

# 1. Struts2拦截器概述

1 struts2是框架，封装了很多的功能，struts2里面封装的功能都是在拦截器里面

2 struts2里面封装了很多的功能，有很多拦截器，不是每次这些拦截器都执行，每次执行默认的拦截器

3 struts2里面默认拦截器位置

struts2-core-2.3.24.jar - C:\l

struts-default.xml

```
<interceptor-stack name="defaultStack">
  <interceptor-ref name="exception"/>
  <interceptor-ref name="alias"/>
  <interceptor-ref name="servletConfig"/>
  <interceptor-ref name="i18n"/>
  <interceptor-ref name="prepare"/>
  <interceptor-ref name="chain"/>
  <interceptor-ref name="scopedModelDriven"/>
  <interceptor-ref name="modelDriven"/>
  <interceptor-ref name="fileUpload"/>
  <interceptor-ref name="checkbox"/>
  <interceptor-ref name="datetime"/>
  <interceptor-ref name="multiselect"/>
  <interceptor-ref name="staticParams"/>
```

4 拦截器在什么时候执行？

(1) 在action对象创建之后，action的方法执行之前

## 2. 拦截器底层原理

1 拦截器底层使用两个原理

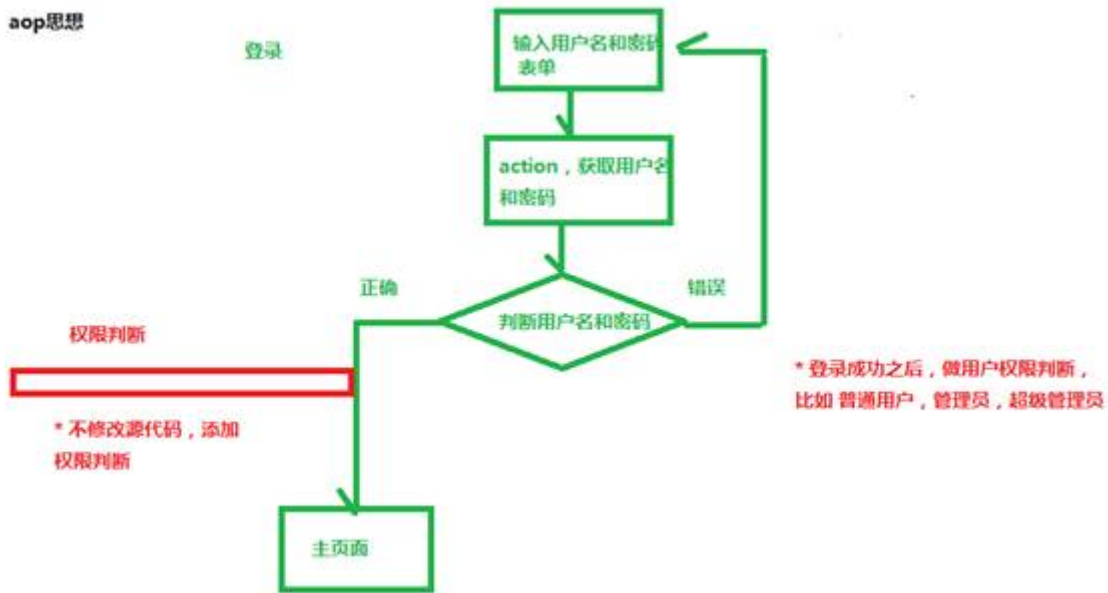
第一个

(0) 后面在spring里面把aop做更深层次分析

(1) 文字描述：

Aop是面向切面（方面）编程，有基本功能，扩展功能，不通过修改源代码方式扩展功能

(2) 画图分析：



## 第二个 责任链模式

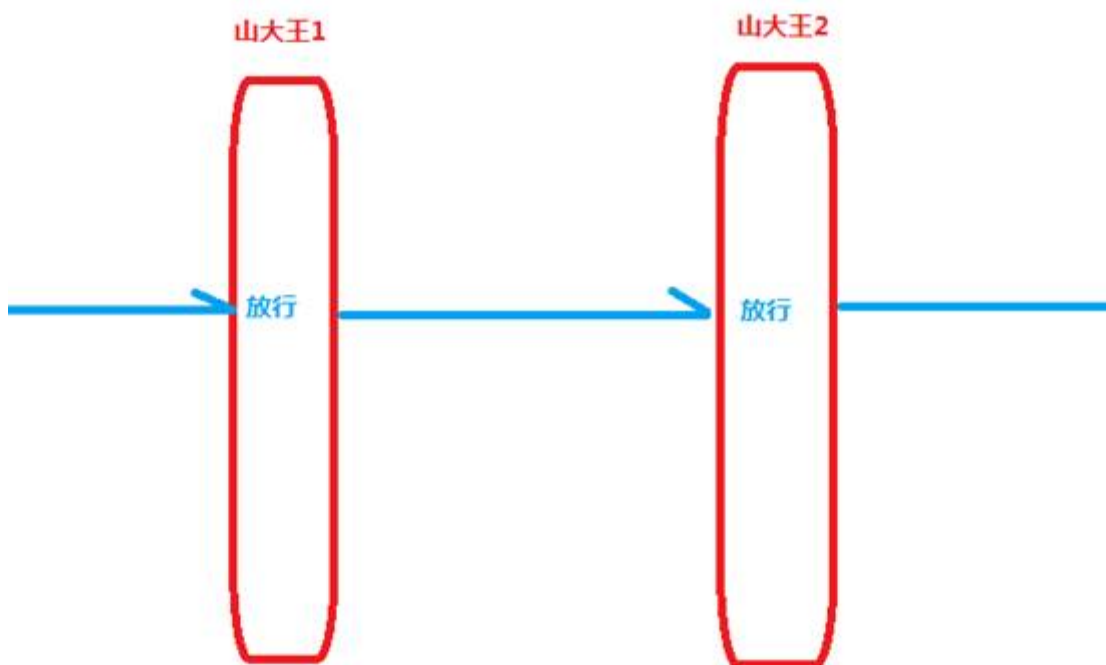
- (1) 在java中有很多的设计模式, 责任链模式是其中的一种
- (2) 责任链模式和过滤链很相似的

责任链模式:

要执行多个操作, 有添加、修改、删除三个操作。

首先执行添加操作, 添加操作执行之后 做类似于放行操作, 执行修改操作, 修改操作执行之后做类似于放行操作, 执行删除操作

过滤链: 一个请求可有多个过滤器进行过滤, 每个过滤器只有做放行才能到下一个过滤器

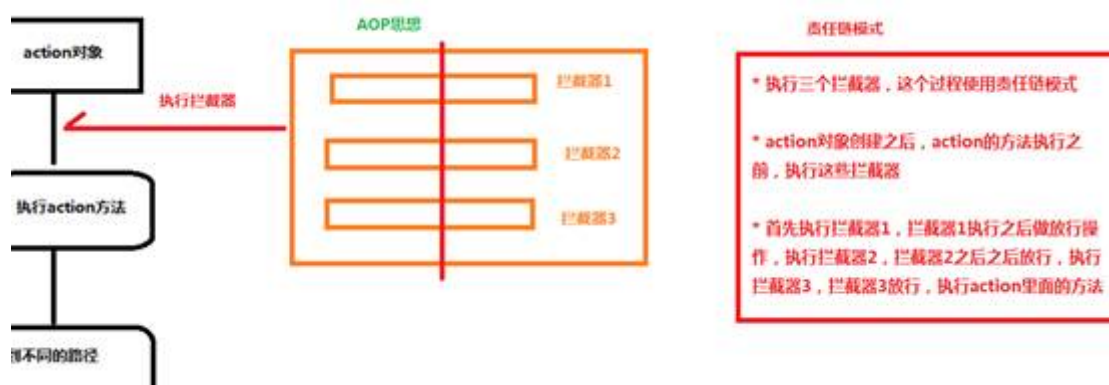


## 2 aop思想和责任链模式如何应用到拦截器里面？

### (1) 文字描述：

- 拦截器在action对象创建之后，action的方法执行之前执行
- 在action方法执行之前执行默认拦截器，执行过程使用aop思想，在action没有直接调用拦截器的方法，使用配置文件方式进行操作
- 在执行拦截器时候，执行很多的拦截器，这个过程使用责任链模式
- 假如执行三个拦截器，执行拦截器1，执行拦截器1之后做放行操作，执行拦截器2，执行拦截器2之后做放行，执行拦截器3，执行拦截器3之后放行，执行action的方法

