



# 今克

## Java EE框架 --- Struts2

Java EE framework --Struts2

王磊

研发部

# CONTENTS



创建JavaWeb项目



搭建Struts2 开发环境



配置struts.xml



部署运行

# • 什么是Struts2框架 •

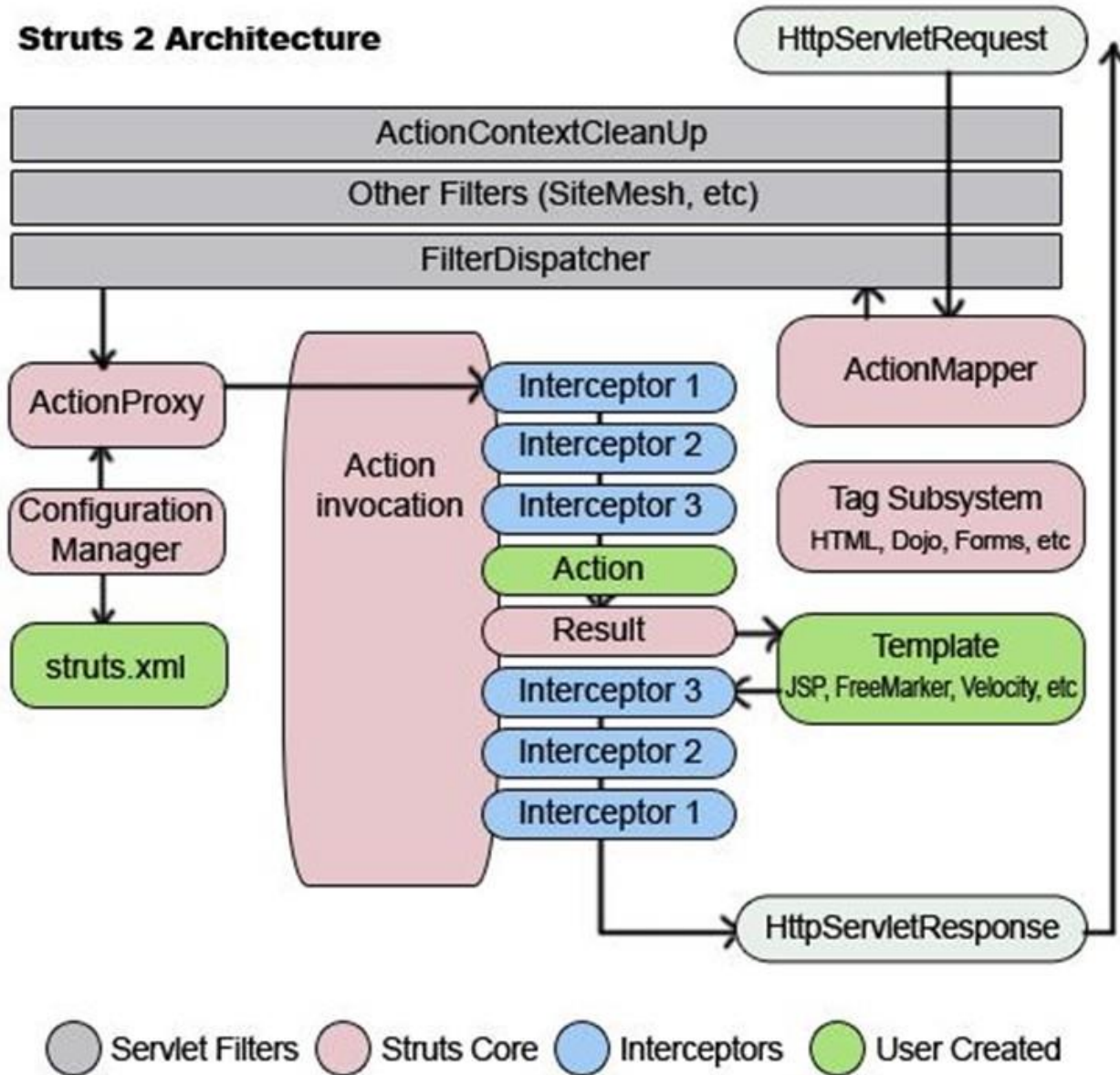
Apache Struts2 是一种流行的 Java模型 - 视图 - 控制器 ( MVC ) 框架，结合了 WebWork和 Struts1.x 两种 web 框架。

