

struts2 补充教程 1-基础入门

2020/11

教程大纲:

1. struts2 的基本概念
2. 应用 struts2 框架的第 1 个项目
3. 简单分析 struts2 的运作流程
4. struts2 配置文件加载
5. struts.xml 配置文件解析
6. struts2 常量文件修改
7. 动作类的三种写法

1. struts2 的基本概念

1.1 struts2 简介

- struts2 是一个 apache 基金会的一个开源项目（open source project）

<https://struts.apache.org/>



● 百度百科—struts2

Struts 2

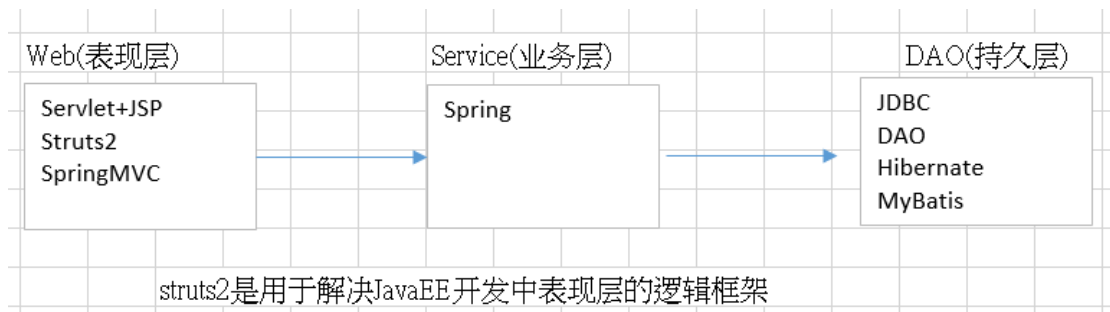
编辑

同义词 Struts2一般指Struts 2

Struts2是一个基于MVC设计模式的Web应用框架，它本质上相当于一个servlet，在MVC设计模式中，Struts2作为控制器(Controller)来建立模型与视图的数据交互。Struts 2是Struts的下一代产品，是在 struts 1和WebWork的技术基础上进行了合并的全新的Struts 2框架。其全新的Struts 2的体系结构与Struts 1的体系结构差别巨大。Struts 2以WebWork为核心，采用拦截器的机制来处理用户的请求，这样的设计也使得业务逻辑控制器能够与ServletAPI完全脱离开，所以Struts 2可以理解为WebWork的更新产品。虽然从Struts 1到Struts 2有着非常大的变化，但是相对于WebWork，Struts 2的变化很小。

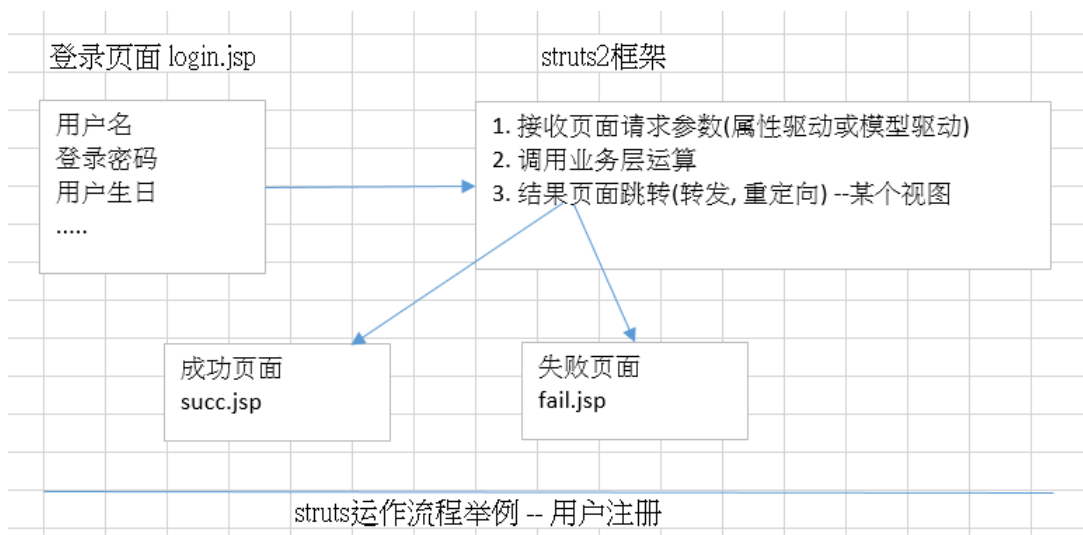
1.2 struts2 框架角色

- struts2 是一个基于 MVC 的表现层框架，角色说明如下：



- SSH 框架—— struts + spring + hibernate 的一个集成框架


- struts2 使用举例—用户注册为例:



2. 应用 struts2 框架的第 1 个项目

2.1 下载 struts2 开发包

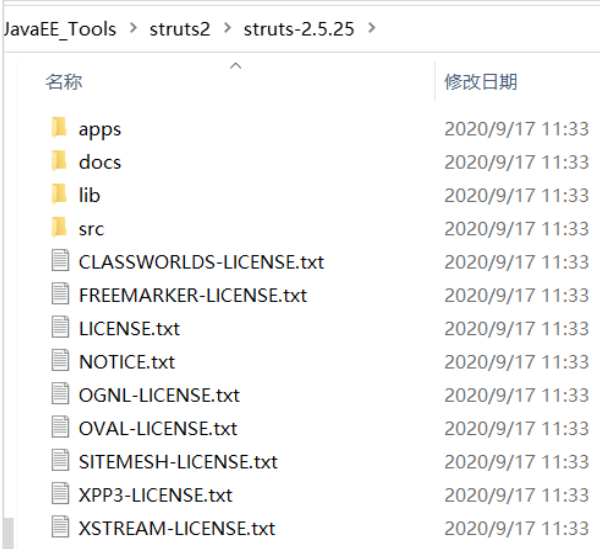
- <https://struts.apache.org/> -> Download



The screenshot shows the Apache Struts 2.5.25 download page. The page has a blue header with the Struts logo and navigation links: Home, Support, Documentation, and Contributing. Below the header, the title "Struts 2.5.25" is followed by a description: "Apache Struts 2.5.25 is an elegant, extensible framework for creating enterprise-ready Java applications. It is available as a full distribution, or as separate library, source, example and documentation distribution available" version of Struts in the 2.5 series.

- [Version Notes](#)
- Full Distribution:
 - [struts-2.5.25-all.zip \(65MB\)](#) [PGP] [SHA256]
- Example Applications:
 - [struts-2.5.25-apps.zip \(35MB\)](#) [PGP] [SHA256]
- Essential Dependencies Only:
 - [struts-2.5.25-min-lib.zip \(4MB\)](#) [PGP] [SHA256]

- Struts2-2.5.25-all.zip 解压后



The screenshot shows a file explorer window with the path "JavaEE_Tools > struts2 > struts-2.5.25". The table lists the files and folders extracted from the zip file, all with a modification date of 2020/9/17 11:33.

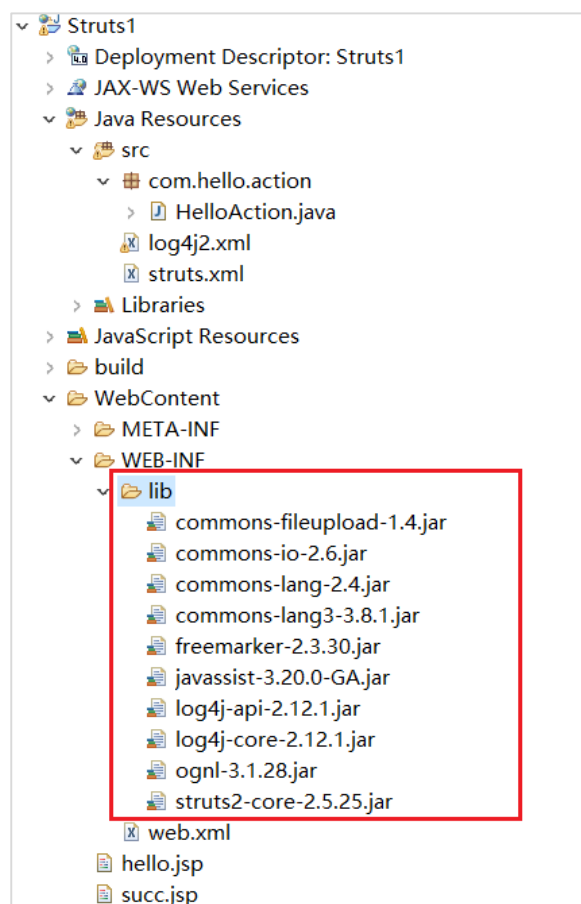
名称	修改日期
apps	2020/9/17 11:33
docs	2020/9/17 11:33
lib	2020/9/17 11:33
src	2020/9/17 11:33
CLASSWORLDS-LICENSE.txt	2020/9/17 11:33
FREEMARKER-LICENSE.txt	2020/9/17 11:33
LICENSE.txt	2020/9/17 11:33
NOTICE.txt	2020/9/17 11:33
OGNL-LICENSE.txt	2020/9/17 11:33
OVAL-LICENSE.txt	2020/9/17 11:33
SITEMESH-LICENSE.txt	2020/9/17 11:33
XPP3-LICENSE.txt	2020/9/17 11:33
XSTREAM-LICENSE.txt	2020/9/17 11:33

