1. OGNL概述

1 之前web阶段，学习过EL表达式，EL表达式在jsp中获取域对象里面的值

2 OGNL是一种表达式，这个表达式功能更加强大，

（1）在struts2里面操作值栈数据

（2）一般把ognl在struts2操作：和struts2标签一起使用操作值栈

3 OGNL不是struts2的一部分，单独的项目，经常和struts2一起使用

（1）使用ognl时候首先导入jar包，struts2提供jar包

2. OGNL入门案例

1 使用ognl+struts2标签实现计算字符串长度

支持对象方法调用。例如：objName.methodName()。

（1）在java代码中，调用字符串.length();

2 使用struts2标签

（1）使用jstl时候，导入jar包之外，在jsp页面中引入标签库

使用struts2标签时候，在jsp中引入标签库

```
<%@ taglib uri="/struts-tags" prefix="s"%>
```

（2）使用struts2标签实现操作

```
<!-- 使用ognl+struts2标签实现计算字符串长度
    value属性值：ognl表达式
-->
<s:property value="'haha'.length()"/>
```

3. 什么是值栈

1 之前在web阶段，在servlet里面进行操作，把数据放到域对象里面，在页面中使用el表达式获取 到，域对象在一定范围内，存值和取值

2 在struts2里面提供本身一种存储机制，类似于域对象，是值栈，可以存值和取值

（1）在action里面把数据放到值栈里面，在页面中获取到值栈数据

3 servlet和action区别

（1）Servlet：默认在第一次访问时候创建，创建一次，单实例对象

（2）Action：访问时候创建，每次访问action时候，都会创建action对象，创建多次，多实例对象

4 值栈存储位置

（1）每次访问action时候，都会创建action对象

（2）在每个action对象里面都会有一个值栈对象（只有一个）

4. 获取值栈对象

1 获取值栈对象有多种方式

（1）常用方式：使用ActionContext类里面的方法得到值栈对象

```
// 1 获取ActionContext类的对象
ActionContext context = ActionContext.getContext();
//2 调用方法得到值栈对象
ValueStack stack1 = context.getValueStack();
```

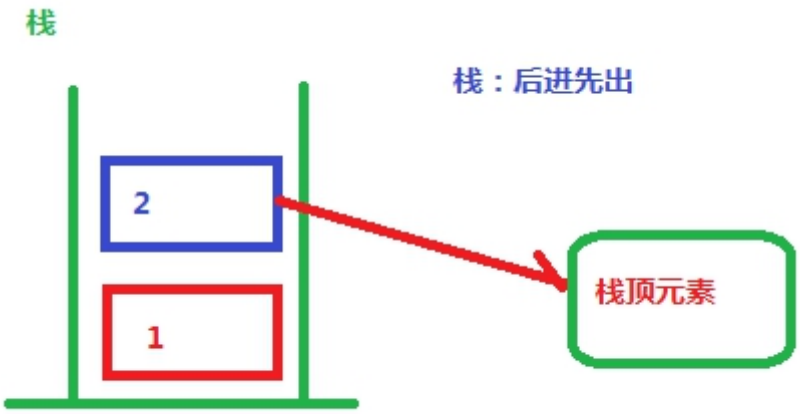
2 每个action对象中只有一个值栈对象

```java
// 1 获取ActionContext类的对象
ActionContext context = ActionContext.getContext();
//2 调用方法得到值栈对象
ValueStack stack1 = context.getValueStack();

ValueStack stack2 = context.getValueStack();

System.out.println(stack1==stack2);
```

```
true
```



5. 值栈内部结构

1 值栈分为两部分：

第一部分 root，结构是list集合

（1）一般操作都是root里面数据


```java
class CompoundRoot extends ArrayList {
```

第二部分 context，结构map集合

2 struts2里面标签 s:debug，使用这个标签可以查看值栈结构和存储值

（1）访问action，执行action的方法有返回值，配置返回值到jsp页面中，在jsp页面中使用这个标签

[Debug]