Apache Struts2与Struts1完全不同，它的核心功能是使用拦截实现“值栈”的概念，OGNL表达式和Struts2标签来解决应用程序数据，很多注解和约定使得框架更加易于使用。

Struts2优点：

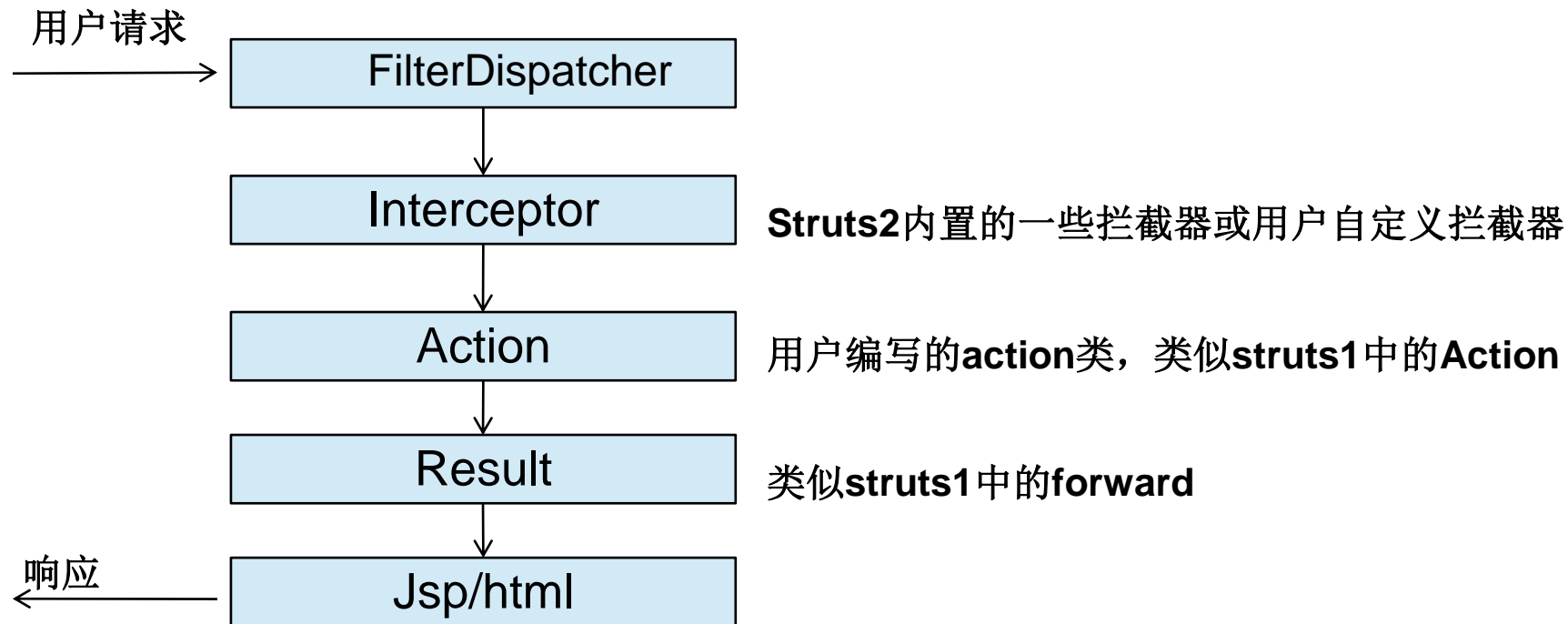
- 1、Struts2的应用可以不依赖于Servlet API和struts API。 Struts2的这种设计属于非侵入式设计，而Struts1却属于侵入式设计。
- 2、Struts2提供了拦截器，实现如权限拦截等功能。
- 3、Strut2提供了类型转换器，我们可以把特殊的请求参数转换成需要的类型。
- 4、Struts2提供支持多种表现层技术，如：JSP、freeMarker、Velocity等。
- 5、Struts2的输入校验可以对指定方法进行校验，解决了Struts1长久之痛。
- 6、提供了全局范围、包范围和Action范围的国际化资源文件管理实现。

