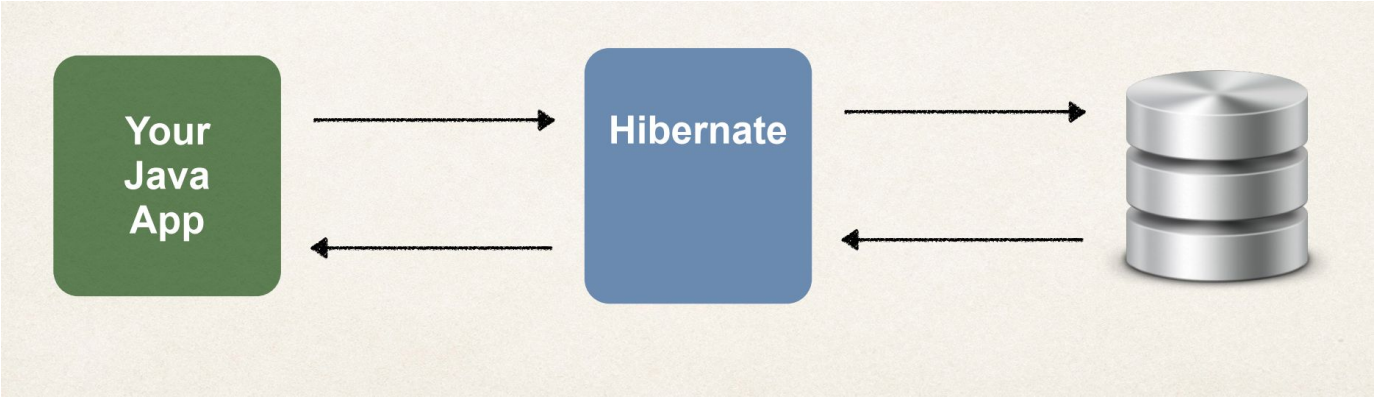
HIBERNATE

Overview

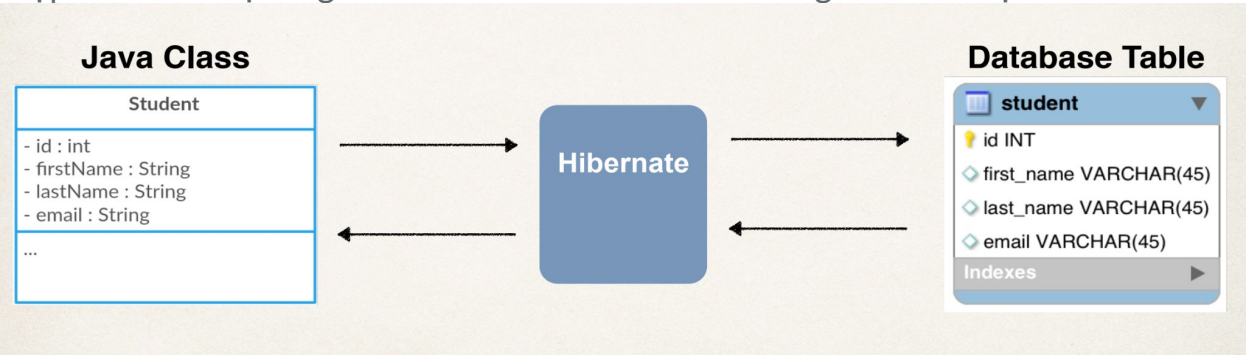
Hibernate là gì

- Là 1 framework dùng để lưu trữ (persisting/saving) các đối tượng Java Objects trong cơ sở dữ liệu
- www.hibernate.com