点击超链接看到结构

Value Stack Contents

Object	Property Name	Property Value
cn.itcast.action.ValueStackDemoAction	texts	null
	actionErrors	[]
	errors	{}
	fieldErrors	{}
	errorMessages	[]
	container	There is no read method for container
	locale	zh_CN
	actionMessages	[]
com.opensymphony.xwork2.DefaultTextProvider	texts	null

（2）在action没有做任何操作，栈顶元素是 action引用

```
cn.itcast.action.ValueStackDemoAction
```

- action对象里面有值栈对象
- 值栈对象里面有action引用

6. 向值栈放数据

1 向值栈放数据多种方式

第一种 获取值栈对象，调用值栈对象里面的 set 方法

```
//第一种方式 使用值栈对象里面的 set方法
//1 获取值栈对象
ActionContext context = ActionContext.getContext();
ValueStack stack = context.getValueStack();
//2 调用方法set方法
stack.set("username", "itcastitheima");
```

```
java.util.HashMap
```

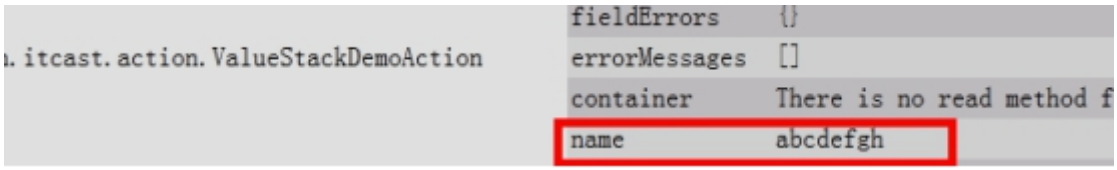
第二种 获取值栈对象，调用值栈对象里面的 push方法

```
//1 获取值栈对象
ActionContext context = ActionContext.getContext
ValueStack stack = context.getValueStack();
//2 调用方法set方法
stack.set("username", "itcastitheima");

//3 调用方法push方法
stack.push("abcd");
```

```
java.lang.String
```

第三种 在action定义变量，生成变量的get方法



7. 向值栈放对象

1 实现步骤

第一步 定义对象变量

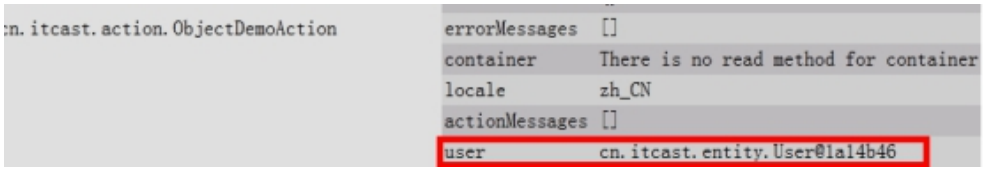
第二步 生成变量的get方法

第三步 在执行的方法里面向对象中设置值

```
//1 定义对象变量
private User user = new User();
//2 生成get方法
public User getUser() {
    return user;
}

public String execute() throws Exception {
    //3 向值栈的user里面放数据
    user.setUsername("lucy");
    user.setPassword("123");
    user.setAddress("美国");

    return "success";
}
```



8. 向值栈放list集合

第一步 定义list集合变量

第二步 生成变量的get方法

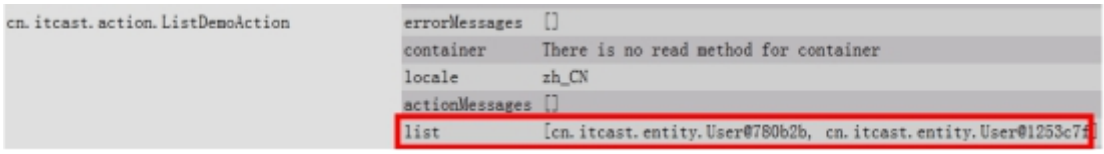
第三步 在执行的方法里面向list集合设置值

```java
// 1 定义list变量
private List<User> list = new ArrayList<User>();
// 2 get方法
public List<User> getList() {
    return list;
}

public String execute() throws Exception {
    //3 向list中设置值
    User user1 = new User();
    user1.setUsername("小奥");
    user1.setPassword("123");
    user1.setAddress("美国");

    User user2 = new User();
    user2.setUsername("小王");
    user2.setPassword("250");
    user2.setAddress("越南");

    list.add(user1);
    list.add(user2);
```



