【黑马程序员济南】JavaEE就业班同步笔记第三阶段： Struts-...

小鲁哥哥 • 2017-7-13 16:42:09

【黑马程序员济南】JavaEE就业班同步笔记第三阶段：Struts-part02

在action获取表单提交数据

使用ActionContext类获取

```
//获取表单提交数据 ActionContext类
//1 得到ActionContext对象
ActionContext context = ActionContext.getContext();
//2 调用方法得到表单提交数据
Map<String, Object> map = context.getParameters();

//遍历
Set<String> keys = map.keySet();
for (String key : keys) {
    //根据key得到value
    //在map里面value，类型数组形式
    String[] obj = (String[]) map.get(key);
    System.out.println(key+"::"+Arrays.toString(obj));
}
```

使用ServletActionContext类获取

1 直接调用ServletActionContext类里面的静态方法实现操作

```java
//获取表单数据，使用ServletActionContext类
//1 直接使用ServletActionContext类得到request对象
HttpServletRequest request = ServletActionContext.getRequest();
//2 调用request的方法得到表单数据
String username = request.getParameter("username");
String password = request.getParameter("password");
```

2 使用表单post提交中，在struts2的action获取数据不会有乱码问题

（1）struts2有常量设置

```
struts.i18n.encoding=UTF-8
```

3 在action操作域对象

```java
//操作域对象
//1 操作request域
HttpServletRequest request = ServletActionContext.getRequest();
request.setAttribute("request", "requestValue");

//2 操作session域
request.getSession().setAttribute("session", "sessionValue");

//3 servletContext域
ServletContext servletContext = ServletActionContext.getServletContext();
servletContext.setAttribute("servletContext", "servletContextValue");
```

使用实现接口方式获取（了解）

1 在action类实现不同的接口

```
blic class UserFormAction extends ActionSupport implements ServletRequestAware {
    HttpServletRequest request;

    public void setServletRequest(HttpServletRequest request) {
        this.request = request;
    }

    public String execute() {
        //定义成员变量，把setServletRequest里面request对象赋值到成员变量的request里面
        request.getParameter("username");
```

结果页面配置

全局结果页面

1 创建两个action，执行默认的方法 execute方法，默认方法都返回success

```
<action name="person" class="cn.itcast.action.result.PersonAction">
    <result name="success">/form.jsp</result>
</action>
<action name="orders" class="cn.itcast.action.result.OrdersAction">
    <result name="success">/form.jsp</result>
</action>
```

（1）上面配置两个action，两个action执行execute方法，两个action的execute方法都返回success，返回之后都到form.jsp页面中，这样写功能没有问题，造成有很多重复配置，使用全局结果页面操作

2 具体配置

（1）action标签所在的package标签里面

（2）使用global-results进行配置

```
<!-- 配置全局结果页面 -->
<global-results>
    <result name="success">/form.jsp</result>
</global-results>

<action name="person" class="cn.itcast.action.result.PersonAction">
    <!-- <result name="success">/form.jsp</result> -->
</action>
<action name="orders" class="cn.itcast.action.result.OrdersAction">
    <!-- <result name="success">/form.jsp</result> -->
</action>
```

局部结果页面

1 局部结果页面：在action标签里面写result标签进行配置

2 如果配置全局，也配置了局部，最终以局部为准

```
<!-- 配置全局结果页面 -->
<global-results>
    <result name="success">/form.jsp</result>
</global-results>

<action name="person" class="cn.itcast.action.result.PersonAction">
    <result name="success">/a.jsp</result>
</action>
```

result标签的type属性

1 result标签作用：在action标签里面配置action的方法返回结果到路径

2 result标签里面有name属性，和方法的返回值一样，进行配置

3 result标签里面有属性 type

（1）type属性作用：因为result标签配置到路径，type属性表示配置如何到路径中（重定向和转发）

（2）type属性值有很多

- 默认值：不写就是这个值，表示转发操作，dispatcher
- 表示重定向操作，redirect

-- 上面的两个值dispatcher、redirect这两个值表示配置到不同的 页面 中

4 演示type属性两个值

（1）dispatcher： 表示的转发操作，默认就是这个值

（2）redirect，表示的重定向操作

在页面中不能获取到request域里面的值

```
<result name="success" type="redirect">/a.jsp</result>
```

Struts2封装数据操作

使用传统方式封装数据到对象

```
//1 获取表单提交数据
HttpServletRequest request = ServletActionContext.getRequest();
String username = request.getParameter("username");
String password = request.getParameter("password");

//2 封装到实体类对象里面
User user = new User();
user.setUsername(username);
user.setPassword(password);
```

