

实验二.txt

实验二，Spring+Struts+Hibernate

```
SSHDemo
|
|-----tarena
|
|-----web
|-----struts-config.xml
|-----ApplicationResource.properties
|-----business
|-----spring-config.xml
|-----persistence
|-----hibernate.cfg.xml
|-----db
|-----user.sql

<!-- 加入sshd-lib-->支持DTD
<!-- 采用springDTD头-->
```

步骤一，创建register.jsp

```
<%@ page language="java" pageEncoding="GBK"%>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01
Transitional//EN">
```

```
<html>
```

```
<head>
```

```
<title>用户注册页面</title>
```

实验二.txt

```
</head>

<body>
    <form action="" method="post">
        <table border="5" align="center">
            <tr>
                <td>
                    用户名
                </td>
                <td>
                    <input
type="text" name="user_name" />
                </td>
            </tr>
            <tr>
                <td>
                    性别
                </td>
                <td>
                    <input
type="text" name="user_gender" />
                </td>
            </tr>
            <tr>
                <td>
                    <input
type="submit" value="注册" />
                </td>
            </tr>
        </table>
    </form>
</body>
</html>
```

实验二.txt

步骤二，创建RegisterForm.java

```
package tarena.web;
```

```
import javax.servlet.http.HttpServletRequest;
import org.apache.struts.action.ActionErrors;
import org.apache.struts.action.ActionForm;
import org.apache.struts.action.ActionMapping;
```

```
public class RegisterForm extends ActionForm {

    private String user_name;

    private String user_gender;

    public ActionErrors validate(ActionMapping mapping,
                                HttpServletRequest request) {
        return null;
    }

    public void reset(ActionMapping mapping,
HttpServletRequest request) {
    }

    public String getUser_name() {
        return user_name;
    }

    public void setUser_name(String user_name) {
        this.user_name = user_name;
    }

    public String getUser_gender() {
        return user_gender;
    }

    public void setUser_gender(String user_gender) {
        this.user_gender = user_gender;
    }
}
```


步骤三，创建RegisterAction.java

```
public class RegisterAction extends Action {

    private IUserDao userDao;

    public void setUserDao(IUserDao userDao) {
        this.userDao = userDao;
        System.out.println("已注入userDao");
    }

    public ActionForward execute(ActionMapping mapping,
        ActionForm form,
                                HttpServletRequest request,
        HttpServletResponse response) {

        RegisterForm registerForm = (RegisterForm)
form; // TODO Auto-generated

        User user = new User();
        user.setUserName(registerForm.getUser_name());

        user.setUserGender(registerForm.getUser_gender());

        userDao.insert(user);

        return mapping.findForward("SUCCESS");
    }
}
```


实验二.txt

步骤四，编写struts-config.xml文件

```
<struts-config>
    <data-sources />

    <form-beans>
        <form-bean name="registerForm"
type="tarena.web.RegisterForm" />
    </form-beans>

    <global-exceptions />

    <global-forwards />

    <action-mappings>
        <action attribute="registerForm"
input="/register.jsp"
                                name="registerForm" path="/register"
scope="request"
type="org.springframework.web.struts.DelegatingActionProxy">
            <forward name="SUCCESS"
path="/success.jsp" redirect="true" />
        </action>
    </action-mappings>

    <message-resources
parameter="tarena.web.ApplicationResources" />

    <plug-in
className="org.springframework.web.struts.ContextLoaderPlugIn">
        <set-property property="contextConfigLocation"
value="/WEB-INF/classes/tarena/business/spring-config.xml" />
    </plug-in>
</struts-config>
```

实验二.txt

步骤五，修改web.xml文件

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee"
```

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
version="2.5"
```

```
    xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
```

```
    http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd">
```

```
    <servlet>
```

```
        <servlet-name>action</servlet-name>
```

```
        <servlet-class>
```

```
            org.apache.struts.action.ActionServlet
```

```
        </servlet-class>
```

```
        <init-param>
```

```
            <param-name>config</param-name>
```

```
            <param-value>
```

```
/WEB-INF/classes/tarena/web/struts-config.xml
```

```
                </param-value>
```

```
        </init-param>
```

```
        <init-param>
```

```
            <param-name>debug</param-name>
```

```
            <param-value>3</param-value>
```

```
        </init-param>
```

```
        <init-param>
```

```
            <param-name>detail</param-name>
```

```
            <param-value>3</param-value>
```

```
        </init-param>
```

```
        <load-on-startup>0</load-on-startup>
```

```
    </servlet>
```

```
    <servlet-mapping>
```

```
        <servlet-name>action</servlet-name>
```

```
        <url-pattern>*.do</url-pattern>
```

```
    </servlet-mapping>
```

```
</web-app>
```