# • Struts2体系结构 •












## • Struts2的处理流程 •

**FilterDispatcher**是Struts 2框架的核心控制器，它负责拦截由<url-pattern>/\*</url-pattern>指定的所有用户请求，当用户请求到达时，该Filter会过滤用户的请求。默认情况下，如果用户请求的路径不带后缀或者后缀以.action结尾，这时请求将被转入Struts 2框架处理，否则Struts 2框架将略过该请求的处理。当请求转入Struts 2框架处理时会先经过一系列的拦截器，然后再到Action。



# • 一、创建JavaWeb项目

- 1、新建Dynamic Web Project
- 2、引入 最小Struts2 jar包
- 3、构建路径


- ✓  HelloWorld\_Struts2
  - >  Deployment Descriptor: HelloWorld\_Struts2
  - >  JAX-WS Web Services
  - >  Java Resources
  - >  JavaScript Resources
  - >  build
  - ✓  WebContent
    - >  META-INF
    - >  WEB-INF

## 二、搭建Stust2 FrameWork开发环境

加入搭建Struts2 FrameWork开发环境的必需Jar包

▼  lib

 commons-fileupload-1.3.1.jar

 commons-io-2.2.jar

 commons-lang3-3.1.jar

 commons-logging-1.1.3.jar

 freemarker-2.3.19.jar

 javassist-3.11.0.GA.jar

 ognl-3.0.6.jar

 struts2-core-2.3.16.1.jar

 xwork-core-2.3.16.1.jar

 web.xml

## 二、搭建Struts2 Framework开发环境

src目录下创建“struts.xml”文件  
在struts.xml文件中加入如下配置信息

```
struts.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE struts PUBLIC
3     "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
4     "http://struts.apache.org/dtds/struts-2.0.dtd">
5
6 <struts>
7     <constant name="struts.devMode" value="true" />
8     <package name="HelloWorld_Struts2" extends="struts-default">
9         <action name="index">
10             <result>/index.jsp</result>
11         </action>
12     </package>
13 </struts>
```



