

Strtus2 系列课程

时间: 2017/11/27

作者:袁毅雄

邮箱: 896778954@qq.com

目录

Strtus2 系列课程	1
目录.....	1
Strtus2 是什么	2
Strtus2 MVC 构架	3
Strtus2MVC 体系结构	3
Strtus2 实现 Hello World.....	4
引入 jar	4
web.xml.....	4
struts.xml	4
MyAction.java	5
index.jsp	5
log4j2.xml.....	5
Strtus2 Action 的动作.....	6
源码分析.....	6
Strtus2 常量.....	6
源码分析:.....	6
常量的配置.....	7
动态方法的运用.....	7
Strtus2 模型驱动	9
传入.....	9
传出.....	12
Strtus2 多种 result 处理.....	14
1. dispatcher(默认的)	14
2. chain.....	14
3. redirect	15
4. redirectAction.....	16
5. stream	17
6. json.....	19
Strtus2 获取 JSP 常用的内置对象.....	21
解耦合.....	21
耦合.....	22
Strtus2 interceptors	23
Strtus2 系统拦截器	23

Strtus2 自定义拦截器	25
Strtus2 Ajax 支持	27
Strtus2 文件上传	27
单文件上传	27
多文件上传	31
异步文件上传	35
Strtus2 文件下载	39
Strtus2 异常处理	39
全局异常处理	39
局部异常处理	41

时间: 2017/11/22

作者: 袁毅雄

邮箱: 896778954@qq.com

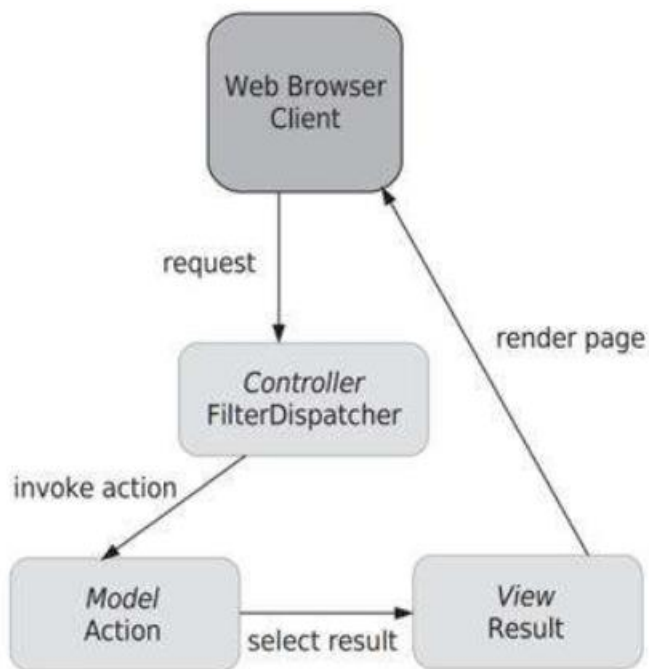
Strtus2 是什么

Strtus2 是基于 MVC 设计模式的流行和成熟的 Web 应用程序框架。Strtus2 并不只是 Struts 1 下一个版本，但它是一个完全重写的 Struts 架构。

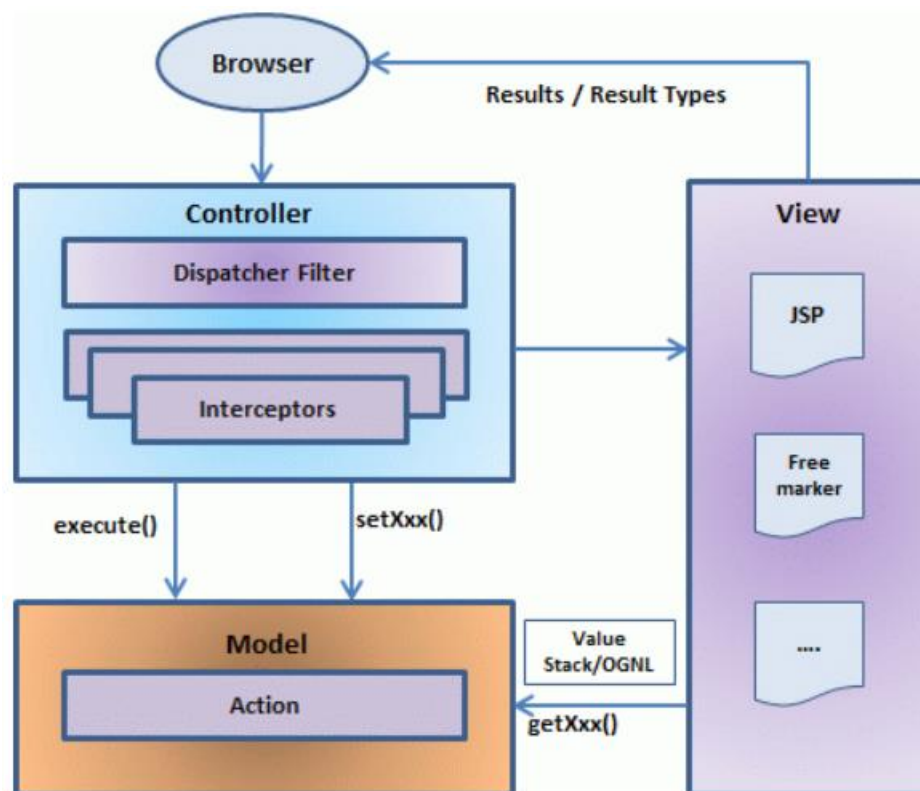
WebWork 框架开始了与 Struts 框架为基础，其目标是建立在 Struts 的 Web 开发更容易为开发人员提供了加强和改进的框架。