- 开发 struts2 项目所需的最小开发 JAR 包(10 个):

课件 > struts2 > struts2所需的JAR包			▼	🔄
名称	状态	修改日期		
 commons-fileupload-1.4.jar	✓	2020/10/30 23:53		
 commons-io-2.6.jar	✓	2020/10/30 23:53		
 commons-lang-2.4.jar	✓	2020/10/30 23:53		
 commons-lang3-3.8.1.jar	✓	2020/10/30 23:53		
 freemarker-2.3.30.jar	✓	2020/10/30 23:53		
 javassist-3.20.0-GA.jar	✓	2020/10/30 23:53		
 log4j-api-2.12.1.jar	✓	2020/10/30 23:53		
 log4j-core-2.12.1.jar	✓	2020/10/30 23:53		
 ognl-3.1.28.jar	✓	2020/10/30 23:53		
 struts2-core-2.5.25.jar	✓	2020/10/30 23:53		

2.2 准备工作一

- 创建 Web 项目，如 Struts1，并将 struts2 最小开发包导入 WEB 项目中



2.3 写码 1—编写 JSP 网页 (hello.jsp 及 succ.jsp)

- hello.jsp (欢迎页面)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Struts2的入门程序</title>
</head>
<body>
    <a href="hello.action">跳转到struts2的动作类</a>
</body>
</html>
```

- succ.jsp (运行成功的页面)

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>结果页面</title>
</head>
<body>
    struts2第一个hello项目已执行成功!
</body>
</html>
```

2.4 写码 2—编写 struts2 的 Action 类

- 在 src 下, com.hello.action, 创建 HelloAction 类

```
package com.hello.action;

public class HelloAction {
    public String hello() {
        System.out.println("进入Struts2的Action类的hello()方法!");
        return "success";
    }
}
```

2.5 写码 3—配置 Action 类(写 struts.xml)

- 文件名: struts.xml, 位置: src 目录 (默认位置)。可以将 struts2 示范项目 struts2-rest-showcase 中的 struts.xml 复制过来再修改。

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

<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.5//EN"
    "http://struts.apache.org/dtds/struts-2.5.dtd">

<struts>
    <package name="base" extends="struts-default" namespace="/">
```

```

        <action name="hello" class="com.hello.action.HelloAction"
method="hello">
            <result name="success">/succ.jsp</result>
        </action>
    </package>
</struts>

```

2.6 写码 4—在 web.xml 中加入 struts 过滤器

- 在 web.xml 中增加 struts 过滤器语句。可以将 struts2 示范项目 struts2-rest-showcase 中的 web.xml 中语句复制过来即可。

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3     xmlns="http://xmlns.jcp.org/xml/ns/javaee"
4     xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
5     id="WebApp_ID" version="4.0">
6     <display-name>Struts1</display-name>
7
8     <!-- struts2的启动程序 -->
9     <filter>
10         <filter-name>struts2</filter-name>
11         <filter-class>org.apache.struts2.dispatcher.filter.StrutsPrepareAndExecuteFilter</filter-class>
12     </filter>
13     <filter-mapping>
14         <filter-name>struts2</filter-name>
15         <url-pattern>/*</url-pattern>
16     </filter-mapping>
17
18     <welcome-file-list>
19         <welcome-file>index.html</welcome-file>
20         <welcome-file>index.jsp</welcome-file>
21     </welcome-file-list>
22 </web-app>

```

加入这些语句

```

<!-- struts2的启动程序 -->
<filter>
    <filter-name>struts2</filter-name>
    <filter-
class>org.apache.struts2.dispatcher.filter.StrutsPrepareAndExecuteFilter<
/filter-class>
</filter>
<filter-mapping>
    <filter-name>struts2</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>