## • 二、搭建Struts2 Framework开发环境 •

配置 `FilterDispatcher` 核心控制器

`FilterDispatcher`控制器负责拦截所有的客户端请求，然后 通过读取配置文件来确定交给哪个Action继续处理。

`FilterDispatcher`只需要在web.xml文件中配置一下。`init()`方法中将会读取类路径下默认的配置文件的完成初始化操作

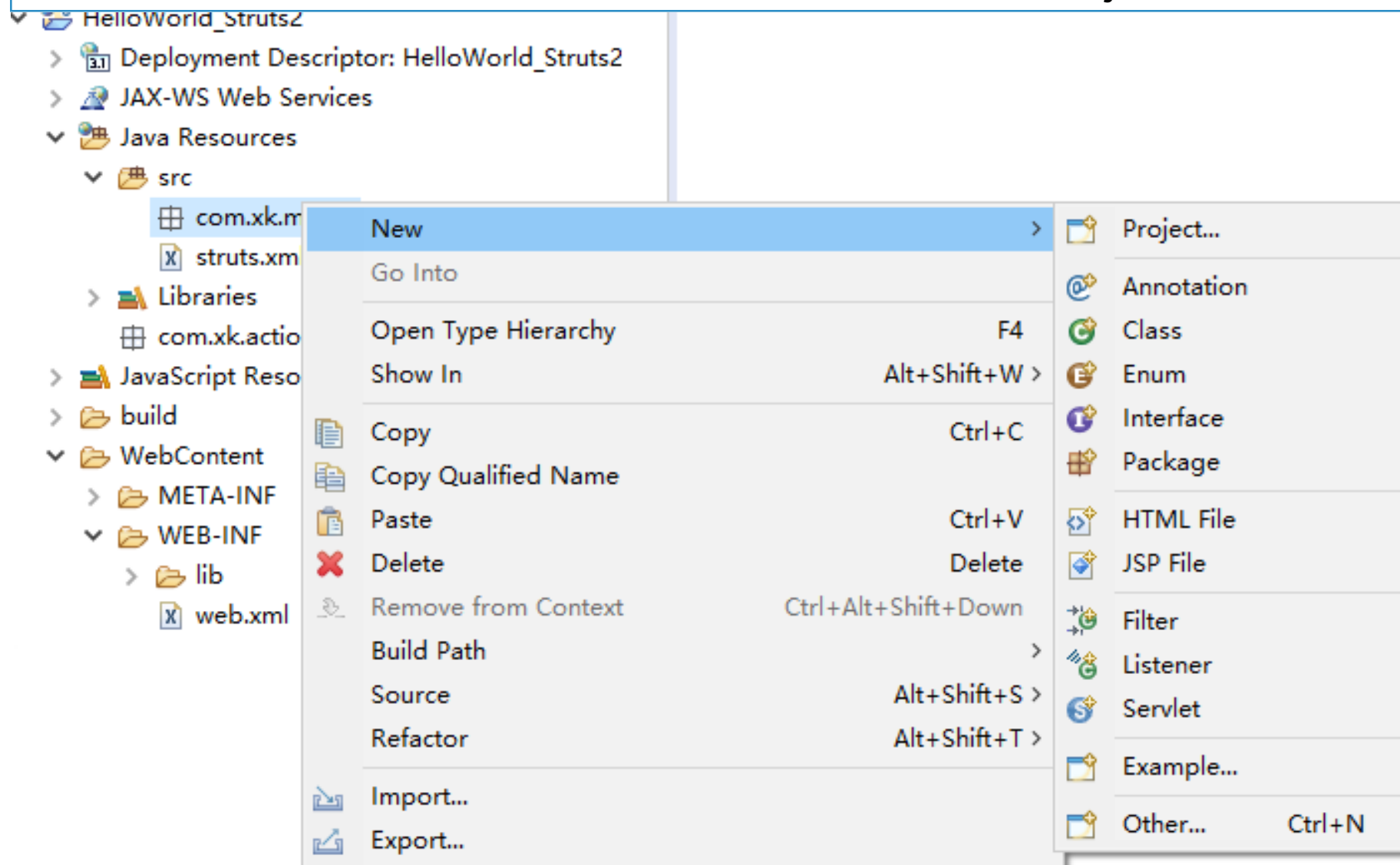
## 二、搭建Struts2 Framework开发环境

在web.xml文件中配置Struts2的核心过滤器(红色部分)

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_3_1.xsd" id="WebApp_ID" version="3.1">
  <display-name>HelloWorld_Struts2</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    .....
  </welcome-file-list>
  <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>
</web-app>
```

### 三、创建Model类User

在项目的src文件夹下添加一个“com.xk.model”包，然后添加User.java文件



### 三、创建Model类User

打开 “Message.java”，编写如下代码：