实验二.txt

步骤六，编写IUserDao.java

```
public interface IUserDao {  
    public void insert(User user);  
}
```


步骤七，编写User.java

```
public class User implements java.io.Serializable {  
  
    private Integer userId;  
    private String userName;  
    private String userGender;  
  
    public User() {  
    }  
  
    public User(String userName, String userGender) {  
        this.userName = userName;  
        this.userGender = userGender;  
    }  
  
    public Integer getUserId() {  
        return this.userId;  
    }  
  
    public void setUserId(Integer userId) {  
        this.userId = userId;  
    }  
  
    public String getUserName() {  
        return this.userName;  
    }  
  
    public void setUserName(String userName) {
```

实验二.txt

```
        this.userName = userName;
    }

    public String getUserGender() {
        return this.userGender;
    }

    public void setUserGender(String userGender) {
        this.userGender = userGender;
    }
}
```

步骤八，编写User.hbm.xml文件

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE hibernate-mapping PUBLIC "-//Hibernate/Hibernate
Mapping DTD 3.0//EN"
"http://hibernate.sourceforge.net/hibernate-mapping-3.0.dtd">

<hibernate-mapping>
    <class name="tarena.persistence.User" table="user"
catalog="test">
        <id name="userId" type="integer">
            <column name="user_id" />
            <generator class="increment" />
        </id>
        <property name="userName" type="string">
            <column name="user_name" length="20" />
        </property>
        <property name="userGender" type="string">
            <column name="user_gender" length="1" />
        </property>
    </class>
</hibernate-mapping>
```


实验二.txt

步骤九，编写 UserDao 实现类

```
public class UserDao implements IUserDao {  
  
    private HibernateTemplate ht;  
  
    public void setSessionFactory(SessionFactory  
sessionFactory) {  
        if (sessionFactory != null) {  
            ht = new  
HibernateTemplate(sessionFactory);  
        }  
    }  
  
    public void insert(User user) {  
        ht.save(user);  
    }  
  
}
```


步骤十，编写 spring-config.xml 文件

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN//EN"  
"http://www.springframework.org/dtd/spring-beans.dtd">  
<beans>  
    <!-- 数据源 -->  
    <bean id="DriverManagerDataSource"
```

```
class="org.springframework.jdbc.datasource.DriverManagerDataSour
```

实验二.txt

```
ce">
    <property name="driverClassName">
<value>com.mysql.jdbc.Driver</value>
    </property>
    <property name="url">
<value>jdbc:mysql://localhost:3306/test</value>
    </property>
    <property name="username">
        <value>root</value>
    </property>
    <property name="password">
        <value>admin</value>
    </property>
</bean>

<!-- SessionFactory -->
<bean id="LocalSessionFactoryBean"
class="org.springframework.orm.hibernate3.LocalSessionFactoryBean"
n">
    <property name="dataSource">
        <ref bean="DriverManagerDataSource"
/>
    </property>
    <property name="mappingResources">
        <list>
<value>tarena/persistence/User.hbm.xml</value>
        </list>
    </property>
</bean>

<!-- 持久层 -->
<bean id="UserDao" class="tarena.persistence.UserDao">
    <property name="sessionFactory">
        <ref bean="LocalSessionFactoryBean"
/>
    </property>
</bean>
```

实验二.txt

```
<!-- 表现层 -->
<bean name="/register"
class="tarena.web.RegisterAction">
    <property name="userDao">
        <ref bean="UserDao" />
    </property>
</bean>
</beans>
```

步骤十一，部署SSHDemo应用，访问

<http://localhost:8080/SSHDemo/register.jsp>

填入：

用户名：Franking
姓别：M

返回：

注册成功

实验二.txt