一段时间后，WebWork 框架和 Struts 社区联手打造著名的 Strtus2 框架。

Strtus2 MVC 构架








Strtus2MVC 体系结构



Strtus2 实现 Hello World

引入 jar

struts-2.3.24.1\apps\struts2-blank.war\WEB-INF\lib 目录下的所有 jar 包				
	asm-3.3.jar	2013/11/23 17:55	Executable Jar File	43 KB
	asm-commons-3.3.jar	2013/11/23 17:55	Executable Jar File	38 KB
	asm-tree-3.3.jar	2013/11/23 17:55	Executable Jar File	21 KB
	commons-fileupload-1.3.1.jar	2014/02/19 16:21	Executable Jar File	68 KB
	commons-io-2.2.jar	2013/11/23 17:55	Executable Jar File	170 KB
	commons-lang3-3.2.jar	2014/01/02 21:45	Executable Jar File	376 KB
	freemarker-2.3.22.jar	2015/04/03 7:09	Executable Jar File	1,271 KB
	javassist-3.11.0.GA.jar	2013/11/23 17:55	Executable Jar File	600 KB
	log4j-api-2.2.jar	2015/04/19 12:04	Executable Jar File	131 KB
	log4j-core-2.2.jar	2015/04/19 12:04	Executable Jar File	808 KB
	ognl-3.0.6.jar	2013/11/23 17:55	Executable Jar File	223 KB
	struts2-core-2.3.24.1.jar	2015/09/22 7:43	Executable Jar File	813 KB
	xwork-core-2.3.24.1.jar	2015/09/22 7:42	Executable Jar File	662 KB

web.xml

```
<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>
```

struts.xml

```
<?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">
<struts>
  <package name="default" namespace="/" extends="struts-default">
    <action name="index" class="online.qsx.action.MyAction">
      <result name="success">/WEB-INF/jsp/index.jsp</result>
    </action>
  </package>
</struts>

```

MyAction.java

```

package online.qsx.action;
import com.opensymphony.xwork2.Action;
public class MyAction implements Action {
    @Override
    public String execute() throws Exception {
        System.out.println("Hello World");
        return this.SUCCESS;
    }
}

```

index.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE >
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    Hello World
</body>
</html>

```

log4j2.xml

```

<?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="debug"/>
  <Logger name="org.apache.struts2" level="debug"/>
  <Root level="warn">
    <AppenderRef ref="STDOUT"/>
  </Root>
</Loggers>
</Configuration>
```

Strtus2 Action 的动作

源码分析

```
package com.opensymphony.xwork2;

public interface Action {

    public static final String SUCCESS = "success";
    public static final String NONE = "none";
    public static final String ERROR = "error";
    public static final String INPUT = "input";
    public static final String LOGIN = "login";

    public String execute() throws Exception;
}
```

Strtus2 常量

源码分析:

org.apache.struts2.**default.properties**

动态方法

```
104### Set this to false if you wish to disable implicit dynamic method invocation
105### via the URL request. This includes URLs like foo!bar.action, as well as params
106### like method:bar (but not action:foo).
107### An alternative to implicit dynamic method invocation is to use wildcard
108### mappings, such as <action name="*/" method="{2}" class="actions.{1}">
109struts.enable.DynamicMethodInvocation = false
```

开发者模式

```
130### when set to true, Struts will act much more friendly for developers. This
131### includes:
132### - struts.i18n.reload = true
133### - struts.configuration.xml.reload = true
134### - raising various debug or ignorable problems to errors
135### For example: normally a request to foo.action?someUnknownField=true should
136### be ignored (given that any value can come from the web and it
137### should not be trusted). However, during development, it may be
138### useful to know when these errors are happening and be told of
139### them right away.
140struts.devMode = false
```

常量的配置

struts.xml

```
<?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">
<struts>
    <!-- 常量 -->
    <constant name="struts.enable.DynamicMethodInvocation" value="true"/>
    <constant name="struts.devMode" value="true"/>
    <!-- 此处省略 -->
</struts>
```

动态方法的运用

strtus2.xml

```
<?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">
<struts>
    <!-- 常量 -->
    <constant name="struts.enable.DynamicMethodInvocation" value="true"/><!-- 动态方法 -->
    <constant name="struts.devMode" value="true"/><!-- 开发者模式 -->

    <package name="default" namespace="/" extends="struts-default">
        <action name="myaction!*" class="online.qsx.action.MyAction" method="{0}">
            <result name="index">/WEB-INF/jsp/index.jsp</result>
            <result name="add">/WEB-INF/jsp/add.jsp</result>
        </action>
    </package>
</struts>
```

MyAction.java

