

在类中用到的方法：

```
HttpServletRequest request = ServletActionContext.getRequest();
```

```
HttpServletResponse response = ServletActionContext.getResponse();
```

```
ServletContext application = ServletActionContext.getServletContext();
```

```
HttpSession session = request.getSession();
```

封装请求参数

第一种：属性驱动之动作类和模型在一起

要求：form表单元素的name属性取值必须和动作类中的属性get/set方法后面保持一致。

是一个名称叫做params的拦截器完成的封装所有的表单参数封装都有该拦截器参与

1、jsp表单：

```
<form action="${pageContext.request.contextPath}/register1" method="POST">
```

```
    用户名： <input type="text" name="name"/><br/>
```

```
    年龄： <input type="text" name="age"/><br/>
```

```
    <input type="submit" value="提交"/>
```

```
</form>
```

2、自定义类 extends ActionSupport 中定义表单的成员变量并定义get/set方法

之后可以在执行方法中直接获取封装用户输入的数据

```
*/  
public class UserAction extends ActionSupport {
```

```
    private String name;  
    private Integer age;
```

对应表单的成员变量 变量名与
表单保持一致

```
    public String register() {  
        System.out.println(name+", "+age);  
        return SUCCESS;  
    }
```

```
    public String getName() {  
        return name;  
    }  
  
    public void setName(String name) {  
        this.name = name;  
    }
```

生成get/set方法

```
    public Integer getAge() {  
        return age;  
    }
```

第二种：属性驱动之动作类和模型分开

要求：表单元素的name属性取值必须是OGNL表达式。

动作类和模型分开的表单

1、表单元素的name属性取值看上去是一个普通的字符串，而实际上已经不是了。

它是一个OGNL表达式。

OGNL Object Graphic Navigation Lanugage

对象 图 导航 语言

```
<form action="${pageContext.request.contextPath}/register2" method="POST">
```

```
    用户名: <input type="text" name="user.name"/><br/>
```

```
    年龄: <input type="text" name="user.age"/><br/>
```

```
    <input type="submit" value="提交"/>
```

```
</form>
```

2、创建独立的JavaBean表单对象，在执行类中通过对象点方法的方式

```
public class User2Action extends ActionSupport {  
    private User user = new User() ;  
  
    public String register() {  
        System.out.println(user.getName()+" "+user.getAge());  
        return SUCCESS;  
    }  
  
    public User getUser() {  
        return user;  
    }  
  
    public void setUser(User user) {  
        this.user = user;  
    }  
}
```

表单Bean对象

第三种：模型驱动

要求：

- 1、动作类必须实现一个ModelDriven接口
- 2、动作类必须提供一个模型对象，并且自己来实例化
- 3、必须提供接口中抽象方法的实现，返回值必须是模型对象

1、JSP表单：

模型驱动：实际开发中采用的方式

```
<form action="${pageContext.request.contextPath}/register3" method="POST">
```

用户名： <input type="text" name="name"/>

年龄： <input type="text" name="age"/>

<input type="submit" value="提交"/>

```
</form>
```

2、类实现方法：

```
public class User3Action extends ActionSupport implements ModelDriven<User>{  
    private User user = new User();  
    @Override  
    public User getModel() {  
        return user;  
    }  
    public String register(){  
        System.out.println(user);  
        return SUCCESS;  
    }  
}
```

实现的接口

表单的实体Bean对象

实现的方法里返回对象
自动将请求的数据封装到对象

使用对象数据