```

2.7 写码 5—加入 log4j2.xml

- 文件名: log4j2.xml，摆放位置: src 目录 (默认位置)。可以将 struts2 示范项目 struts2-rest-showcase 中的 log4j2.xml 直接复制过使用即可。

```

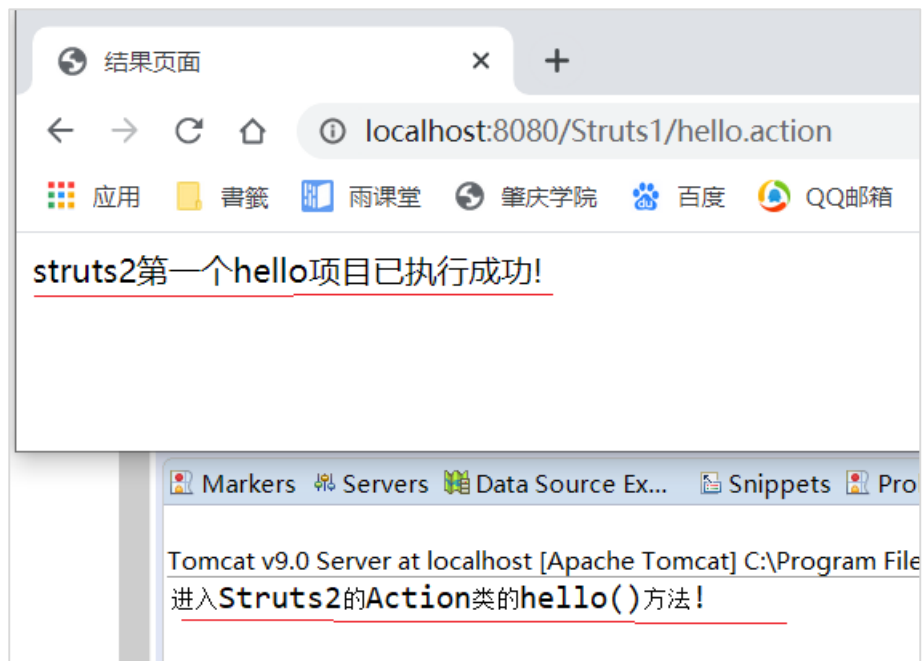
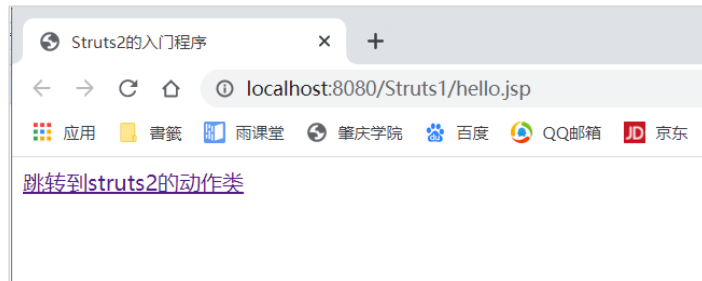
<?xml version="1.0" encoding="UTF-8"?>

<Configuration>
    <Appenders>
        <Console name="STDOUT" target="SYSTEM_OUT">
            <PatternLayout pattern="%d %-5p [%t] %C{2} (%F:%L) - %m%n"/>
        </Console>
    </Appenders>
    <Loggers>
        <Logger name="com.opensymphony.xwork2" level="info"/>
        <Logger name="org.apache.struts2" level="info"/>
        <Logger name="org.springframework" level="info"/>
        <Root level="info">

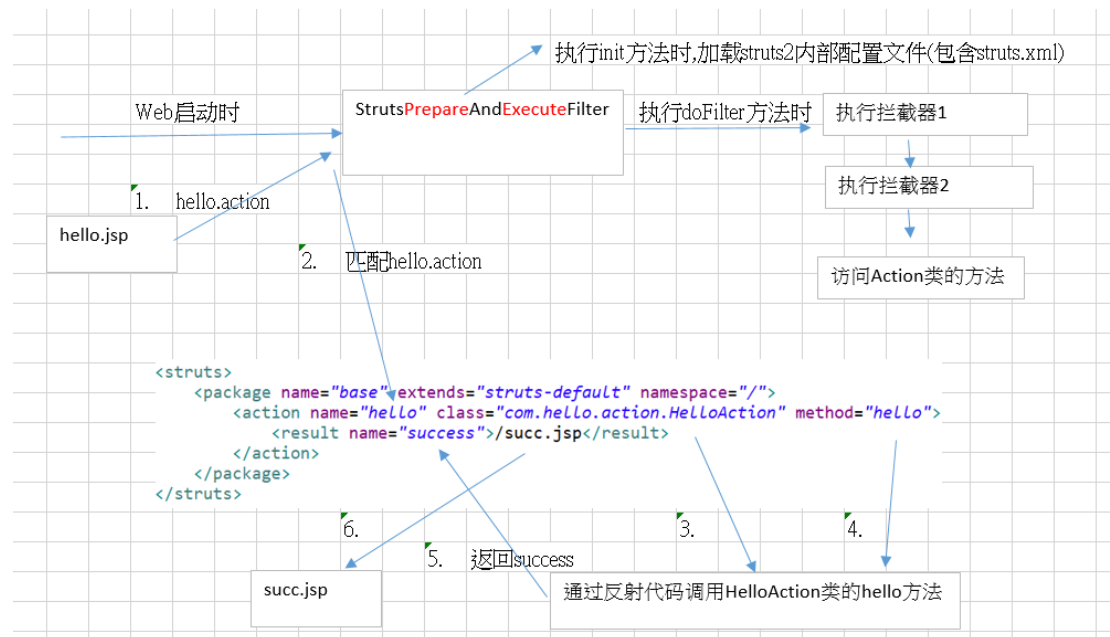
```

```
<AppenderRef ref="STDOUT"/>
</Root>
</Loggers>
</Configuration>
```