```
package online.qsx.action;

public class MyAction {
    //http://localhost/Strtus2_HelloWorld/myaction!to_index
    public String to_index() {
        System.out.println("index");
        return "index";
    }
    //http://localhost/Strtus2_HelloWorld/myaction!to_add
    public String to_add() {
        System.out.println("add");
        return "add";
    }
}
```

add.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html >
```



```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    add
</body>
</html>
```

index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html >
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    index
</body>
</html>
```

Strtus2 模型驱动

传入

User.java

```
package online.qsx.model;

public class User {

    private int id;

    private String name;

    public int getId() {return id;}

    public void setId(int id) {this.id = id;}

    public String getName() {return name;}
```

```

    public void setName(String name) {this.name = name;}

    public User() {super();}

    public User(int id, String name) {
        super();
        this.id = id;
        this.name = name;
    }

    @Override
    public String toString() {
        return "User [id=" + id + ", name=" + name + "];"
    }
}

```

MyAction.java

```

package online.qsx.action;
import java.util.ArrayList;
import java.util.List;
import online.qsx.model.User;
public class MyAction {

    private String mess;
    private User user;
    private List<User> users;

    public void setMess(String mess) {this.mess = mess;}
    public void setUser(User user) {this.user = user;}
    public void setUsers(List<User> users) {this.users = users;}

    //地址栏传入参数
    // mess
    // http://localhost/Strtus2_HelloWorld/myaction!put?mess=arvin

    // user
    // http://localhost/Strtus2_HelloWorld/myaction!put?user.id=1001&user.name=arvin

    // users
    // http://localhost/Strtus2_HelloWorld/myaction!put?users[1].id=1001&users[1].name=arvin

```

```

//表单传入参数

/*
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>

<!DOCTYPE html >

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<title>Insert title here</title>

<link rel="stylesheet" href="https://cdn.bootcss.com/bootstrap/4.0.0-beta/css/bootstrap.min.css">

<script src="https://cdn.bootcss.com/jquery/3.2.1/jquery.min.js"></script>

<script src="https://cdn.bootcss.com/popper.js/1.12.5/umd/popper.min.js"></script>

<script src="https://cdn.bootcss.com/bootstrap/4.0.0-beta/js/bootstrap.min.js"></script>

</head>

<body>

    <div class="container w-50 h-100">

        <form action="myaction!put" method="post">

            <div class="form-group">

                <label for="id">Id:</label>

                <input class="form-control" id="id" type="text" name="user.id"/>

            </div>

            <div class="form-group">

                <label for="name">Name</label>

                <input class="form-control" id="name" type="password" name="user.name"/>

            </div>

            <div class="form-group">

                <input class="btn btn-primary" type="submit" value="提交"/>

                <input class="btn btn-primary" type="reset" value="取消"/>

            </div>

        </form>

    </div>

</body>

</html>

*/

public String put() {

    System.out.println("mess:" + mess);

    System.out.println("user:" + user);

    System.out.println("users:" + users);

    return "index";

}

}

```

传出

User.java

```
package online.qsx.model;

public class User {

    private int id;
    private String name;

    public int getId() {return id;}
    public void setId(int id) {this.id = id;}

    public String getName() {return name;}
    public void setName(String name) {this.name = name;}

    public User() {super();}

    public User(int id, String name) {
        super();
        this.id = id;
        this.name = name;
    }

    @Override
    public String toString() {
        return "User [id=" + id + ", name=" + name + "];"
    }
}
```

MyAction

```
package online.qsx.action;

import java.util.ArrayList;
import java.util.List;
import online.qsx.model.User;

public class MyAction {

    private String mess;
    private User user;
    private List<User> users;
```

```

    public String getMess() {return mess;}
    public User getUser() {return user;}
    public List<User> getUsers() {return users;}

    //http://localhost/Strtus2_HelloWorld/myaction!out
    public String out() {
        mess = "阿文";

        user = new User(1001, "阿文");

        users=new ArrayList<User>();
        users.add(new User(1001, "阿文"));
        users.add(new User(1002, "杰克"));

        return "index";
    }
}

```

index.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html >
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    ${mess }
    <br/>
    ${user.id }&nbsp;${user.name }
    <br/>
    ${users }
</body>
</html>

```

Struts2 多种 result 处理

1. dispatcher(默认的)

转发到指定的页面,数据默认携带

```
<?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">
<struts>
    <!-- 常量 -->
    <constant name="struts.enable.DynamicMethodInvocation" value="true"/><!-- 动态方法 -->
    <constant name="struts.devMode" value="true"/><!-- 开发者模式 -->

    <package name="default" namespace="/" extends="struts-default">
        <action name="b!" class="online.qsx.action.B_Action" method="{0}">
            <!--转发到页面自动携带数据-->
            <result name="succeed" type="dispatcher" >/WEB-INF/jsp/index.jsp</result>
        </action>
    </package>
</struts>
```