属性封装

1 把表单提交数据，封装到action里面的属性中

2 具体实现

```
username:<input type="text" name="username"/>
password:<input type="password" name="password"/>
<br/>
```

（1）在action里面定义成员变量

- 成员变量名称和表单输入项的name属性值一样

（2）生成定义的变量set和get方法

```java
//2 生成set和get方法
public String getUsername() {
    return username;
}
public void setUsername(String username) {
    this.username = username;
}
public String getPassword() {
    return password;
}
public void setPassword(String password) {
    this.password = password;
}
```

模型驱动封装（重点）

1 使用属性封装，把数据封装到action里面的属性中，不能直接把数据封装到对象里面

2 使用模型驱动封装：把直接把表单提交数据封装到实体类对象里面

3 具体实现

第一步 让action类实现接口 ModelDriven

（1）ModelDriven<封装实体类名称>

- dbutils: new BeanHandler<User>(User.class)

第二步 实现接口里面的方法

（1）getModel方法

第三步 在action的成员变量位置，手动创建实体类对象

注意问题

1 在一个action中获取同一个表单提交数据，

（1）可以使用属性封装，可以使用模型驱动，但是这两种方式不能同时使用，

（2）如果同时使用，只会使用到其中的一种，使用模型驱动。

回复帖子...　回帖

Struts2封装复杂数据（会用）

封装数据到List集合

1 写表单，提交表单之后，把数据封装到action里面list集合中

2 具体实现

第一步 在action里面声明list集合变量，生成get和set方法

第二步 生成list变量的set和get方法

第三步 在页面的表单里面写法

```
<!-- list遍历普通for循环
    list[0] : list集合中第一个元素
    list集合中元素是user对象
    list[0].username: list集合中第一个user对象里面的username属性
-->
username:<input type="text" name="list[0].username"/>
password:<input type="password" name="list[0].password"/>
<br/>
username:<input type="text" name="list[1].username"/>
password:<input type="password" name="list[1].password"/>
```

封装数据到Map集合

1 map集合结构 key-value结构

2 实现步骤

第一步 声明map集合变量

第二步 生成map集合变量的set和get方法

```
//声明map变量
private Map<String,User> map = new HashMap<String,User>();
public Map<String, User> getMap() {
    return map;
}
public void setMap(Map<String, User> map) {
    this.map = map;
}
```

第三步 在表单输入项的name属性值写操作

```
username:<input type="text" name="map['one'].username"/>
password:<input type="password" name="map['one'].password"/>
<br/>
username:<input type="text" name="map['abcd'].username"/>
password:<input type="password" name="map['abcd'].password"/>
```

使用属性封装数据到对象（会用）

第一步 在action声明变量，这个变量是实体类变量

（1）private User user;

第二步 生成变量的set和get方法

```
//1 声明实体类变量
private User user;
//生成user的set和get方法
public User getUser() {
    return user;
}
public void setUser(User user) {
    this.user = user;
}
```

第三步 在表单输入项里面写表达式

案例-添加客户功能

1 点击新增客户，到添加页面中

```
//1 到添加页面
public String toAddPage() {

    return "toAddPage";
}
```

2 在添加页面中，输入不同的值，点击保存，提交表单到action

（1）在action使用模型驱动获取表单数据

- 实体类属性名称 和 表单输入项的name属性值一样

```
public class CustomerAction extends ActionSupport implements ModelDriven<Customer>{

    //创建实体类对象
    private Customer customer = new Customer();
    public Customer getModel() {
        return customer;
    }
```

（2）调用方法把数据添加到数据库

```
//2 添加客户的方法
public String add() {
    //调用方法把数据添加到数据库
    CustomerService service = new CustomerService();
    service.addCustomer(customer);
    return "add";
}
```

总结

1 action获取表单数据

（1）使用ActionContext类

（2）使用ServletActionContext类

（3）使用属性封装

（4）使用模型驱动封装（重点）

- action实现接口，
- 在action创建实体类对象
- 实现接口里面的方法，返回创建对象
- 前提条件：表单输入项的name属性值和实体类属性名称一样

2 结果页面配置

（1）全局和局部结果页面配置

（2）result标签type属性

- 默认值是 dispatcher，转发
- redirect，重定向

3 封装复杂数据（会用）
4 使用属性封装把数据封装到对象里面（会用）

回帖

3条回帖

k1453711238

多谢大哥分享啊,希望持续更新啊

沙发 • 2017-8-2 22:35:33

回帖

rimfwfn

多谢大哥分享啊,跟进中

藤椅 • 2017-8-3 18:57:53

回帖

at123

黑马币来了，大家加油哦。。。

板凳 • 2017-8-7 23:13:12

回帖