Hibernate

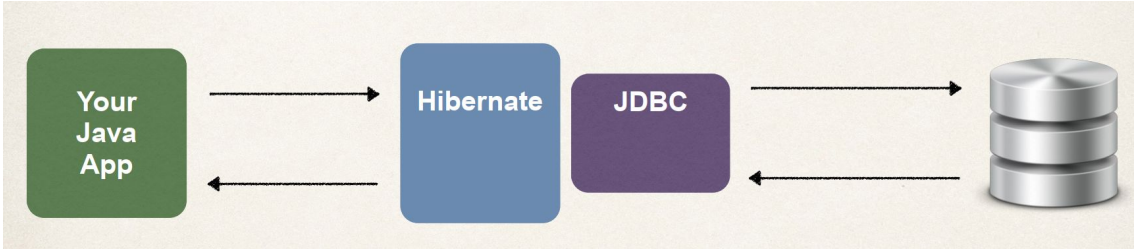
- Hibernate quản lý tất cả các mức độ truy cập cấp thấp nhất của SQL
- Hibernate tối ưu đến mức nhỏ nhất tài nguyên truy cập JDBC khi bạn lập trình
- Cung cấp cơ chế kết nối ORM (Object to Relational Mapping)
  - Lập trình viên định nghĩa và kết nối Java Class với bảng cơ sở dữ liệu



- Hibernate sử dụng ngôn ngữ hql để truy vấn
  - Hibernate Query Language (HQL) là một ngôn ngữ truy vấn hướng đối tượng (OOP), tương tự như SQL (Structured Query Language) nhưng thay vì làm việc trên các bảng và cột, HQL làm việc với các đối tượng persistent và các thuộc tính của chúng

Hibernate

- Hibernate sử dụng JDBC để kết nối giao tiếp cho tất cả các cơ sở dữ liệu quan hệ



- Yêu cầu khi sử dụng Hibernate
  - Hibernate JAR và JDBC Driver
  - Cơ sở dữ liệu

```
<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>5.6.3.Final</version>
</dependency>
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.16</version>
</dependency>
```

Hibernate

- Lưu trữ Java Object với Hibernate

```
// create Java object
Student theStudent = new Student("John", "Doe", "john@luv2code.com");

// save it to database
int theId = (Integer) session.save(theStudent);
```

- Lấy Java Object

```
// create Java object
Student theStudent = new Student("John", "Doe", "john@luv2code.com");

// save it to database
int theId = (Integer) session.save(theStudent);

// now retrieve from database using the primary key
Student myStudent = session.get(Student.class, theId);
```

Hibernate

- Truy vấn Java Objects với Hibernate

```
Query query = session.createQuery("from Student");

List<Student> students= query.list();
```

- Sử dụng Hql để truy vấn
  - HQL không phân biệt hoa thường (giống với ngôn ngữ SQL)
  - From trong HQL cũng tương tự như SQL và chúng ta cũng có thể đặt alias cho nó ví dụ như mình muốn lấy ra tất cả các sản phẩm của Product câu lệnh sẽ là: Select Product p tương đương với câu lệnh select * from Product as p trong SQL
  - HQL hỗ trợ tất cả các loại Join có trong SQL như là Left Join, Right Join, Inner Join, Full Join,...
  - HQL hỗ trợ các hàm thực hiện các phép toán như count, max, min, sum,...
  - Chúng ta cũng có thể sử dụng các biểu thức so sánh trong HQL tương tự như là SQL
  - Hỗ trợ câu lệnh order by để sắp xếp, Group By để phân nhóm

Hibernate HQL

- FROM
- X
- AS
- X
- SELECT
- X
- WHERE
- X
- ORDER BY
- X
- GROUP BY
- X
- THAM SỐ
-

```
String hql = "FROM Employee";
Query query = session.createQuery(hql);
List results = query.list();
```

```
String hql = "FROM Employee AS E";
Query query = session.createQuery(hql);
List results = query.list();
```

```
String hql = "SELECT E.firstName FROM Employee E";
Query query = session.createQuery(hql);
List results = query.list();
```

```
String hql = "FROM Employee E WHERE E.id = 10";
Query query = session.createQuery(hql);
List results = query.list();
```

```
String hql = "FROM Employee E WHERE E.id > 10 ORDER BY E.salary DESC";
Query query = session.createQuery(hql);
List results = query.list();
```

```
String hql = "SELECT SUM(E.salary), E.firtName FROM Employee E " +
             "GROUP BY E.firstName";
Query query = session.createQuery(hql);
List results = query.list();
```

```
String hql = "FROM Employee E WHERE E.id = :employee_id";
Query query = session.createQuery(hql);
query.setParameter("employee_id",10);
List results = query.list();
```