```
package com.xk.model;
public class User {
    private String username;
    private Integer age;
    public String getUsername() {
        return username;
    }
    public void setUsername(String username) {
        this.username = username;
    }
    public Integer getAge() {
        return age;
    }
    public void setAge(Integer age) {
        this.age = age;
    }
}
```

## 四、创建GetUserAction Action类

New Java Class

Java Class

Create a new Java class.

Source folder: HelloWorld\_Struts2/src Browse...

Package: com.xk.action Browse...

☐ Enclosing type: Browse...

Name: GetUserAction

Modifiers: ☒ public ☐ package ☐ private ☐ protected  
☐ abstract ☐ final ☐ static

Superclass: java.lang.Object Browse...

Interfaces: com.opensymphony.xwork2.Action Add... Remove

Which method stubs would you like to create?

☐ public static void main(String[] args)

☐ Constructors from superclass

☒ Inherited abstract methods

Do you want to add comments? (Configure templates and default value [here](#))














☐ Generate comments

Finish Cancel

在项目的src文件夹下添加一个 “GetUserAction.java”文件，放到 “com.xk.action”包中。  
注意：  
com.opensymphony.xwork2.Action继承自ActionSupport接口。

## 四、创建GetUserAction Action类

点击Finish完成添加，此时在src文件夹下就可以看到 “GetUserAction.java”文件

- ▼  HelloWorld\_Struts2
  - >  Deployment Descriptor: HelloWorld\_Struts2
  - >  JAX-WS Web Services
  - ▼  Java Resources
    - ▼  src
      - ▼  com.xk.action
        - >  GetUserAction.java
      - ▼  com.xk.model
        - >  User.java
      -  struts.xml
    - >  Libraries
  - >  JavaScript Resources
  - >  build

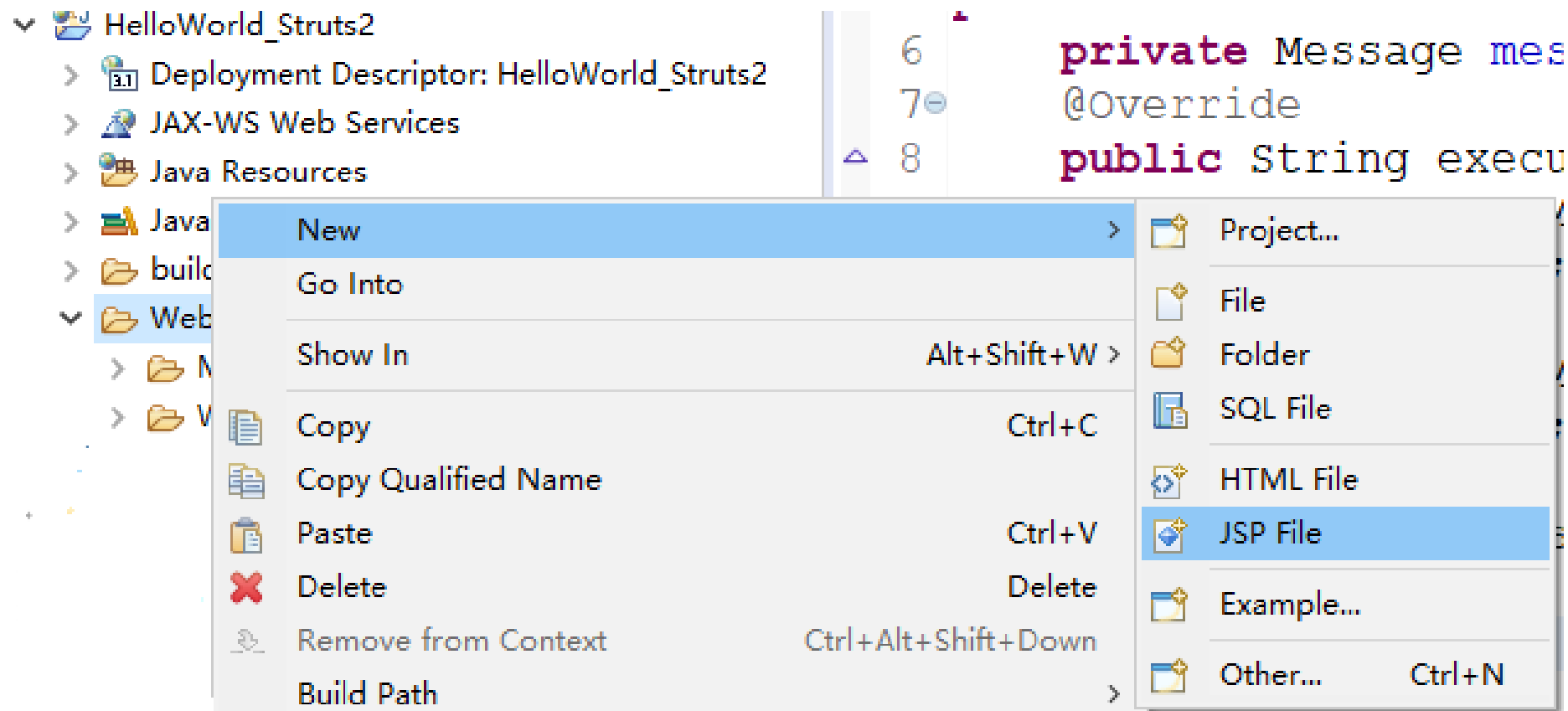
## 四、创建GetUserAction Action类

在 “GetUserAction.java”文件中编写如下代码：

