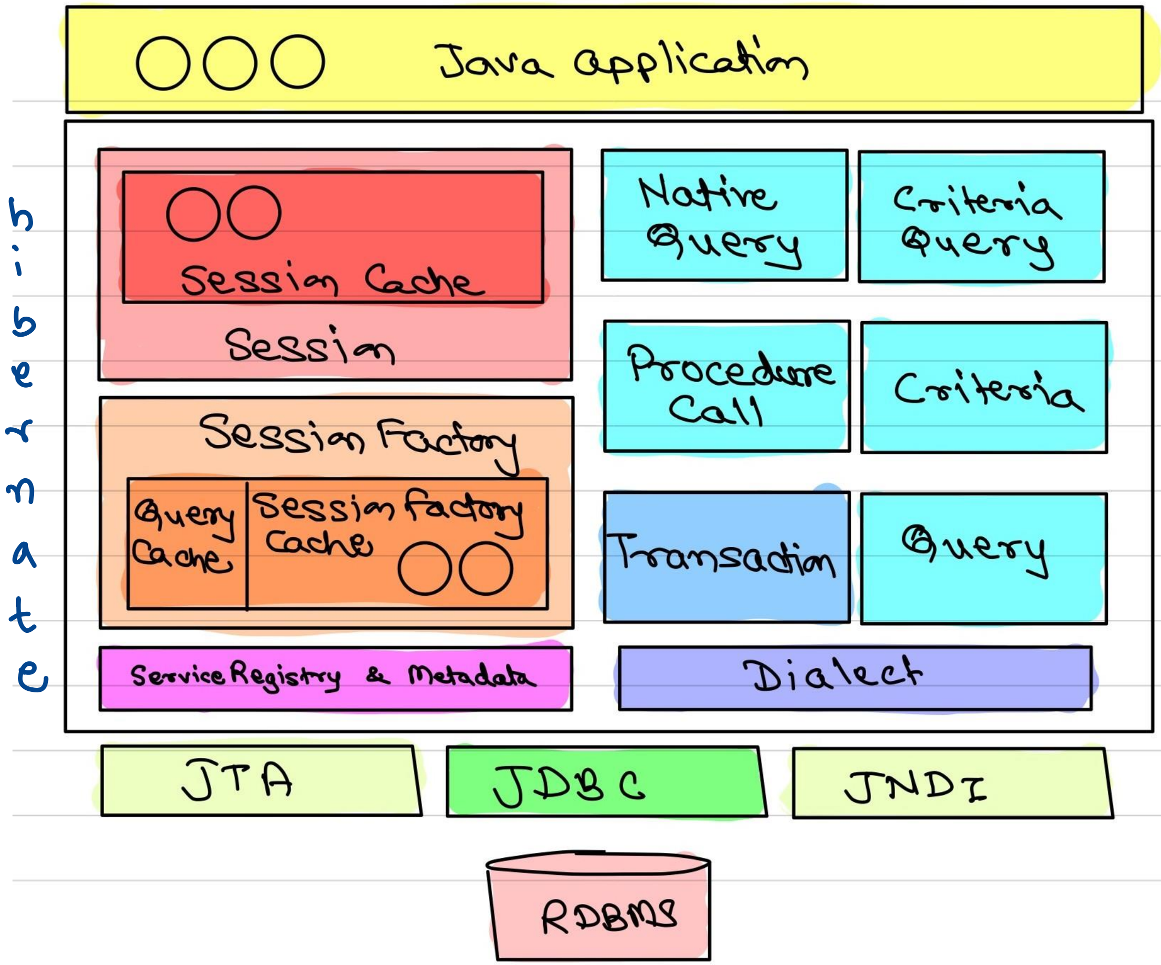


Object Relational Mapping

- Converting Java objects into RDBMS rows and vice-versa is done manually in JDBC code.
- This can be automated using Object Relational Mapping.
- Class → Table and Field → Column
- It also map table relations into entities associations/inheritance and auto-generates SQL queries.
- Hibernate is most popular ORM tool.
- Other popular ORM are EclipseLink, iBatis, Torque, ...
- JPA is specification for ORM.