Spring MVC - Hibernate

- Công nghệ và công cụ
  - Spring MVC
  - Hibernate
  - JDK
  - Maven
  - Apache Tomcat
  - Eclipse
  - JSTL
  - JSP
  - MySQL

Spring MVC - Hibernate

- Các bước phát triển
  - Create a Maven Web Application
  - Add Dependencies - pom.xml File
  - Project Structure
  - AppInitializer - Register a DispatcherServlet using Java-based Spring configuration
  - AppContext - Spring and Hibernate Integration using Java-based Spring configuration
  - WebMvcConfig - Spring MVC Bean Configuration using Java-based Spring configuration
  - Entity - Customer.java
  - Spring MVC Controller Class - CustomerController.java
  - Service Layer - CustomerService.java and CustomerServiceImpl.java
  - DAO Layer - CustomerDAO.java and CustomerDAOImpl.java
  - JSP Views - customer-form.jsp and list-customers.jsp
  - Server Static Resources - CSS and JS
  - Build and Run an application

Spring MVC - Hibernate

Add Dependencies

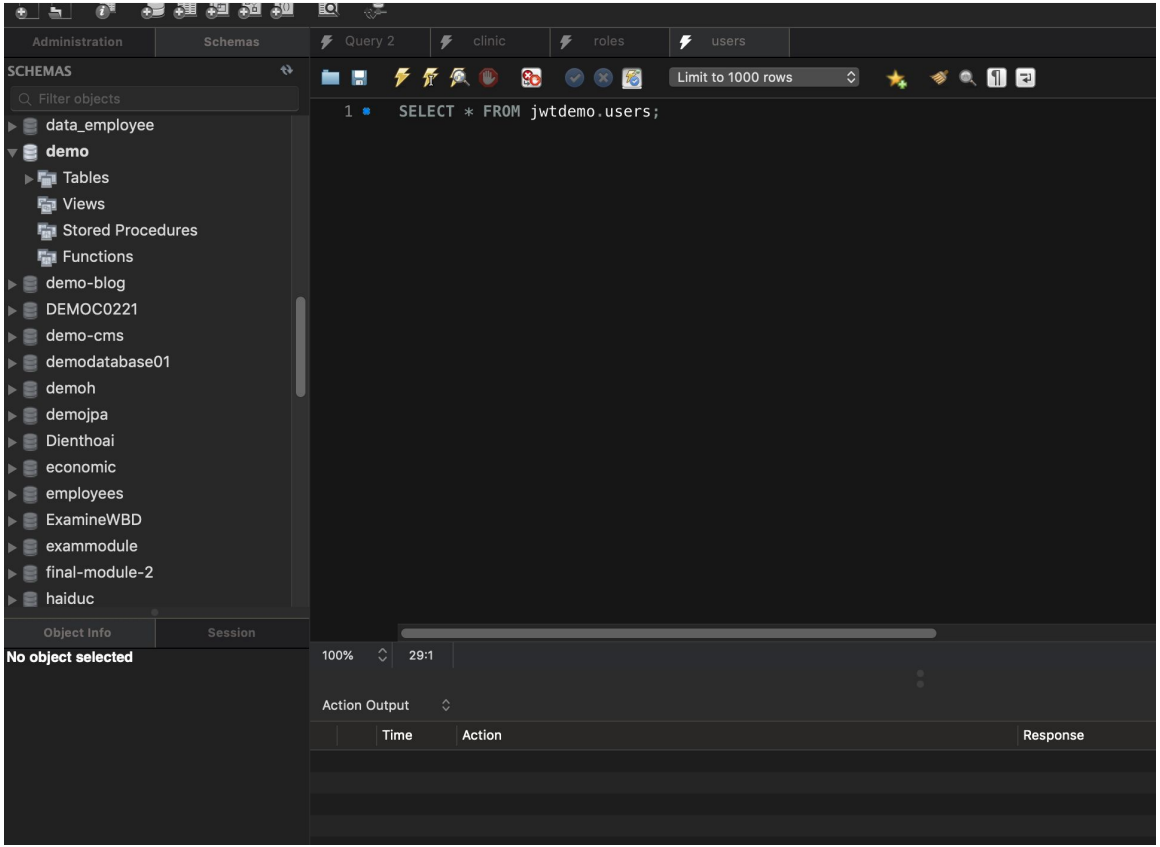
pom.xml File

```
<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>5.3.9</version>
</dependency>

        <!-- https://mvnrepository.com/artifact/org.springframework/spring-orm -->
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-orm</artifactId>
    <version>5.3.9</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-validator -->
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-validator</artifactId>
    <version>6.0.22.Final</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.hibernate/hibernate-core -->
<dependency>
    <groupId>org.hibernate</groupId>
    <artifactId>hibernate-core</artifactId>
    <version>5.6.3.Final</version>
</dependency>
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.16</version>
</dependency>
```