### (2) 画图分析



## 3 查看源代码

### (1) 执行action

```
execute.executeAction(request, response, mapping);
```

### (2) 创建action对象，使用动态代理方式

```
ActionProxy proxy = getContainer().getInstance(ActionProxyFactory.class).createActionProxy(namespace, name, method, extraContext, true, false);
```

### (3) 执行action的方法

```
proxy.execute();
```

### (4) 执行很多的拦截器，遍历执行

```
if (interceptors.hasNext()) {
```

类似于放行的操作的方法

```
return invocation.invoke();
```

## 3. 重要的概念

### 1 过滤器和拦截器区别

- (1) 过滤器：过滤器理论上可以任意内容，比如html、jsp、servlet、图片路径
- (2) 拦截器：拦截器只可以拦截action

### 2 Servlet和action区别

- (1) servlet默认第一次访问时候创建，创建一次，单实例对象
- (2) action每次访问时候创建，创建多次，多实例对象

## 4. 自定义拦截器

1 在struts2里面有很多的拦截器，这些拦截器是struts2封装的功能，但是在实际开发中，struts2里面的拦截器中可以没有要使用的功能，这个时候需要自己写拦截器实现功能

### 2 拦截器结构

- (1) 查看源代码看拦截器结构
  - 继承类

```
class ModelDrivenInterceptor extends AbstractInterceptor
```

```
class AbstractInterceptor implements Interceptor
```

在接口里面有三个方法：

void init(): 初始化操作

void destroy(): 销毁

String intercept(ActionInvocation invocation): 拦截逻辑的操作

- (2) 开发中，建议使用另外一种方式
  - 写类，继承 `MethodFilterInterceptor` 类实现
  - 让action里面某个的方法不进行拦截
- (3) 让拦截器和action有关系
  - 不是在action调用拦截器的方法，而是通过配置文件方式让建立关系

## 5. 自定义登录拦截器

1 需求：在项目中，有很多的action的超链接，实现只有是登录的状态，才可以点击action的超链接实现功能，如果不是登录状态，点击action超链接返回到登录页面

## 2 登录的状态：使用session域对象实现

- (1) 登录成功之后，把数据放到session里面
- (2) 判断session是否有值，可以知道是否是登录状态

## 3 实现登录的基本功能

- (1) 查询数据库判断用户名和密码（作业）

```
public String login() {  
    //1 得到request对象  
    HttpServletRequest request = ServletActionContext.getRequest();  
    String username = request.getParameter("username");  
    String password = request.getParameter("password");  
    //2 作业：查询数据库判断用户名和密码是否正确  
    // 用户名 admin 密码 250  
    if("admin".equals(username) && "250".equals(password)) {  
        //成功  
        //向session里面放值  
        request.getSession().setAttribute("username", username);  
        return "loginsuccess";  
    } else {  
        //失败  
        return "login";  
    }  
}
```

当前用户: admin

## 4 添加登录拦截器功能

- (1) 判断是否登录：判断session里面是否有名称是username的值
- (2) 拦截器实现过程

第一步 创建类，继承MethodFilterInterceptor类

第二步 重写MethodFilterInterceptor类里面的方法写拦截器逻辑

```
/**  
 * 拦截器类  
 * @author asus  
 */  
public class LoginInterceptor extends MethodFilterInterceptor {  
    //这个方法里面写拦截器逻辑  
    protected String doIntercept(ActionInvocation invocation) throws  
        return null;  
}
```



```

// 判断session里面是否有名称是username的值
// 得到session
HttpServletRequest request = ServletActionContext.getRequest();
Object obj = request.getSession().getAttribute("username");
//判断
if(obj != null) {
    //登录状态
    //做类似于放行操作，执行action的方法
    return invocation.invoke();
} else {
    //不是登录状态
    //不到登录，不执行action的方法，返回登录页面
    //到result标签里面找到名称是login的值，到配置路径里面
    return "login";
}