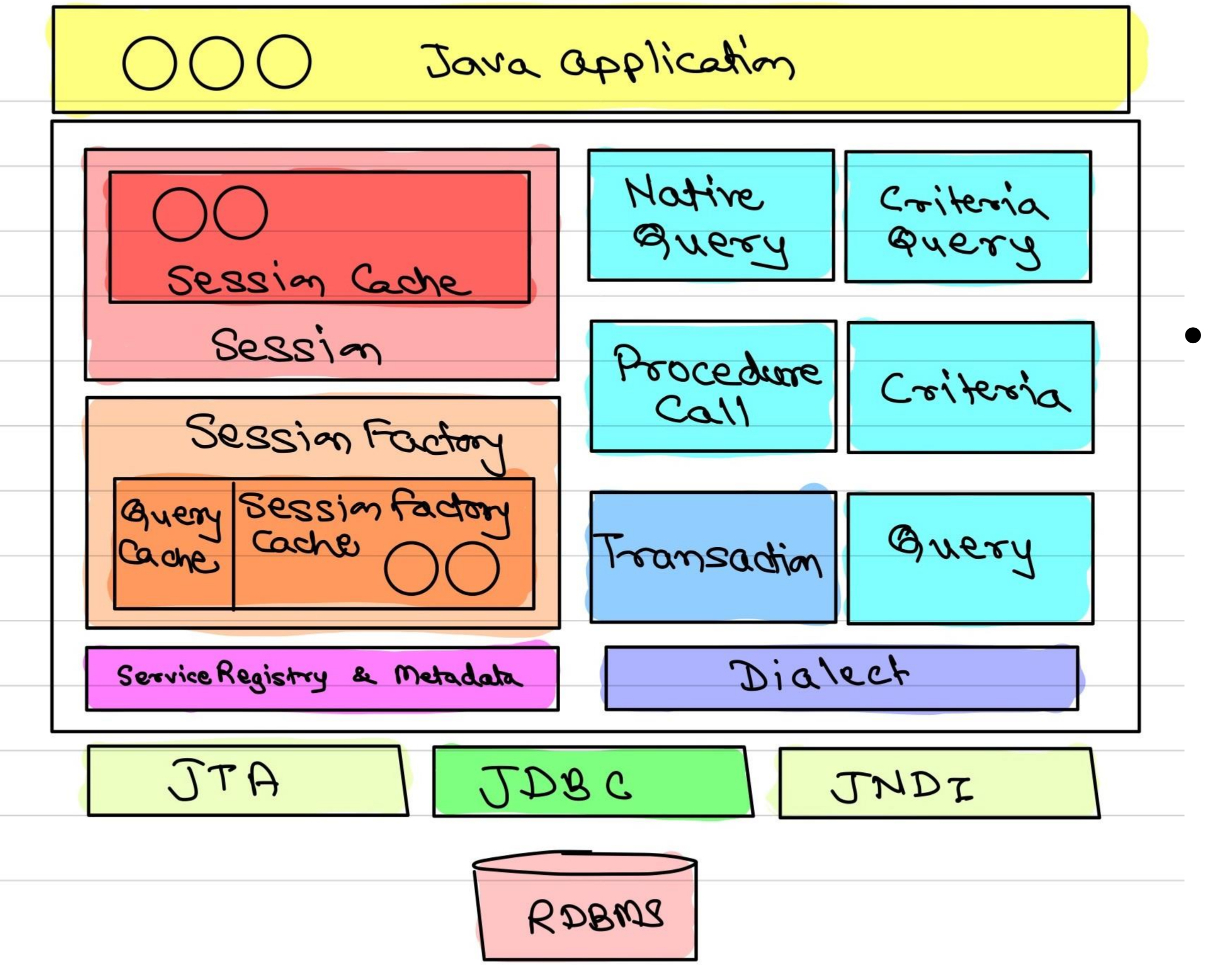


Hibernate



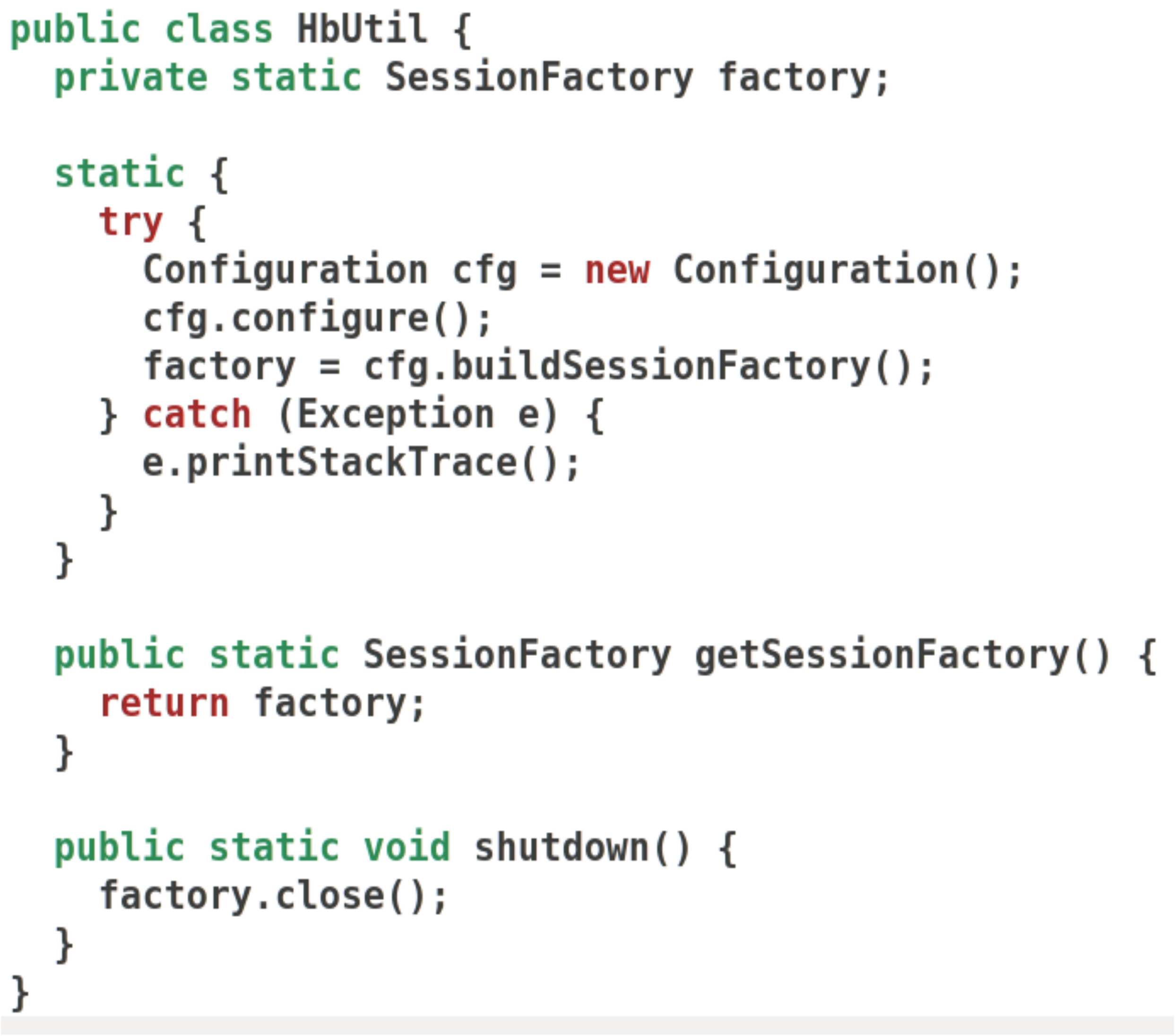
- SessionFactory
  - One SessionFactory per application (per db).
  - Heavy-weight object. Not recommended to create multiple instances.
  - Thread-safe. Can be accessed from multiple threads (synchronization is built-in).
  - Typical practice is to create singleton utility class for that.
- Session
  - Created by SessionFactory & it encapsulate JDBC connection.
  - All hibernate operations are done on hibernate sessions.
  - Is not thread-safe. Should not access same session from multiple threads.
  - Light-weight. Can be created and destroyed as per need.

