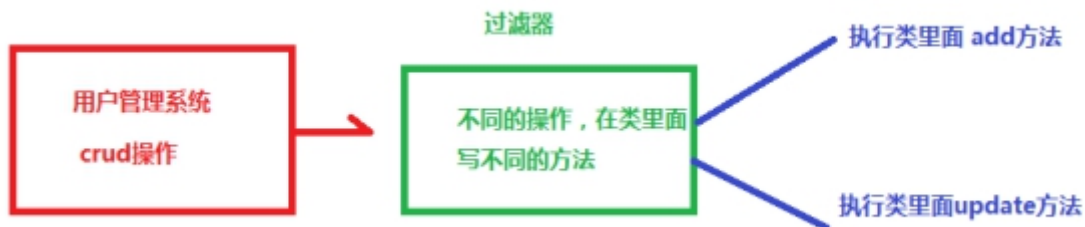
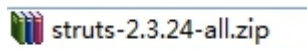


1. Struts2概述

- 1 struts2框架应用javaee三层结构中web层框架
- 2 struts2框架在struts1和webwork基础之上发展全新的框架
- 3 struts2解决问题：



4 struts2版本



5 web层常见框架

- (1) struts2
- (2) springMVC

2. Struts2框架入门

第一步 导入jar包



- (1) 在lib中有jar包，不能把这些jar都导入到项目中
- (2) 到apps目录里面，找到示例程序，从示例程序复制jar包



struts2-blank.war\WEB-INF\lib - 解包大小为 5.2 MB		
名称	压缩前	压缩后
.. (上级目录)		
asm-3.3.jar	42.6 KB	40.4 KB
asm-commons-3.3.jar	37.4 KB	34.4 KB
asm-tree-3.3.jar	21.0 KB	17.8 KB
commons-fileupload-1.3.1.jar	67.4 KB	58.6 KB
commons-io-2.2.jar	169.5 KB	154.8 KB
commons-lang3-3.2.jar	375.7 KB	348.8 KB
freemarker-2.3.22.jar	1.2 MB	1.1 MB
javassist-3.11.0.GA.jar	599.8 KB	563.1 KB
log4j-api-2.2.jar	130.4 KB	117.9 KB
log4j-core-2.2.jar	807.4 KB	713.9 KB
ognl-3.0.6.jar	222.7 KB	212.3 KB
struts2-core-2.3.24.jar	812.5 KB	718.9 KB
xwork-core-2.3.24.jar	660.9 KB	585.1 KB

第二步 创建action

```
public class HelloAction {
```

```
    /*
     * (1) 每次访问servlet时候，都会执行service方法
     *
     * (2) 访问action，每次访问action时候，默认执行名称execute方法
     * */
    public String execute() {
        return "ok";
    }
}
```

第三步 配置action类访问路径

(1) 创建struts2核心配置文件

- 核心配置文件名称和位置是固定的
- 位置必须在src下面，名称 struts.xml

(2) 引入dtd约束

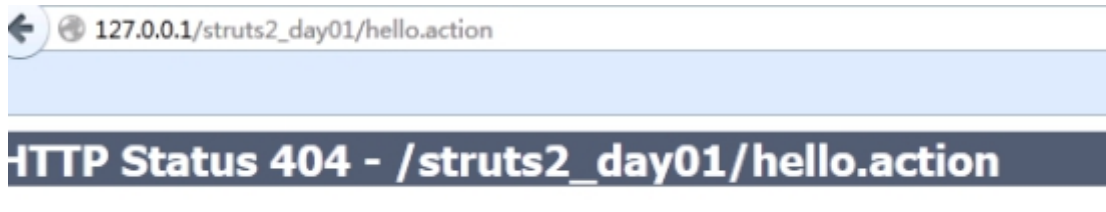
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
    "http://struts.apache.org/dtds/struts-2.3.dtd">
```

(3) action配置

```
<package name="helldemo" extends="struts-default" namespace="/">
    <!-- name: 访问名称 -->
    <action name="hello" class="cn.itcast.action.HelloAction">
        <!-- 配置方法的返回值到页面 -->
        <result name="ok"/>hello.jsp</result>
    </action>
</package>
```

访问路径:

http://127.0.0.1:8080/struts2_day01/hello.action

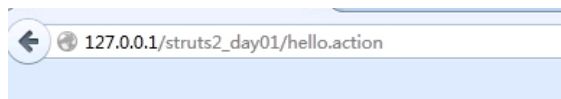


第四步 在web.xml中配置struts2过滤器

```
<filter>
    <filter-name>struts2</filter-name>
    <filter-class>org.apache.struts2.dispatcher.ng.filter.StrutsPrepareAndExecuteFilter</filter-class>
</filter>

<filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

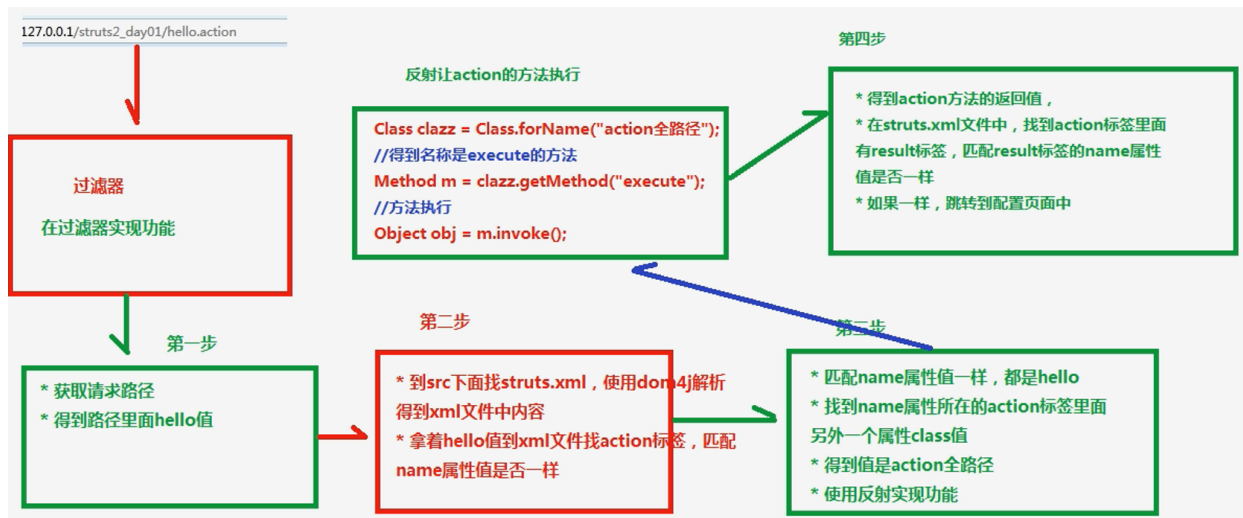
```
class StrutsPrepareAndExecuteFilter implements StrutsStatics, Filter {
    ...
}
```



hello struts2...

3. Struts2执行过程

画图分析过程



4. 查看源代码

```
class StrutsPrepareAndExecuteFilter implements StrutsStatics, Filter
```

1 过滤器在服务器启动时候创建, 创建过滤器时候执行init方法

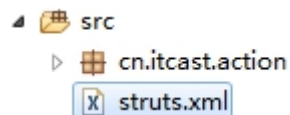
(1) 在init方法中主要加载配置文件

- 包含自己创建的配置文件和struts2自带配置文件

** struts.xml

** web.xml

5. Struts2的核心配置文件



1 名称和位置固定的

2 在配置文件中主要三个标签 package、action、result, 标签里面的属性

6. package标签

1 类似于代码包, 区别不同的action, 要配置action, 必须首先写package标签, 在package里面 才能 配置action

2 package标签属性

(1) name属性

- name属性值跟功能本身没有关系的
- 在一个配置文件中可以写多个package标签，name属性值不能相同的

(2) extends属性

`extends="struts-default"`

- 属性值固定的，struts-default
- 写了这个属性之后，在package里面配置的类具有action功能

(3) namespace属性

- namespace属性值和action标签里面的name属性值构成访问路径

```
<package name="helldemo" extends="struts-default" namespace="/">
  <!-- name: 访问名称 -->
  <action name="hello" class="cn.itcast.action.HelloAction">
    <!-- 配置方法的返回值到页面 -->
```

7. action标签

1 action标签配置action访问路径

2 action标签属性

(1) name属性

- namespace属性值和action标签里面的name属性值构成访问路径
- 在package标签里面写多个action标签，但是action的name属性值不能相同的

(2) class属性

- action全路径

(3) method属性

- 比如在action里面默认执行的方法execute方法，但是在action里面写其他的方法
- 让action里面多个方法执行，使用method进行配置

8. result标签

1 根据action的方法返回值，配置到不同的路径里面

2 result标签属性

(1) name属性

- 和方法返回值一样

```
<!-- 配置方法的返回值到页面 -->
<result name="ok"/>hello.jsp</result>
...
```

(2) type属性

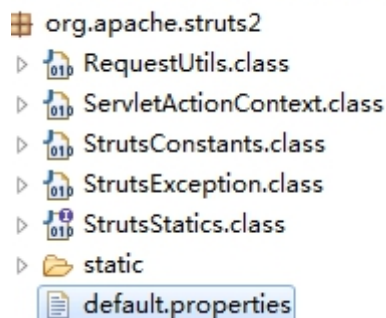
- 配置如何到路径中（转发或者重定向）
- type属性默认值 做转发操作

9. Struts2常量配置

1 struts2框架，帮我们实现一部分功能，struts2里面有常量，在常量里面封装一部分功能

2 struts2默认的常量位置（记住）

struts2-core-2.3.24.jar - C:\Users\asus\De:



3 修改struts2默认常量值

(1) 常用的方式

- 在struts.xml中进行配置

```
<constant name="struts.i18n.encoding" value="UTF-8"></constant>
```

(2) 还有两种方式（了解）

- 在src下面创建 struts.properties，进行修改
- 在web.xml进行配置

4 介绍最常用常量（明天演示）

struts.i18n.encoding=UTF-8

(1) 表单提交数据到action里面，在action可以获取表单提交数据

(2) 表单提交数据有中文，有乱码问题，解决：

- post提交直接设置编码
- get提交做编码转换

(3) 如果在action获取表单通过post方式提交中文，中文乱码问题帮解决了，不需要自己处理问题

10. 分模块开发

1 队员分别单独写配置文件，把配置文件引入到核心配置文件中

```
struts.xml  HelloAction.java  hello.xml
2 <!DOCTYPE struts PUBLIC
3     "-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
4     "http://struts.apache.org/dtds/struts-2.3.dtd">
5 <struts>
6 |
7 <package name="helldemo" extends="struts-default" namespace="/">
8     <!-- name: 访问名称 -->
9     <action name="hello" class="cn.itcast.action.HelloAction">
10         <!-- 配置方法的返回值到页面 -->
11         <result name="ok"/>/hello.jsp</result>
12     </action>
13 </package>
14
15 </struts>

<!-- 引入hello.xml文件 -->
<include file="cn/itcast/action/hello.xml"></include>
```

11. Action编写方式

1 action编写有三种方式

第一种 创建普通类，这个不继承任何类，不实现任何接口

```
public class HelloAction {
```

第二种 创建类，实现接口 Action


```

/**
 * 实现接口
 * @author asus
 *
 */
public class UserAction implements Action{

    @Override
    public String execute() throws Exception {
        return null;
    }

}

@Override
public String execute() throws Exception {
    return "success";
    return SUCCESS;
}

```

第三种 创建类，继承类 ActionSupport (一般使用)

```

/**
 * 继承类
 * @author asus
 *
 */
public class PersonAction extends ActionSupport {

}

```

```

class ActionSupport implements Action,

```

```

@Override
public String execute() throws Exception {
    // TODO Auto-generated method stub
    return SUCCESS;
}

```


12. 访问action的方法（重点）

1 有三种方式实现

第一种 使用action标签的method属性，在这个属性里面写执行的action的方法

第二种 使用通配符方式实现

第三种 动态访问实现（不用）

2 演示错误

(1) 如果action方法有返回值，在配置文件中没有配置，出现错误

s 404 - No result defined for action cn.itcast.action.HelloAction and result ok

(2) 在action里面的方法有返回值，如果有返回值时候类型必须是String

(3) action里面的方法可以没有返回值，没有返回值时候，在result标签不需要配置

- 把方法写成void

- 让返回值，返回 " none"

```
public String execute() {  
    return "none";  
    return NONE;  
}
```

12-1. 使用action标签method属性

1 创建action，创建多个方法

```
public class BookAction extends ActionSupport {  
  
    //添加  
    public String add() {  
        System.out.println("add.....");  
        return NONE;  
    }  
  
    //修改  
    public String update() {  
        System.out.println("update.....");  
        return NONE;  
    }  
}
```

2 使用method配置

```

<!-- 配置action的方法调用 -->
<package name="methoddemo" extends="struts-default" namespace="/">
  <!-- 有method属性: 写要执行的action里面的方法的名称 -->
  <action name="addAction" class="cn.itcast.method.BookAction" method="add"></action>
  <!-- 执行update方法 -->
  <action name="updateAction" class="cn.itcast.method.BookAction" method="update">
</package>
</struts>

```

缺陷: action每个方法都需要配置, 如果action里面有多方法, 配置很多的action

12-2. 使用通配符实现 (重点)

1 在action标签里面name属性, name属性值里面写 符号 * 星号

(1) * 理解: 表示匹配任意内容

- 比如访问hello, * 可以匹配到

- 比如访问add, * 可以匹配到

```

<!-- 通配符方式实现 -->
<package name="methoddemo" extends="struts-default" namespace="/">
  <!--
    name属性值里面写符号 *
    (1)执行action里面的add方法, 访问 book_add, 使用book_*可以匹配到, *相当于变成 add
    (2)执行action里面的update方法, 访问book_update, 使用book_*可以匹配到, *相当于变成 update
    上面两个路径使用 book_* 可以匹配到
    method属性写*值
  -->
  <action name="book_*" class="cn.itcast.method.BookAction" method="{1}"></action>
</package>

```



