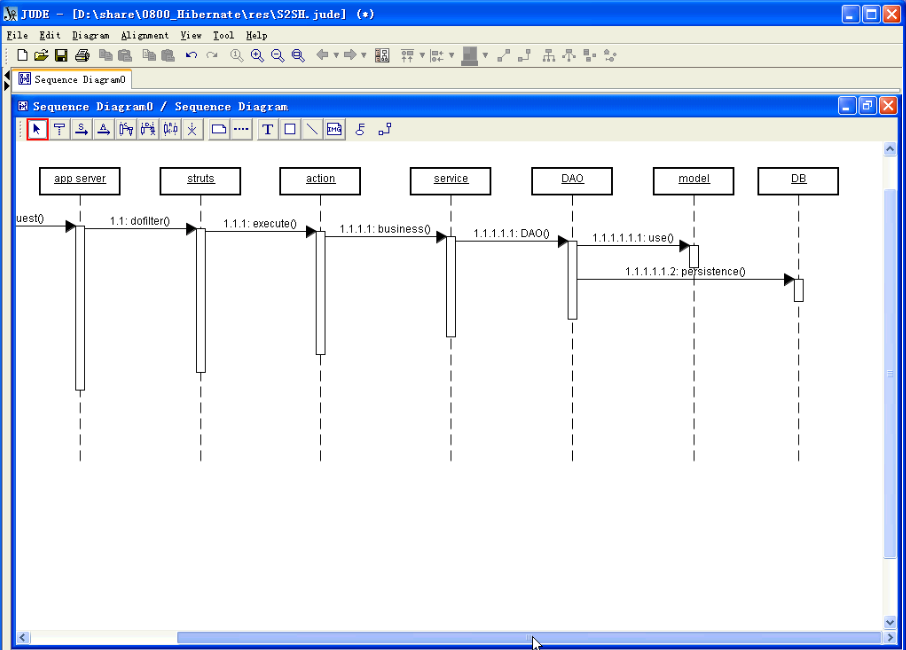
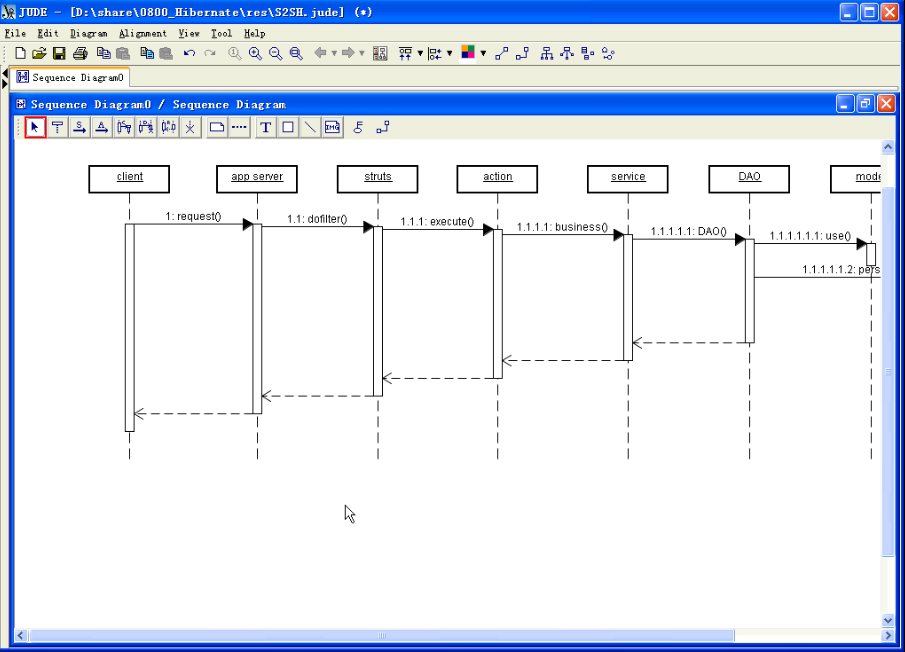
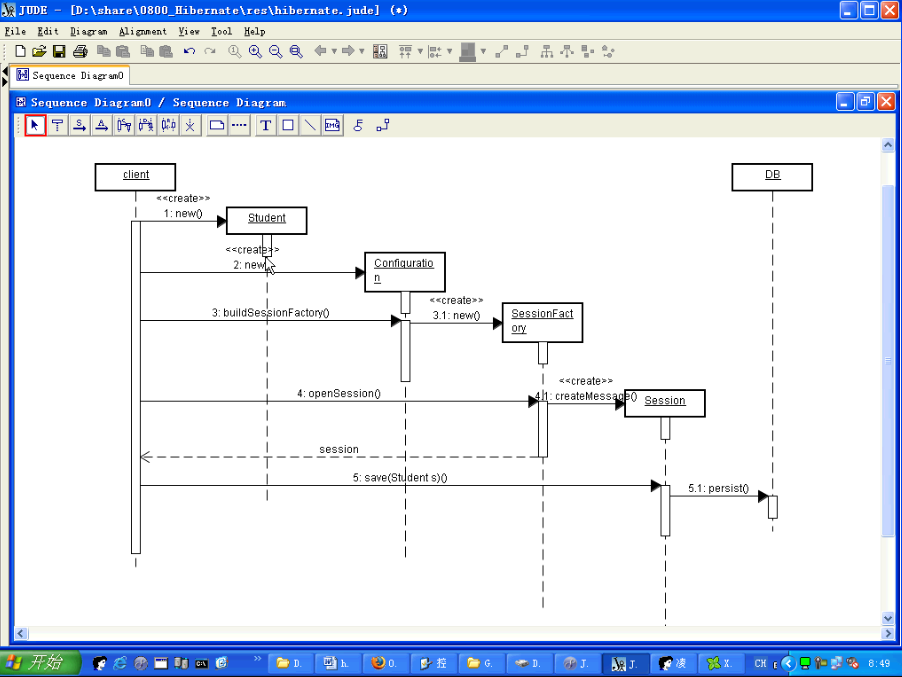
# SSh框架