```
package com.xk.action;
import com.opensymphony.xwork2.Action;
import com.opensymphony.xwork2.ActionContext;
import com.xk.model.User;
public class GetUserAction implements Action {
    private User user = new User();
    @Override
    public String execute() throws Exception {
        user.setUsername("测试用户1");
        user.setAge(22);
        User user2 = new User();
        user2.setUsername("测试用户2");
        user2.setAge(33);
        ActionContext ac = ActionContext.getContext();
        ac.put("user", user);
        ac.put("user2", user2);
        return "success";
    }
    public User getUser() {
        return user;
    }
    public void setUser(User user) {
        this.user = user;
    }
}
```

## • 五、创建View(视图)main.jsp •

在WebContent文件夹下创建”main.jsp”文件





## 五、创建View(视图)main.jsp

在main.jsp编写如下代码

```
<body>
<s:debug/>
<hr/>
```

使用EL表达式取出user和user2两个用户的信息: <br/>

```
<h2> user的姓名: ${user.username} &nbsp;&nbsp;&nbsp; user的年龄: ${user.age}</h2>
<h2> user2的姓名: ${user2.username} &nbsp;&nbsp;&nbsp; user2的年龄: ${user2.age}</h2>
```

```
<hr/>
```

使用s标签取出user和user2两个用户的信息: <br/>

```
<h2>user的姓名: <s:property value="user.username" />&nbsp;&nbsp;&nbsp;
    user的年龄: <s:property value="user.age" />
</h2>
<h2>user的姓名: <s:property value="#user.username" />&nbsp;&nbsp;&nbsp;
    user的年龄: <s:property value="#user.age" />
</h2>
<hr/>
<h2><font color="red">user2的姓名: <s:property value="#user2.username" /></font></h2>
<h2><font color="red">user2的年龄: <s:property value="#user2.age" /></font></h2>
```

