

# Basic Hibernate Framework

<i>Course</i>	Java programming
<i>Trainer</i>	
<i>Designed by</i>	To Chau
<i>Last updated</i>	Jul 25, 2011

# Objectives

- This seminar will provide you a solid understanding of what is Hibernate framework and how we can use it in project.

# Contents

- What is ORM?
- How to save data into Database?
- What is Hibernate framework?
- Why we use Hibernate?
- Hibernate architecture.
- How to use Hibernate in project?
- Example
- Hibernate tools
- References
- Q&A

# What is ORM

- Object-relational mapping (ORM) is a mechanism that makes it possible to address, access and manipulate objects without having to consider how those objects relate to their data sources.

Person
-firstName: String
-lastName: String
-birthday: Calendar



Person	
<u>firstName</u>	<u>varchar</u>
<u>lastName</u>	<u>varchar</u>
birthday	date

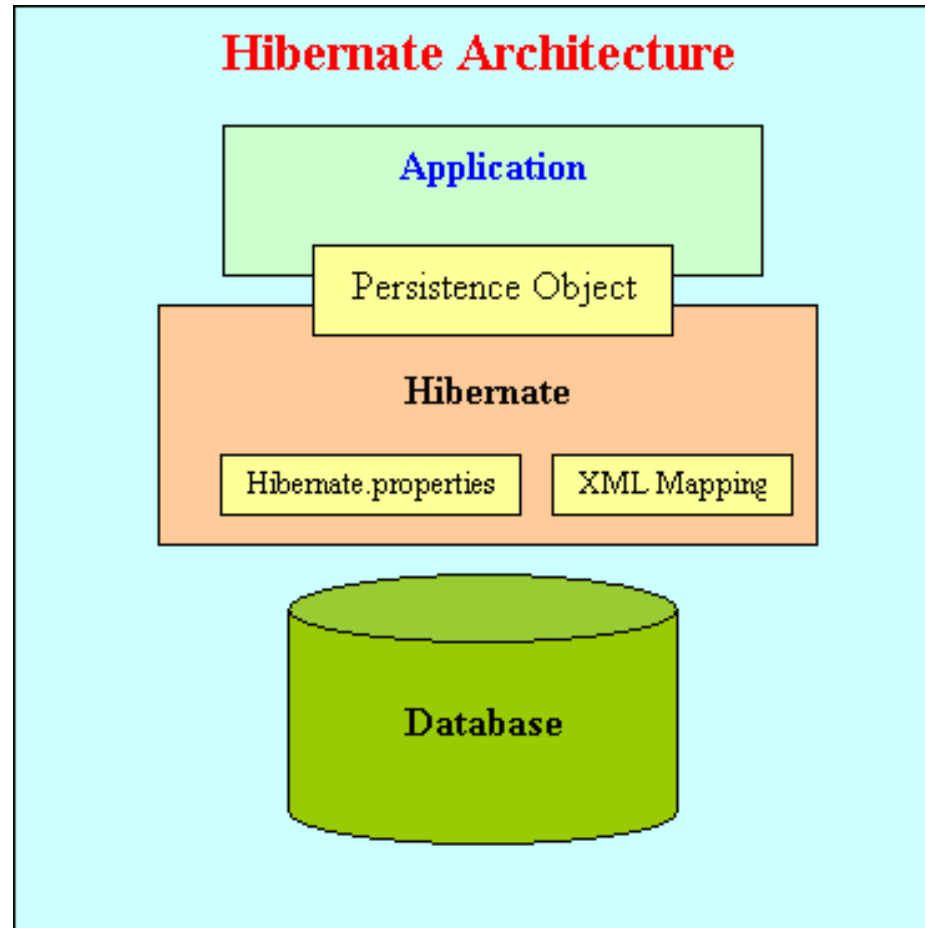
# How to save data into database?

- What you can imagine that you have your application with some functions (business logic) and you want to save data into a database?
- **Without Hibernate:** the application code usually calls the connection pool to obtain JDBC connections and execute SQL statements.
- **Problem:** when you use Java all the business logic normally works with objects of different class types. Your database tables are not at all objects.

# What is Hibernate?

- Hibernate is a powerful, high performance object/relational persistence and query service.
- Hibernate provides a solution to map database tables to classes using XML configuration.
- Act as a client of the JDBC connection pool. The application code uses classes, the Hibernate Session and Query APIs for persistence operations.
- The main feature of Hibernate: simplify the job of accessing a database.