# Hibernate



# holleword

导入Hibernate Jar包

导入slf4j.jar,mysqy.jar

创建model类

创建数据库表

创建xxx.hbm.xml 映射配置文件

创建hibernate.cfg.xml配置文件

client:

创建model对象

创建Configuration cfg对象

加载hibernate.cfg.xml文件

从Configuration cfg对象中构建SessionFactory sf对象

从SessionFactory 对象打开Session session对象

打开Session对象事务,session.beginTransaction();

执行持久化操作

session.getTransaction().commit();

session.close();

sf.close();

# ID生成策略

## 单一主键

### Xml配置(常用的类型)

#### 使用<generator>标签

* **identity**

对DB2,MySQL, MS SQL Server, Sybase和HypersonicSQL的内置标识字段提供支持。 返回的标识符是long, short或者int类型的。

* **sequence**

在DB2,PostgreSQL, Oracle, SAP DB, McKoi中使用序列（sequence)， 而在Interbase中使用生成器(generator)。返回的标识符是long, short或者 int类型的。

* **guid**

在MS SQL Server 和 MySQL 中使用数据库生成的GUID字符串。

* **native**

selects identity, sequence or hilo depending upon the capabilities of the underlying database.

### annotation配置

#### @Id

#### @GeneratedValue

* AUTO - 可以是identity column类型,或者sequence类型或者table类型,取决于不同的底层数据库.默认使用auto\_increment
* TABLE - 使用表保存id值
* IDENTITY - identity column
* SEQUENCE - sequence ,默认生成hibernate\_sequence

## 联合主键

### Xml配置

#### 使用<composite-id>标签

#### 创建主键类,要实现Serializable接口,重写equals,hashCode方法

### annotation配置

* 将组件类注解为@Embeddable,并将组件的属性注解为@Id
* 将组件的属性注解为@EmbeddedId
* 将类注解为@IdClass,并将该实体中所有属于主键的属性都注解为@Id

# 核心开发接口