```
<hr/>
</body>
```

## 六、查看Debug

查看Debug

[Debug]

### Struts ValueStack Debug

#### Value Stack Contents

Object	Property Name	Property Value
com.xk.action.GetUserAction	user	com.xk.model.User@7d2ddcd6
com.opensymphony.xwork2.DefaultTextProvider	texts	null

user在Value Stack Contents中，可以直接用s标签或EL表达式取值

## 六、查看Debug

### 查看Debug

#### Stack Context

*These items are available using the #key notation*

User和user2也被置入Stack Context中，用Struts2标签取值，必须加上#前缀

current.property.path	null
user2	com.xk.model.User@175b2e7
com.opensymphony.xwork2.ActionContext.actionInvocation	com.opensymphony.xwork2.DefaultActionInvocation@4fd821a5
com.opensymphony.xwork2.util.ValueStack.ValueStack	com.opensymphony.xwork2.ognl.OgnlValueStack@5023cbd6
com.opensymphony.xwork2.dispatcher.HttpServletResponse	org.apache.catalina.connector.ResponseFacade@5691d4dc
com.opensymphony.xwork2.ActionContext.conversionErrors	{}
parameters	{}
user	com.xk.model.User@7d2ddcd6

## • 七、配置struts.xml •

在struts.xml中加入红色部分的配置信息

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
    "http://struts.apache.org/dtds/struts-2.0.dtd">
<struts>
    <constant name="struts.devMode" value="true" />
    <package name="Hello_World_Struts2" extends="struts-default">
        <action name="index">
            <result>/index.jsp</result>
        </action>
        <action name="getUser"
            class="com.xk.action. GetUserAction" method="execute">
            <result name="success">/main.jsp</result>
        </action>
    </package>
</struts>
```

默认只有一个execute方法的时候，可以省略

## 八、创建Action的URL链接

在index.jsp中添加Action的URL链接,index.jsp中的代码如下：

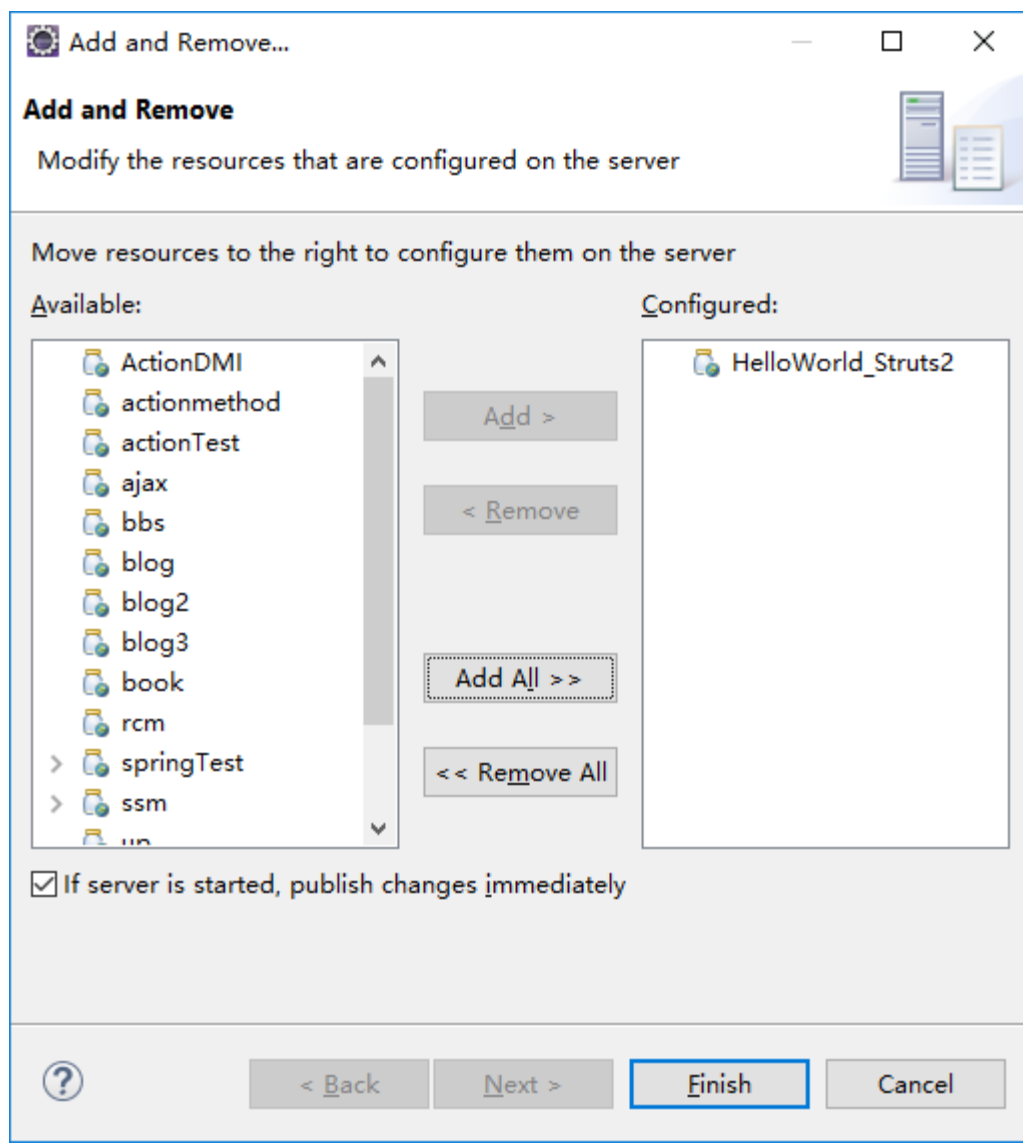
```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="s" uri="/struts-tags" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http:
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Struts 2 Application</title>
</head>
<body>
    <h1>Welcome To Struts2!</h1>
    <p><a href="<s:url action='getUser'/">">用不同的方式取出用户信息</a></p>
</body>
</html>
```