2. chain

转发到指定的 Action,数据默认携带

```
<?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">
<struts>
    <!-- 常量 -->
    <constant name="struts.enable.DynamicMethodInvocation" value="true"/><!-- 动态方法 -->
    <constant name="struts.devMode" value="true"/><!-- 开发者模式 -->

    <package name="default" namespace="/" extends="struts-default">
        <action name="a!" class="online.qsx.action.A_Action" method="{0}">
            <!--转发到Action 自动携带数据-->
            <result name="succeed" type="chain">
                <param name="actionName">b</param><!-- 转发的Action -->
            </result>
        </action>
    </package>
</struts>
```

```

        <param name="method">to_b</param><!-- 转发的方法 -->
    </result>
</action>

    <action name="b!*" class="online.qsx.action.B_Action" method="${0}">
        <result name="succeed" >/WEB-INF/jsp/index.jsp</result>
    </action>
</package>

</struts>

```

3. redirect

重定向页面,不携带数据

```

<?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">
<struts>
    <!-- 常量 -->
    <constant name="struts.enable.DynamicMethodInvocation" value="true"/><!-- 动态方法 -->
    <constant name="struts.devMode" value="true"/><!-- 开发者模式 -->

    <package name="default" namespace="/" extends="struts-default">
        <action name="b!*" class="online.qsx.action.B_Action" method="${0}">
            <!-- 重定向到页面,无数据携带-->
            <result name="succeed" type="redirect" >/index.jsp</result>
        </action>
    </package>
</struts>

```

重定向页面,携带数据

```

<?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">
<struts>
    <!-- 常量 -->
    <constant name="struts.enable.DynamicMethodInvocation" value="true"/><!-- 动态方法 -->
    <constant name="struts.devMode" value="true"/><!-- 开发者模式 -->

```

```

<package name="default" namespace="/" extends="struts-default">
    <action name="b!" class="online.qsx.action.B_Action" method="{0}">
        <!-- 重定向到页面,携带数据 -->
        <result name="succeed" type="redirect" >/index.jsp?name=${name}</result>
    </action>
</package>
</struts>

```

4. redirectAction

重定向 Action,不携带数据

```

<?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">
<struts>
    <!-- 常量 -->
    <constant name="struts.enable.DynamicMethodInvocation" value="true"/><!-- 动态方法 -->
    <constant name="struts.devMode" value="true"/><!-- 开发者模式 -->

    <package name="default" namespace="/" extends="struts-default">
        <action name="a!" class="online.qsx.action.A_Action" method="{0}">
            <!-- 重定向到Action,无数据携带-->
            <result name="succeed" type="redirectAction" >
                <param name="actionName">b</param> <!-- 重定向的Action -->
                <param name="method">to_b</param><!-- 重定向的方法 -->
            </result>
        </action>
        <action name="b!" class="online.qsx.action.B_Action" method="{0}">
            <result name="succeed">/WEB-INF/jsp/index.jsp</result>
        </action>
    </package>
</struts>

```

重定向 Action,携带数据

```

<?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">

```



```

<struts>

    <!-- 常量 -->

    <constant name="struts.enable.DynamicMethodInvocation" value="true"/><!-- 动态方法 -->

    <constant name="struts.devMode" value="true"/><!-- 开发者模式 -->

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

        <action name="a!*" class="online.qsx.action.A_Action" method="{0}">

            <!-- 重定向到Action,无数据携带-->

            <result name="succeed" type="redirectAction" >

                <param name="actionName">b</param> <!-- 重定向的Action -->

                <param name="method">to_b</param><!-- 重定向的方法 -->

                <param name="name">${name}</param><!-- 重定向携带的数据 -->

            </result>

        </action>

        <action name="b!*" class="online.qsx.action.B_Action" method="{0}">

            <result name="succeed">/WEB-INF/jsp/index.jsp</result>

        </action>

    </package>

</struts>

```

5. stream

strtus2.xml

```

<?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">
<struts>

    <!-- 常量 -->

    <constant name="struts.enable.DynamicMethodInvocation" value="true" /><!-- 动态方法 -->

    <constant name="struts.devMode" value="true" /><!-- 开发者模式 -->

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

        <!-- http://localhost/Strtus2_HelloWorld/download!downloadFile -->

        <action name="download!*" class="online.qsx.action.DownloadAction" method="{0}">

            <!-- 以stream二进制流的方式打开 -->

            <result name="success" type="stream">

                <!-- 指明文件的下载类型 -->

                <param name="contentType">image/jpeg</param>

                <!-- 指明如果取得需要下载文件的InputStream输入流 -->

            </result>

        </action>

    </package>

</struts>

```

```

        <param name="inputName">imageStream</param>

        <!-- 指明让浏览器以下载框的方式打开 -->

        <param name="contentDisposition">attachment;filename=${fileName}</param>

        <!-- 指明下载文件时的字符数组byte[]大小 -->

        <param name="bufferSize">1024</param>

    </result>

</action>

</package>

</struts>

```

Download.java

```

package online.qsx.action;

import java.io.InputStream;
import java.net.URLEncoder;
import org.apache.struts2.ServletActionContext;

public class DownloadAction {

    String fileName = "IMG_20171103_105557.jpg";

    /**
     * 提供下载文件浏览器下载框提示的文件名
     * @return
     * @throws Exception
     */

    public String getFileName() throws Exception {

        return URLEncoder.encode(fileName, "utf-8");

    }

    /**
     * 提供下载文件的InputStream输入流
     * @return
     * @throws Exception
     */

    public InputStream getImageStream() throws Exception {

        return ServletActionContext.getServletContext().getResourceAsStream("/WEB-INF/download/" +
fileName);

    }

    /**
     * 下载请求
     * @return
     */

    public String downloadFile() {