2.8 运行测试—运行 hello.jsp



3. 简单分析 struts2 的运作流程



4. struts2 配置文件加载

- struts2 的初始化过程中（在核心过滤器调用 init 方法的时候）加载 6 个配置文件：

- 1) default.properties：常量文件
- 2) struts-default.xml：struts2 的核心配置（拦截器等）
- 3) struts-plugin.xml：struts2 的插件配置（这个文件包含在插件包里面）
- 4) struts.xml（开发者自己编写的）：struts2 的配置
- 5) struts.properties：struts2 的配置（比较少见）
- 6) web.xml：web 部署描述文件

注意：以上 6 个文件加载过程中，后面加载的文件会覆盖前面加载的文件参数

5. struts.xml 配置文件解析

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

<!DOCTYPE struts PUBLIC
  "-//Apache Software Foundation//DTD Struts Configuration 2.5//EN"
  "http://struts.apache.org/dtds/struts-2.5.dtd">

<struts>
  <package name="base" extends="struts-default" namespace="/">
    <action name="hello" class="com.hello.action.HelloAction"
method="hello">
      <result name="success"/>succ.jsp</result>
    </action>
  </package>
</struts>
```



```
</action>
</package>
</struts>
```

- **package** 标签：用于管理不同的 Action。name 属性:包的名；extends 属性:继承，上例我们继承了 **struts-default** 包，也就有了所有父包的特征；namespace 属性:命名空间,区分不同 package 的访问路径。
- **action** 标签：配置动作类。name 属性:动作类的名称,默认加上.action 为调用名称；class 属性:动作类的类名；method 属性:动作类的执行方法,默认为 excute 方法。
- **result** 标签：配置视图,用于跳转页面。name 属性:视图名称,用于动作类返回值进行匹配,默认值 success；type 属性:视图类型及如何跳转,跳转有分转发或重定向,默认值 dispatcher (转发)。

6. struts2 常量文件修改

- default.properties：常量文件

```
struts.action.extension=action,,
```

这个是 struts2 的访问后缀名配置，默认值为.action

位置：struts2-core-2.5.25.jar 下，/org/apache/struts2/default.properties

7. 动作类的三种写法

7.1 方式 1：不实现接口和继承

```
public class Demo1Action {
    public String hello(){
        System.out.println("方式 1：不实现接口和继承");
        return "success";
    }
}
```

7.2 方式 2：实现 Action 接口

```
public class Demo2Action implements Action{
    //execute:是 struts2 的 Action 类的默认方法
    @Override
    public String execute() throws Exception {
        System.out.println("方式 2：实现 Action 接口");
    }
}
```

```
        return SUCCESS; //SUCCESS 是常量
    }
}
```

7.3 方式 3：继承 ActionSupport 类（推荐使用）

```
public class Demo3Action extends ActionSupport{
    @Override
    public String execute() throws Exception {
        System.out.println("方式 3：继承 ActionSupport 类");
        return SUCCESS; //SUCCESS 是常量
    }
}
```