Hibernate



- Transaction
  - In hibernate, autocommit is false by default.
  - DML operations should be performed using tx.
  - session.beginTransaction(): to start new tx.
  - tx.commit() & tx.rollback(): to commit/rollback tx.
- Dialect
  - RDBMS have specific features like data types, stored procedures, primary key generation, etc.
  - Hibernate support all RDBMS.
  - Most of code base of Hibernate is common.
  - Database level changes are to be handled specifically and appropriate queries should be generated. This is handled by Dialect.
  - Hibernate have dialects for all RDBMS. Programmer should configure appropriate dialect to utilize full features of RDBMS.

Hibernate



```java
public class HbUtil {
  private static SessionFactory factory;

  static {
    try {
      Configuration cfg = new Configuration();
      cfg.configure();
      factory = cfg.buildSessionFactory();
    } catch (Exception e) {
      e.printStackTrace();
    }
  }

  public static SessionFactory getSessionFactory() {
    return factory;
  }

  public static void shutdown() {
    factory.close();
  }
}
```

- Hibernate Configuration
  - hibernate.connection.driver_class
  - hibernate.connection.url
  - hibernate.connection.username
  - hibernate.connection.password
  - hibernate.dialect
  - hibernate.show_sql
- Hibernate3 Bootstrapping
  - Create Configuration object.
  - Read hibernate.cfg.xml file using its configure() method.
  - Create SessionFactory using its buildSessionFactory() method.

Hibernate

- Hibernate3 added annotations for ORM.