9. 从值栈获取数据

1 使用struts2的标签+ognl表达式获取值栈数据

（1）<s:property value=”ognl表达式”/>

9-1. 获取字符串

1 向值栈放字符串

```java
private String username;
public String getUsername() {
    return username;
}

public String execute() throws Exception {
    username = "itcast";
    return "success";
}
```

2 在jsp使用struts2标签+ognl表达式获取

```jsp
<!-- 1获取字符串值 -->
<s:property value="username"/>
```

9-2. 获取对象

1 向值栈放对象

```java
private User user = new User();
public User getUser() {
    return user;
}

public String execute() throws Exception {
    username = "itcast";

    user.setUsername("mary");
    user.setPassword("250");
    user.setAddress("china");
```

2 在页面中获取值

```jsp
<s:property value="user.username"/>
<s:property value="user.password"/>
<s:property value="user.address"/>
```

9-3. 获取list集合

第一种方式：

```
<!-- 第一种方式实现 -->
<s:property value="list[0].username"/>
<s:property value="list[0].password"/>
<s:property value="list[0].address"/>
<br/>
<s:property value="list[1].username"/>
<s:property value="list[1].password"/>
<s:property value="list[1].address"/>
```

第二种方式：

```
<s:iterator value="list">
    <!-- 遍历list得到list里面每个user对象 -->
    <s:property value="username"/>
    <s:property value="password"/>
    <s:property value="address"/>
    <br/>
</s:iterator>
```

第三种方式：

```
获取list的值第三种方式：
<br/>
<s:iterator value="list" var="user">
    <!--
        遍历值栈list集合，得到每个user对象
        机制：把每次遍历出来的user对象放到 context里面
        获取context里面数据特点：写ognl表达式，
        使用特殊符号 #
     -->
    <s:property value="#user.username"/>
    <s:property value="#user.password"/>
    <s:property value="#user.address"/>
    <br/>
</s:iterator>
```

9-4. 其他操作

1 使用set方法向值栈放数据，获取

```
//2 调用方法set方法
stack.set("itcast", "itcastitheima");
```

```
<!-- 获取set方法设置的值 根据名称获取值-->
<s:property value="itcast"/>
```

2 使用push方法向值栈放数据，获取

```
//3 调用方法push方法
stack.push("abcd");
```

（1）使用push方法设置值，没有名称，只有设置的值

（2）向值栈放数据，把向值栈放数据存到数组里面，数组名称 top，根据数组获取值

```
<s:property value="[0].top"/>
```

10. EL表达式获取值栈数据（为什么）

1 EL表达式获取域对象值

2 向域对象里面放值使用setAttribute方法，获取值使用getAttribute方法

3 底层增强request对象里面的方法getAttribute方法

（1）首先从request域获取值，如果获取到，直接返回

（2）如果从request域获取不到值，到值栈中把值获取出来，把值放到域对象里面

4 查看源代码

```
class StrutsRequestWrapper extends HttpServletRequestWrapper
```

```
public Object getAttribute(String key) {
```

11. OGNL的#、%使用

11-1. #使用

1 使用#获取context里面数据

获取list的值第三种方式：

```
<br/>
<s:iterator value="list" var="user">
    <!--
        遍历值栈list集合，得到每个user对象
        机制：把每次遍历出来的user对象放到 context里面
        获取context里面数据特点：写ognl表达式，
        使用特殊符号 #
     -->
    <s:property value="#user.username"/>
    <s:property value="#user.password"/>
    <s:property value="#user.address"/>
    <br/>
</s:iterator>
```

2 演示# 操作

（1）向request域放值

（2）在页面中使用ognl获取

```
<!-- 获取context里面数据，写ognl时候，首先添加符号
    #context的key名称.域对象名称
-->
<s:property value="#request.req"/>
```

11-2. %使用

1 在struts2标签中表单标签

（1）在struts2标签里面使用ognl表达式，如果直接在struts2表单标签里面使用ognl表达式不识别，只有%之后才会识别。

```
<s:textfield name="username" value="%{#request.req}"><
```