Spring MVC - Hibernate

Cài MySQL

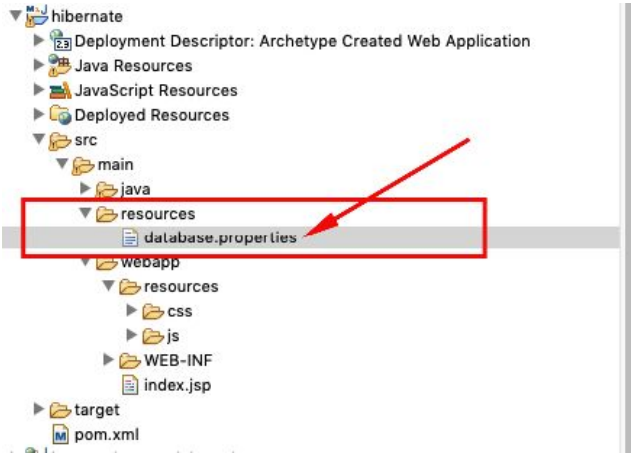
Cài Workbench



Spring MVC - Hibernate

Cấu hình Hibernate

Add file resources



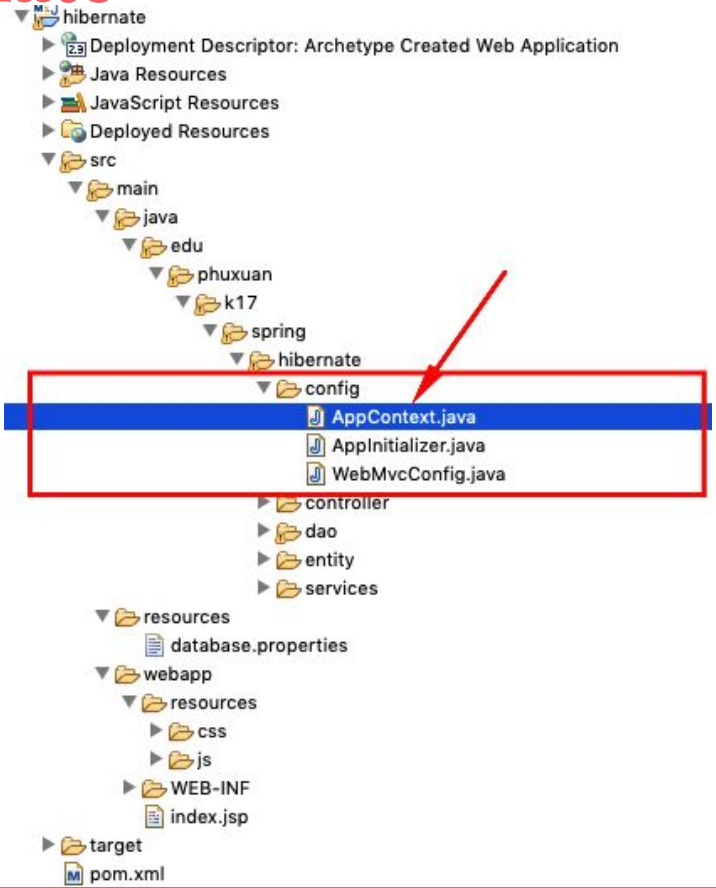
eclipse-workspace - hibernate/src/main/resources/database.properties - Eclipse IDE

```
1 jdbc.driverClassName = com.mysql.jdbc.Driver
2 jdbc.url = jdbc:mysql://localhost:3306/demo
3 jdbc.username = root
4 jdbc.password = 1qaz0plm
5 hibernate.dialect = org.hibernate.dialect.MySQL5InnoDBDialect
6 hibernate.show_sql = true
7 hibernate.format_sql = true
8 hibernate.hbm2ddl.auto = update
```