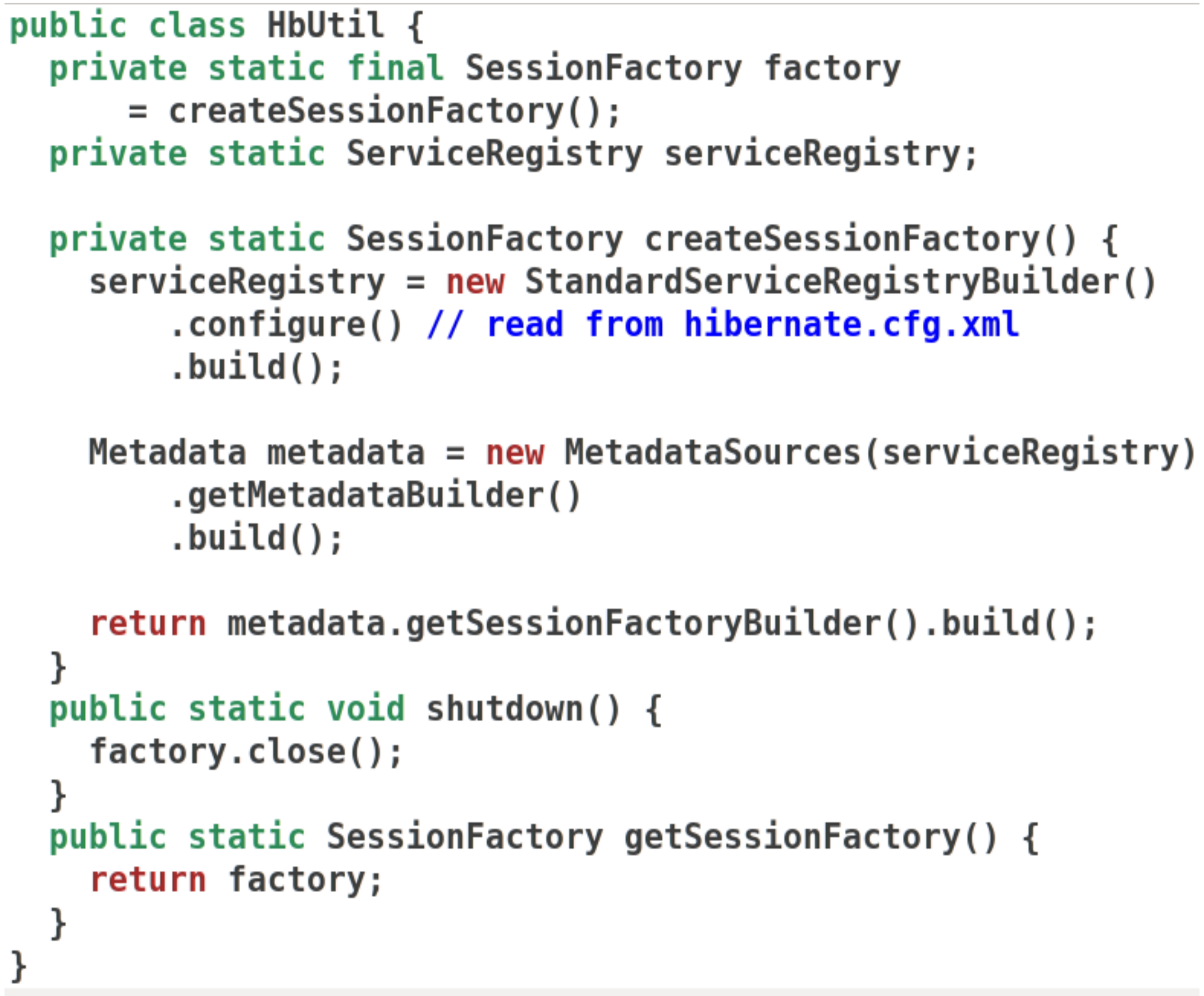
- ORM using annotations
  - @Entity
  - @Table
  - @Column
  - @Id
  - @Temporal
  - @Transient

- @Column can be used on field level or on getter methods.

CRUD operations

- Hibernate Session methods
  - get()
  - load()
  - find()
  - save()
  - persist()
  - update()
  - saveOrUpdate()
  - merge()
  - delete()
  - remove()
- Hibernate transactions
  - tx = session.beginTransaction()
  - tx.commit()
  - tx.rollback()

Hibernate



```
public class HbUtil {
  private static final SessionFactory factory
      = createSessionFactory();
  private static ServiceRegistry serviceRegistry;

  private static SessionFactory createSessionFactory() {
    serviceRegistry = new StandardServiceRegistryBuilder()
        .configure() // read from hibernate.cfg.xml
        .build();

    Metadata metadata = new MetadataSources(serviceRegistry)
        .getMetadataBuilder()
        .build();

    return metadata.getSessionFactoryBuilder().build();
  }
  public static void shutdown() {
    factory.close();
  }
  public static SessionFactory getSessionFactory() {
    return factory;
  }
}
```

- Hibernate 5 Bootstrapping
  - Create ServiceRegistry.
  - Create Metadata.
  - Create SessionFactory.
- ServiceRegistry
  - ServiceRegistry is interface.
  - Some implementations are StandardServiceRegistry, BootstrapServiceRegistry, EventListenerRegistry, ...
  - Add, manage hibernate services.
- Metadata
  - Represents application’s domain model & its database mapping.

SUNBEAM

Thank you!

Nilesh Ghule <nilesh@sunbeaminfo.com>