```

```
        return "success";
    }
}
```

6. json

前置条件:

导入 **struts2-json-plugin-2.3.24.1.jar**

strtus2.xml

```
<?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">
<struts>
    <!-- 常量 -->
    <constant name="struts.enable.DynamicMethodInvocation" value="true" /><!-- 动态方法 -->
    <constant name="struts.devMode" value="true" /><!-- 开发者模式 -->

    <package name="default" namespace="/" extends="json-default">
        <!-- http://localhost/Strtus2_HelloWorld/json_1!ajax -->
        <action name="json_1!*" class="online.qsx.action.JSONAction" method="{0}">
            <result name="success" type="json">
                <!-- 默认返回该Action中所有有get方法的属性,自动转成JSON -->
            </result>
        </action>

        <!-- http://localhost/Strtus2_HelloWorld/json_2!ajax -->
        <action name="json_2!*" class="online.qsx.action.JSONAction" method="{0}">
            <result name="success" type="json">
                <!-- 指定该Action那些有get方法的属性,自动转成JSON -->
                <param name="includeProperties">
                    <!-- mess,user.*,users.* -->
                    users.*
                </param>
            </result>
        </action>
    </package>
</struts>
```

```

<!-- http://localhost/Strtus2_HelloWorld/json_3!ajax -->
<action name="json_3!*" class="online.qsx.action.JSONAction" method="{0}">
    <result name="success" type="json">
        <!-- 指定该Action除了那些有get方法的属性之外,其他的自动转成JSON -->
        <param name="excludeProperties">
            <!-- mess,user.*,users.* -->
            mess
        </param>
    </result>
</action>
</package>
</struts>

```

JSONAction.java

```

package online.qsx.action;
import java.util.ArrayList;
import java.util.List;
import online.qsx.model.User;
public class JSONAction {

    private String mess;
    private User user;
    private List<User> users;

    public String getMess() {return mess;}
    public User getUser() {return user;}
    public List<User> getUsers() {return users;}

    public String ajax() {
        mess = "arvin";

        user = new User(1, "arvin_01");

        users = new ArrayList<>();
        users.add(new User(2, "arvin_02"));
        users.add(new User(3, "arvin_03"));

        return "success";
    }
}

```

User.java

```
package online.qsx.model;

public class User {

    private int id;

    private String name;

    public int getId() {return id;}

    public void setId(int id) {this.id = id;}

    public String getName() {return name;}

    public void setName(String name) {this.name = name;}

    public User(int id, String name) {

        super();

        this.id = id;

        this.name = name;

    }

    public User() {

        super();

    }

}
```

Strtus2 获取 JSP 常用的内置对象

常用的jsp 内置对象：request,response,session,application

解耦合

局部变量

session: `ActionContext.getContext().getSession();`

request: `ActionContext.getContext().get("request");` (Map<String,Object>接收)

application: `ActionContext.getContext().getApplication();`

成员变量

session: 实现 SessionAware 接口

request: 实现 RequestAware 接口

application: 实现 ApplicationAware 接口

耦合

局部变量

session: ServletActionContext.getRequest().getSession();

request: ServletActionContext.getRequest();

response: ServletActionContext.getResponse();

application: ServletContext application=ServletActionContext.getServletContext();

成员变量

session: 实现 ServletRequestAware 接口 , 通过 request.getSession()来取到

request: 实现 ServletRequestAware 接口

response: 实现 ServletResponseAware 接口

application: 实现 ServletApplicationAware 接口

Strtus2 interceptors

Strtut2 系统拦截器

1.alias

Allows parameters to have different name aliases across requests.

2.checkbox

Assists in managing check boxes by adding a parameter value of false for check boxes that are not checked.

3.conversionError

Places error information from converting strings to parameter types into the action's field errors.

4.createSession

Automatically creates an HTTP session if one does not already exist.

5.debugging

Provides several different debugging screens to the developer.

6.execAndWait

Sends the user to an intermediary waiting page while the action executes in the background.

7.exception

Maps exceptions that are thrown from an action to a result, allowing automatic exception handling via redirection.

8.fileUpload

Facilitates easy file uploading.

9.i18n

Keeps track of the selected locale during a user's session.

10.logger

Provides simple logging by outputting the name of the action being executed.

11.params

Sets the request parameters on the action.

12.prepare

This is typically used to do pre-processing work, such as setup database connections.

13.profile

Allows simple profiling information to be logged for actions.

14.scope

Stores and retrieves the action's state in the session or application scope.

15.ServletConfig

Provides the action with access to various servlet-based information.

16.timer

Provides simple profiling information in the form of how long the action takes to execute.

17.token

Checks the action for a valid token to prevent duplicate formsubmission.