Spring MVC - Hibernate

Cấu hình Hibernate

Config qua Java - Anotation



Spring MVC – Hibernate

Cấu hình Hibernate

Config qua

Java - Anotation

```java
@Configuration
@PropertySource("classpath:database.properties")
@EnableTransactionManagement
public class AppContext {

    @Autowired
    private Environment environment;

    @Bean
    public LocalSessionFactoryBean sessionFactory() {
        LocalSessionFactoryBean sessionFactory = new LocalSessionFactoryBean();
        sessionFactory.setDataSource(dataSource());
        sessionFactory.setPackagesToScan(new String[] {
            "edu.phuxuan.k17.spring.hibernate.entity"
        });
        sessionFactory.setHibernateProperties(hibernateProperties());
        return sessionFactory;
    }

    @Bean
    public DataSource dataSource() {
        DriverManagerDataSource dataSource = new DriverManagerDataSource();
        dataSource.setDriverClassName(environment.getRequiredProperty("jdbc.driverClassName"));
        dataSource.setUrl(environment.getRequiredProperty("jdbc.url"));
        dataSource.setUsername(environment.getRequiredProperty("jdbc.username"));
        dataSource.setPassword(environment.getRequiredProperty("jdbc.password"));
        return dataSource;
    }

    private Properties hibernateProperties() {
        Properties properties = new Properties();
        properties.put("hibernate.dialect", environment.getRequiredProperty("hibernate.dialect"));
        properties.put("hibernate.show_sql", environment.getRequiredProperty("hibernate.show_sql"));
        properties.put("hibernate.format_sql", environment.getRequiredProperty("hibernate.format_sql"));
        properties.put("hibernate.hbm2ddl.auto", environment.getRequiredProperty("hibernate.hbm2ddl.auto"));
        return properties;
    }

    @Bean
    public HibernateTransactionManager getTransactionManager() {
        HibernateTransactionManager transactionManager = new HibernateTransactionManager();
        transactionManager.setSessionFactory(sessionFactory().getObject());
        return transactionManager;
    }
}
```

Kết nối file resources

Khai báo package entity

Spring MVC - Hibernate

Hibernate sử dụng 2 key player để tương tác dữ liệu

- SessionFactory
  - Đọc hibernate config file
  - Tạo đối tượng session
  - Chỉ tạo 1 lần trong app
- Session
  - Để chuyển đổi các JDBC connection
  - Đối tượng chính để save/retrieve objects
  - Vòng đời ngắn
  - Tạo ra từ SessionFactory