# Hibernate Architecture





# How to use Hibernate?

- Create **Java classes** that represent the tables in the database
- Write **Hibernate Mapping files** to map the instance variables in the class with the columns of table (\*.hbm.xml).
- Create **Hibernate configuration file**:  
hibernate.cfg.xml or hibernate.properties.
- Use Hibernate API in business logic layer for database operations.

# How to use Hibernate: Example

- Data table: CONTACT

Person
<b>ID</b>
FIRSTNAME
LASTNAME
EMAIL

# How to use Hibernate: Example

## ■ Persistence Class : Person

```
package roseindia.tutorial.hibernate;
public class Contact {
    private String firstName;
    private String lastName;
    private String email;
    private long id;

    public String getFirstName() { return firstName; }
    public void setFirstName(String string) { firstName = string; }
    public String getLastName() { return lastName; }
    public void setLastName(String string) { lastName = string; }
    ...
}
```

# How to use Hibernate: Example

- **Hibernate Mapping file:** `person.hbm.xml`

```
<?xml version="1.0"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">
<hibernate-mapping>
  <class name="Person" table="PERSON">
    <id name="id" type="long" column="ID" >
      <generator class="increment"/>
    </id>
    <property name="firstName">
      <column name="FIRSTNAME" />
    </property>
    <property name="lastName">
      <column name="LASTNAME"/>
    </property>
    <property name="email">
      <column name="EMAIL"/>
    </property>
  </class>
</hibernate-mapping>
```

# How to use Hibernate: Example

- **Hibernate configuration file:** `hibernate.cfg.xml`

```
<?xml version='1.0' encoding='utf-8'?>
<!DOCTYPE hibernate-configuration PUBLIC "-//Hibernate/Hibernate Configuration DTD//EN"
"http://hibernate.sourceforge.net/hibernate-configuration-3.0.dtd">
<hibernate-configuration>
  <session-factory>
    <property name="hibernate.connection.driver_class">
      com.mysql.jdbc.Driver</property>
    <property name="hibernate.connection.url">
      jdbc:mysql://localhost/hibernatetutorial</property>
    <property name="hibernate.connection.username">root</property>
    <property name="hibernate.connection.password"></property>
    <property name="hibernate.connection.pool_size">10</property>
    <property name="show_sql">true</property>
    <property name="dialect">org.hibernate.dialect.MySQLDialect</property>
    <!-- Mapping files -->
    <mapping resource="person.hbm.xml"/>
  </session-factory>
</hibernate-configuration>
```

# How to use Hibernate: Example

- Use Hibernate API in business logic layer



```
public class FirstExample {  
    public static void main(String[] args) {  
        Session session = null;  
        try{  
            SessionFactory sessionFactory = new Configuration().configure().  
                buildSessionFactory();  
            session = sessionFactory.openSession();  
            Person person = new Person();  
            person.setId(3);  
            person.setFirstName("Deepak");  
            person.setLastName("Kumar");  
            person.setEmail("deepak_38@yahoo.com");  
            session.save(contact);  
        }catch(Exception e){  
            ex.printStackTrace();  
        }finally{  
            // Actual contact insertion will happen at this step  
            session.flush();  
            session.close();  
        }  
    }  
}
```

# Exercise:

# Why we use Hibernate?

## ■ Productivity:

- Reduce the development time: Hibernate eliminates much of the grunt and lets you concentrate on the business problem.

## ■ Maintainability:

- Disadvantages when including all application functionality and the access to the database within your dialogs: **difficult to reuse code or apply changes.**
- Hibernate provide a solution to separate your dialogs from business logic layer and data access layer.
- Easily apply changes to one part without influencing the other parts.

# Why we use Hibernate?

- **Advanced features difficult to develop yourself:**  
caching solutions, transactions and other features that Hibernate provides are not so easy to implement. Using Hibernate everything is there. You just have to use it.
- **High performance**
- **Support for a wide range of databases**
- **Free/open source**

# Hibernate Tools

- Middlegen for Hibernate
- Hibernate Plugin for Eclipse
- XDoclet for Hibernate

# References

- <http://www.eclipse.org>
- <http://www.hibernate.org>
- <http://www.roseindia.net/hibernate/index.shtml>

# Document Revision History

Date	Version	Description	Revised by
13 Sep 2011	1.0	First version	To Chau