18.validation

Provides validation support for actions

Strtus2 自定义拦截器

源码分析

Interceptor

```
package com.opensymphony.xwork2.interceptor;
import com.opensymphony.xwork2.ActionInvocation;
import java.io.Serializable;
public interface Interceptor extends Serializable {

    void destroy();

    void init();

    String intercept(ActionInvocation invocation) throws Exception;
}
```

AbstractInterceptor

```
package com.opensymphony.xwork2.interceptor;
import com.opensymphony.xwork2.ActionInvocation;
public abstract class AbstractInterceptor implements Interceptor {

    public void init() {}

    public void destroy() {}

    public abstract String intercept(ActionInvocation invocation) throws Exception;
}
```

strtus.xml

```
<?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">
<struts>

    <!-- 常量 -->
    <constant name="struts.enable.DynamicMethodInvocation" value="true" /><!-- 动态方法 -->
    <constant name="struts.devMode" value="true" /><!-- 开发者模式 -->

    <package name="default" namespace="/" extends="struts-default">
        <interceptors>
            <!-- 自定义拦截器 -->
            <interceptor name="myInterceptor" class="online.qsx.interceptor.MyInterceptor" />

            <!-- 定义拦截器栈引用struts2默认的系统拦截器,在引入自定义的拦截器 -->
            <interceptor-stack name="myStack">
                <interceptor-ref name="defaultStack" />
                <interceptor-ref name="myInterceptor" />
            </interceptor-stack>

        </interceptors>

        <!-- 设置所有Action自动调用的拦截器堆栈 -->
        <default-interceptor-ref name="myStack" />

        <!-- http://localhost/Strtus2_HelloWorld/demo!to_index -->
        <action name="demo!*" class="online.qsx.action.DemoAction" method="{0}">
            <!-- 指定该Action触发就会调用的拦截器 -->
            <!-- <interceptor-ref name="myStack" /> -->
            <result name="success">/WEB-INF/jsp/index.jsp</result>
        </action>
    </package>
</struts>
```

DemoAction.java

```
package online.qsx.action;

public class DemoAction {

    public String to_index() {
```

```

        System.out.println("DemoAction.....");
        return "success";
    }
}

```

MyInterceptor.java

```

package online.qsx.interceptor;

import com.opensymphony.xwork2.ActionInvocation;
import com.opensymphony.xwork2.interceptor.AbstractInterceptor;
import com.opensymphony.xwork2.interceptor.Interceptor;

public class MyInterceptor extends AbstractInterceptor{

    @Override
    public String intercept(ActionInvocation invocation) throws Exception {
        System.out.println("=====MyInterceptor intercept begin");
        // 进行后续操作
        String result = invocation.invoke();
        System.out.println("=====MyInterceptor intercept end");
        return result;
    }
}

```

Strtus2 Ajax 支持

请浏览 [Strtus2 多种 result 处理,json](#)

Strtus2 文件上传

单文件上传

web.xml

```

<?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>Strtus2</display-name>

<welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.htm</welcome-file>
    <welcome-file>default.jsp</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>*.action</url-pattern>
</filter-mapping>
</web-app>

```

strtus2.xml

```

<?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">
<struts>
    <!-- 常量 -->
    <constant name="struts.enable.DynamicMethodInvocation" value="true" /><!-- 动态方法 -->
    <constant name="struts.devMode" value="true" /><!-- 开发者模式 -->
    <constant name="struts.multipart.maxSize" value="2097152" /><!-- 文件上传得大小 -->

    <package name="default" namespace="/" extends="struts-default">
        <action name="upload!" class="online.qsx.action.UploadAction" method="{0}">
            <result name="success">/index.jsp</result>
        </action>
    </package>
</struts>

```

UploadAction.java

```
package online.qsx.action;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.UUID;
import org.apache.struts2.ServletActionContext;
import com.opensymphony.xwork2.ActionSupport;

public class UploadAction extends ActionSupport {

    // setX
    // setXFileName
    // setXContentType

    // 注意，file并不是指前端jsp上传过来的文件本身，而是文件上传过来存放在临时文件夹下面的文件
    private File file;

    // 提交过来的file的名字
    private String fileFileName;

    // 提交过来的file的MIME类型
    private String fileContentType;

    public void setFile(File file) {
        this.file = file;
    }

    public void setFileFileName(String fileFileName) {
        this.fileFileName = fileFileName;
    }

    public void setFileContentType(String fileContentType) {
        this.fileContentType = fileContentType;
    }

    public String uploadFile() throws Exception {
        String folder = ServletActionContext.getServletContext().getRealPath("/upload");
        String newFileName = randomFileName(fileFileName);
        copyFile(file, folder, newFileName);
        return "success";
    }

    /**
     * 复制文件到指定得目录
     */
}
```

```

*
* @param file
*         要复制得文件
* @param folder
*         存放文件得目录
* @param fileFileName
*         文件保存得名称
* @throws Exception
*/
public void copyFile(File file, String folder, String fileFileName) throws Exception {
    if (!new File(folder).exists()) {
        new File(folder).mkdirs();
    }
    InputStream is = new FileInputStream(file);
    OutputStream os = new FileOutputStream(folder + File.separator + fileFileName);
    byte[] buffer = new byte[500];
    int length = 0;
    while (-1 != (length = is.read(buffer, 0, buffer.length))) {
        os.write(buffer, 0, length);
    }
    os.close();
    is.close();
}

/**
 * 随机生成一个文件名称,文件类型一致(保留原文件后缀)
 *
 * @param oldFileName
 *         原文件名
 * @return 新文件名
 */
public String randomFileName(String oldFileName) {
    // 获取原文件得后缀
    String postfix = oldFileName.substring(oldFileName.lastIndexOf("."), oldFileName.length());
    // 随机一个字符串拼接原文件后缀
    return UUID.randomUUID().toString().concat(postfix);
}
}