```

### 第三步 配置action和拦截器关系（注册拦截器）

(1) 在要拦截的action标签所在的package标签里面声明拦截器

```

<!-- 声明拦截器 -->
<interceptors>
    <interceptor name="loginintercept" class="cn.itcast.interceptor.LoginInterceptor">
</interceptors>

```

(2) 在具体的action标签里面使用声明的拦截器

```

<!-- 2 使用自定义拦截器 -->
<interceptor-ref name="Loginintercept"></interceptor-ref>

```

(3) struts2里面执行很多的默认拦截器，但是如果在action里面配置自定义拦截器，

问题：默认的拦截器不会执行了

解决：把默认拦截器手动使用一次

```

<!-- 3 把默认拦截器手动使用一次 -->
<interceptor-ref name="defaultStack"></interceptor-ref>

```

### 5 配置拦截器，对action里面所有的方法都进行拦截

(1) 在action里面有login的登录的方法，这个方法不需要拦截，如果这个方法都拦截，问题是，永远登录不进去了

(2) 解决：让login方法不进行拦截

- 直接通过配置方式让action里面某些方法不进行拦截

```

<!-- 2 使用自定义拦截器 -->
<interceptor-ref name="LoginIntercept">
    <!-- 配置action里面某些方法不进行拦截
        name属性值: excludeMethods
        值: action不拦截的方法名称
    -->
    <param name="excludeMethods">login</param>
</interceptor-ref>

```

6 如果登录状态，直接到功能页面，如果不是登录显示登陆页面  
登录之后出现小问题：



(1) 设置打开位置，在form标签里面

```
target="_parent">
```

## 6. Struts2的标签库

0 struts2标签使用jsp页面中, 不能用在html页面中

1 s:property: 和ognl表达式在jsp页面中获取值栈数据

2 s:iterator: 获取值栈list集合数据, 表示list集合

3 s:debug: 查看值栈结构和数据

## 7. Struts2表单标签 (会用)

1 html表单标签

(1) form: action、method、enctype

(2) 输入项

- 大部分在input里面封装 type=" 值"

- text: 普通输入项
  - password: 密码输入项
  - radio: 单选输入项
  - checkbox: 复选输入项
  - file: 文件上传项
  - hidden: 隐藏项
  - button: 普通按钮
  - submit: 提交按钮
  - image: 图片提交
  - reset: 重置
- 
- select: 下拉输入项
  - textarea: 文本域

2 在struts2里面对应html表单标签大部分都有

```
<s:form>
  <!-- 1 普通输入项 -->
  <s:textfield name="username" label="username"></s:textfield>

  <!-- 2 密码输入项 -->
  <s:password name="password" label="password"></s:password>

  <!-- 3 单选输入项 -->
  <!-- value属性值和显示值一样的 -->
  <s:radio list="{ '女', '男' }" name="sex" label="性别"></s:radio>

  <!-- value属性值和显示值不一样的 -->
  <s:radio list="#{'nv': '女', 'nan': '男'}" name="sex1" label="性别"></s:radio>

  <!-- 4 复选输入项 -->
  <s:checkboxlist list="{ '吃饭', '睡觉', '敲代码' }" name="love" label="爱好"></s:checkboxlist>

  <!-- 5 下拉输入框 -->
  <s:select list="{ '幼儿园', '博士后', '教授' }" name="college" label="学历"></s:select>

  <!-- 6 文件上传项 -->
  <s:file name="file" label="上传文件"></s:file>

  <!-- 7 隐藏项 -->
  <s:hidden name="hid" value="abcd"></s:hidden>

  <!-- 文本域 -->
  <s:textarea rows="10" cols="3" name="resume" label="简历"></s:textarea>

  <!-- 8 提交按钮 -->
  <s:submit value="提交"></s:submit>

  <!-- 9 重置 -->
  <s:reset value="重置"></s:reset>
```



## 8. 缺少两个知识点

放到ssh练习中讲到

1 struts2文件上传

2 错误处理机制 input