Spring MVC - Hibernate

Hibernate sử dụng 2 key player để tương tác dữ liệu

- SessionFactory

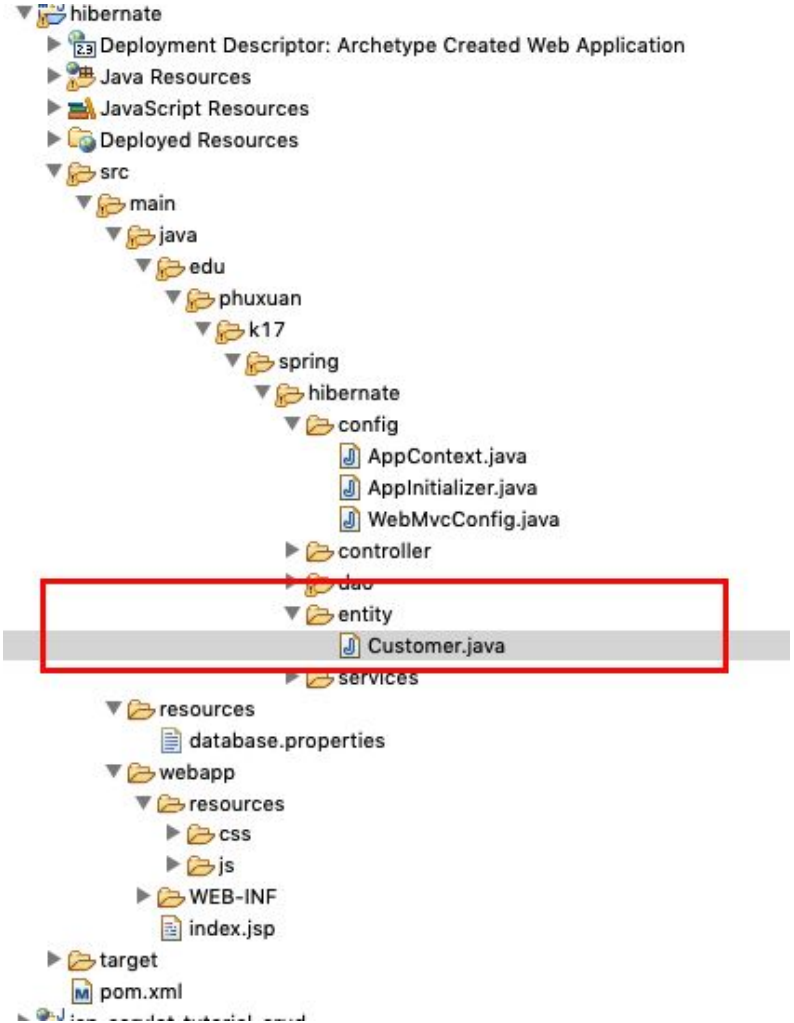
```
@Bean
public LocalSessionFactoryBean sessionFactory() {
    LocalSessionFactoryBean sessionFactory = new LocalSessionFactoryBean();
    sessionFactory.setDataSource(dataSource());
    sessionFactory.setPackagesToScan(new String[] {
        "edu.phuxuan.k17.spring.hibernate.entity"
    });
    sessionFactory.setHibernateProperties(hibernateProperties());
    return sessionFactory;
}
```

- Session

```
Session session = sessionFactory.getCurrentSession();
CriteriaBuilder cb = session.getCriteriaBuilder();
CriteriaQuery < Customer > cq = cb.createQuery(Customer.class);
Root < Customer > root = cq.from(Customer.class);
cq.select(root);
Query query = session.createQuery(cq);
return query.getResultList();
```

Spring MVC - Hibernate

Entity



Spring MVC - Hibernate

Entity Anotation

@Entity: Quy định đó là 1 entity của hibernate

@Table(name="") định nghĩa tên bảng trong Database

@Column(name="") : định nghĩa trường(cột) của bảng

@id: định nghĩa là trường khoá ID của bảng

```java
@Entity
@Table(name = "customer")
public class Customer {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    @Column(name = "id")
    private int id;

    @Column(name = "first_name")
    private String firstName;

    @Column(name = "last_name")
    private String lastName;

    @Column(name = "email")
    private String email;

    public Customer() {

    }
```

Spring MVC -Hibernate

Entity Mapping

```
@Entity
@Table(name="student")
public class Student {

   @Id
   @Column(name="id")
   private int id;

   @Column(name="first_name")
   private String firstName;
  …
}
```