```

index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
```

```
<!DOCTYPE html >
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    <form action="upload!uploadFile.action" method="post" enctype="multipart/form-data">
        <input type="file" name="file"/>
        <input type="submit" value="提交" />
    </form>
</body>
</html>
```

多文件上传

web.xml

```
<?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>Strtus2</display-name>
    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>index.jsp</welcome-file>
        <welcome-file>default.html</welcome-file>
        <welcome-file>default.htm</welcome-file>
        <welcome-file>default.jsp</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>*.action</url-pattern>
    </filter-mapping>
```

```
</web-app>
```

struts2.xml

```
<?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">
<struts>
    <!-- 常量 -->
    <constant name="struts.enable.DynamicMethodInvocation" value="true" /><!-- 动态方法 -->
    <constant name="struts.devMode" value="true" /><!-- 开发者模式 -->
    <constant name="struts.multipart.maxSize" value="2097152" /><!-- 文件上传得大小 -->

    <package name="default" namespace="/" extends="struts-default">
        <action name="upload!" class="online.qsx.action.UploadAction" method="{0}">
            <result name="success">/index.jsp</result>
        </action>
    </package>
</struts>
```

UploadAction.java

```
package online.qsx.action;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.UUID;
import org.apache.struts2.ServletActionContext;
import com.opensymphony.xwork2.ActionSupport;

public class UploadAction extends ActionSupport {
    // setX
    // setXFileName
    // setXContentType

    // 注意，file并不是指前端jsp上传过来的文件本身，而是文件上传过来存放在临时文件夹下面的文件
    private File[] files;
```



```

// 提交过来的file的名字
private String[] filesFileName;

// 提交过来的file的MIME类型
private String[] filesContentType;

public void setFiles(File[] files) {
    this.files = files;
}

public void setFilesFileName(String[] filesFileName) {
    this.filesFileName = filesFileName;
}

public void setFilesContentType(String[] filesContentType) {
    this.filesContentType = filesContentType;
}

public String uploadFile() throws Exception {
    String folder = ServletActionContext.getServletContext().getRealPath("/upload");
    for (int i = 0; i < files.length; i++) {
        String newFileName = randomFileName(filesFileName[i]);
        copyFile(files[i], folder, newFileName);
    }
    return "success";
}

/**
 * 复制文件到指定得目录
 *
 * @param file
 *          要复制得文件
 * @param folder
 *          存放文件得目录
 * @param fileFileName
 *          文件保存得名称
 * @throws Exception
 */
public void copyFile(File file, String folder, String fileFileName) throws Exception {
    if (!new File(folder).exists()) {
        new File(folder).mkdirs();
    }
    InputStream is = new FileInputStream(file);
    OutputStream os = new FileOutputStream(folder + File.separator + fileFileName);
    byte[] buffer = new byte[500];
    int length = 0;
    while (-1 != (length = is.read(buffer, 0, buffer.length))) {

```

```

        os.write(buffer, 0, length);
    }
    os.close();
    is.close();
}

/**
 * 随机生成一个文件名称,文件类型一致(保留原文件后缀)
 *
 * @param oldFileName
 *         原文件名
 * @return 新文件名
 */
public String randomFileName(String oldFileName) {
    // 获取原文件得后缀
    String postfix = oldFileName.substring(oldFileName.lastIndexOf("."), oldFileName.length());
    // 随机一个字符串拼接原文件后缀
    return UUID.randomUUID().toString().concat(postfix);
}
}

```

index.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html >
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    <form action="upload!uploadFile.action" method="post" enctype="multipart/form-data">
        <input type="file" name="files"/>
        <input type="file" name="files"/>
        <input type="file" name="files"/>
        <input type="submit" value="提交" />
    </form>
</body>
</html>

```

异步文件上传

web.xml

```
<?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>Strtus2</display-name>
    <welcome-file-list>
        <welcome-file>index.html</welcome-file>
        <welcome-file>index.htm</welcome-file>
        <welcome-file>index.jsp</welcome-file>
        <welcome-file>default.html</welcome-file>
        <welcome-file>default.htm</welcome-file>
        <welcome-file>default.jsp</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>*.action</url-pattern>
    </filter-mapping>
</web-app>
```

strtus2.xml

```
<?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">
<struts>
    <!-- 常量 -->
    <constant name="struts.enable.DynamicMethodInvocation" value="true" /><!-- 动态方法 -->
    <constant name="struts.devMode" value="true" /><!-- 开发者模式 -->
    <constant name="struts.multipart.maxSize" value="2097152" /><!-- 文件上传得大小 -->
```

```

<package name="default" namespace="/" extends="json-default">
    <action name="upload!" class="online.qsx.action.UploadAction" method="{0}">
        <result name="success" type="json">
            <!-- 指定该Action那些有get方法的属性,自动转成JSON -->
            <param name="includeProperties">
                newFileName
            </param>
        </result>
    </action>
</package>
</struts>