引入struts2标签

“<s:url action>”是一个Struts2的标签，用于创建Action的URL链接，“<s:url action='getUser'/">”表示链接到一个名字为 “getUser”的action。

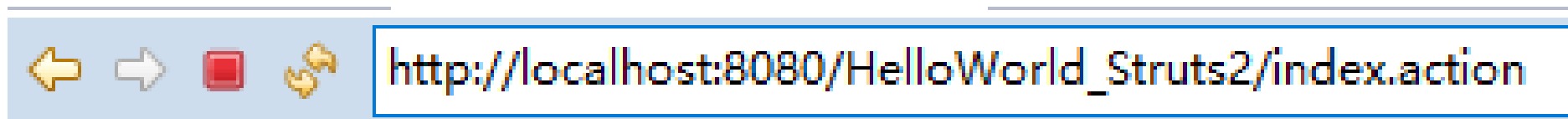
## 九、部署运行

将项目发布到Tomcat服务器中，如图所示



## • 十、部署运行 •

打开浏览器，在浏览器中输入[http://localhost:8080/HelloWorld\\_Struts2/index.action](http://localhost:8080/HelloWorld_Struts2/index.action)，此时就可以看到运行效果，如下图所示：



# Welcome To Struts2!

用不同的方式取出用户信息

## 十一、部署运行

点击“用不同的方式取出用户信息”超链接，此时看到的效果如下：



http://localhost:8080/HelloWorld\_Struts2/getUser.action

[\[Debug\]](#)

使用EL表达式取出user和 user2两个用户的信息：

user的姓名: `${user.username}`

user的姓名：测试用户1    user的年龄：22

user2的姓名：测试用户2    user2的年龄：33

使用s标签取出user和 user2两个用户的信息：

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

user的姓名：测试用户1    user的年龄：22

`<s:property value="#user.username" />`

user的姓名：测试用户1    user的年龄：22

`<s:property value="#user2.username" />`

user2的姓名：测试用户2

user2的年龄：33

`<s:property value="user2.username" />`



## • 作业 •

- 1、在windows实体机或者虚拟机中配置ssm开发环境
- 2、理解MVC架构体系，掌握Struts2替代Servlet的开发模式。
- 3、为充分理解Struts2框架，要求先使用model2模式进行设计，即项目中使用servlet充当控制器，对比本章案例。
- 4、编写Hello\_Struts2程序。
- 5、陈述项目开发流程。