

Struts2框架

明确:

它是一个apache组织推出的一套开源的表现层框架。

它只能干表现层该干的事。

它也是基于MVC模型的。

历史由来:

它是2005年之后的产物

搭建开发环境

前期准备:

前往官网下载开发包。

我们用的是struts2-2.3.24版本。

步骤:

1、拷贝必备jar包到web工程的lib目录, 基础JAR包13个

在apps目录中有一个struts-blank的例子, 其中的lib目录下的jar包全需要拷贝

2、在类的根路径下创建一个名称为struts.xml的配置文件

在配置文件中导入dtd约束。

The screenshot shows an IDE with two windows. The top window displays the 'lib' directory of a web project, listing various JAR files. A red box highlights 'struts2-core-2.3.24.jar', with a red arrow pointing to it and the label '核心包' (Core Package). Another red box highlights 'struts-2.3.dtd', with a red arrow pointing to it and the label 'DTD约束文件' (DTD Constraint File). The bottom window shows the 'struts.xml' configuration file. A red box highlights the DTD declaration section, with a red arrow pointing to it and the label '导入约束' (Import Constraint).

```
23 -->
24 <!-- START SNIPPET: strutsDtd -->
25
26 <!-- 核心包
27 Struts configuration DTD.
28 Use the following DOCTYPE
29
30 <!DOCTYPE struts PUBLIC
31     "-//Apache Software Foundation//DTD Struts Configuration 2.3//EN"
32     "http://struts.apache.org/dtds/struts-2.3.dtd">
33 -->
34
35 <!ELEMENT struts ((package|include|bean|constant)*, unknown-handler-stack?)>
36 <!-- ATTLIST struts
-->
```

```
1 <?xml version="1.0" encoding="UTF-8"?>
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></struts>
```

3、在web.xml中配置核心控制器 (Filter)

```
<!-- 配置struts2的核心控制器 -->
<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>
```

```
<!-- 配置struts2的核心控制器 -->
```

```
<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>
```

```
<!-- 配置struts2的核心控制器：过滤器
```

它默认会去类的根路径下加载一个名称为struts.xml的配置文件。我们可以通过手动设置，来指定配置文件的位置
指定的方式是：

配置过滤器的初始化参数

参数的key: config

参数的value:

config/struts/struts.xml (这个是在SRC下创建的目录)

当我们手动指定了struts2的配置文件，必须在写完自己的配置文件位置之后

还得写上: struts-default.xml和struts-plugin.xml

不写会报错

```
-->
```

```
<filter>
```

```
    <filter-name>struts2</filter-name>
```

手动指定配置文件的位置

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

```
    <init-param>
```

```
        <param-name>config</param-name>
```

```
        <param-value>struts-default.xml, struts-plugin.xml, config/struts/struts.xml</param-value>
```

```
    </init-param>
```

```
</filter>
```

```
<filter-mapping>
```

```
    <filter-name>struts2</filter-name>
```

```
    <url-pattern>/*</url-pattern>
```

```
</filter-mapping>
```

```
<filter>
```

```
    <filter-name>struts2</filter-name>
```

```
    <filter-
```

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

```
    <init-param>
```

```
        <param-name>config</param-name>
```

```
        <param-value>struts-default.xml, struts-  
plugin.xml, config/struts/struts.xml</param-value>
```

```
    </init-param>
```

```
</filter>
```

```
<filter-mapping>
```

```
<filter-name>struts2</filter-name>

<url-pattern>/*</url-pattern>

</filter-mapping>
```

验证环境是否搭建成功

部署应用到tomcat,启动tomcat没有报异常，就算搭建成功。

Servlet的执行过程：

第一部分：

应用加载——读取web.xml配置文件

第二部分：

浏览器——tomcat服务器——我们的应用——web.xml——Servlet——执行service——找到对应jsp——响应浏览器

Struts2的执行过程：

第一部分：

应用加载——读取web.xml配置文件——初始化核心控制——struts.xml

第二部分：

浏览器——tomcat服务器——我们的应用——web.xml——StrutsPrepareAndExecuteFilter——HelloAction——执行sayHello方法——比较方法返回值和配置文件中是否一致——执行success.jsp——响应浏览器

此处必须明确的事情：

我们的动作类是多例的。每个线程有自己的实例，所以它不会有线程安全问题。

struts2内部是如何执行的

明确：

不是为了掌握这张图，而是要明确我们在实际开发中需要做什么事情。

我们在使用**struts2**中天天做的就是写jsp,写动作类和写配置文件。

其中配置文件使我们的学习重点。