```

UploadAction

```

package online.qsx.action;

import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.InputStream;
import java.io.OutputStream;
import java.util.UUID;
import org.apache.struts2.ServletActionContext;
import com.opensymphony.xwork2.ActionSupport;

public class UploadAction extends ActionSupport {
    // setX
    // setXFileName
    // setXContentType

    // 注意, file并不是指前端jsp上传过来的文件本身, 而是文件上传过来存放在临时文件夹下面的文件
    private File file;
    // 提交过来的file的名字
    private String fileFileName;
    // 提交过来的file的MIME类型
    private String fileContentType;

    public void setFile(File file) {
        this.file = file;
    }

    public void setFileFileName(String fileFileName) {
        this.fileFileName = fileFileName;
    }
}

```

```

public void setFileContentType(String fileContentType) {
    this.fileContentType = fileContentType;
}

// JSON返回
String newFileName;

public String getNewFileName() {
    return newFileName;
}

public String uploadFile() throws Exception {
    String folder = ServletActionContext.getServletContext().getRealPath("/upload");
    newFileName = randomFileName(fileFileName);
    copyFile(file, folder, newFileName);
    return "success";
}

/**
 * 复制文件到指定得目录
 *
 * @param file
 *      要复制得文件
 * @param folder
 *      存放文件得目录
 * @param fileFileName
 *      文件保存得名称
 * @throws Exception
 */
public void copyFile(File file, String folder, String fileFileName) throws Exception {
    if (!new File(folder).exists()) {
        new File(folder).mkdirs();
    }
    InputStream is = new FileInputStream(file);
    OutputStream os = new FileOutputStream(folder + File.separator + fileFileName);
    byte[] buffer = new byte[500];
    int length = 0;
    while (-1 != (length = is.read(buffer, 0, buffer.length))) {
        os.write(buffer, 0, length);
    }
    os.close();
    is.close();
}

```

```

/**
 * 随机生成一个文件名称,文件类型一致(保留原文件后缀)
 *
 * @param oldFileName
 *         原文件名
 * @return 新文件名
 */
public String randomFileName(String oldFileName) {
    // 获取原文件得后缀
    String postfix = oldFileName.substring(oldFileName.lastIndexOf("."), oldFileName.length());
    // 随机一个字符串拼接原文件后缀
    return UUID.randomUUID().toString().concat(postfix);
}
}

```

index.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html >
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<script src="https://cdn.bootcss.com/jquery/3.2.1/jquery.min.js"></script>
<script type="text/javascript">
    function upload() {
        var formData = new FormData();//js 的 from
        formData.append("file", $("input[type='file']")[0].files[0]);
        var opt = {
            url : "upload!uploadFile.action",
            type : "post",
            data : formData,//文件
            dataType : "json",
            processData : false,
            contentType : false,
            success : function(json) {
                var img = "<img src='upload/'+json.newFileName+' ' width='100' height='100' />";
                $("#pic").html(img);
            }
        }
        $.ajax(opt);
    }
}

```

```
</script>
</head>
<body>
    <div id="pic" style="width: 50px; height: 50px;"></div>
    <input type="file" name="file" onchange="upload()"/>
</body>
</html>
```

Strtus2 文件下载

请浏览 [Strtus2 多种 result 处理,stream](#)

Strtus2 异常处理

全局异常处理

strtus.xml

```
<?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">
<struts>
    <!-- 常量 -->
    <constant name="struts.enable.DynamicMethodInvocation" value="true" /><!-- 动态方法 -->
    <constant name="struts.devMode" value="true" /><!-- 开发者模式 -->

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

        <!-- 全局的返回界面 -->
        <global-results>
            <result name="error1">/WEB-INF/jsp/error1.jsp</result>
            <result name="error2">/WEB-INF/jsp/error2.jsp</result>
        </global-results>

        <!-- 全局异常 -->
        <global-exception-mappings>
            <exception-mapping result="error1" exception="java.Lang.NullPointerException" />
            <exception-mapping result="error2" exception="java.Lang.ArithmeticException" />
        </global-exception-mappings>
```

```

        <!-- http://localhost/Strtus2_HelloWorld/demo_1!to_index -->
        <action name="demo_1!*" class="online.qsx.action.DemoAction" method="{0}">
            <result name="success">/WEB-INF/jsp/index.jsp</result>
        </action>
    </package>
</struts>

```

DemoAction.java

```

package online.qsx.action;

public class DemoAction {
    public String to_index() {
        System.out.println("DemoAction.....");

        // 模拟异常
        Object obj = null;
        obj.toString();

        // int i = 0 / 0;
        return "success";
    }
}

```

error1.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
<!DOCTYPE html >
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    java.lang.NullPointerException[全局异常]
</body>
</html>

```


error2.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
<!DOCTYPE html >
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>
    java.lang.ArithmeticException[全局异常]
</body>
</html>
```

局部异常处理

strtus.xml

```
<?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">

<struts>
    <!-- 常量 -->
    <constant name="struts.enable.DynamicMethodInvocation" value="true" /><!-- 动态方法 -->
    <constant name="struts.devMode" value="true" /><!-- 开发者模式 -->

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

        <action name="demo_2!*" class="online.qsx.action.DemoAction" method="{0}">
            <exception-mapping result="error3" exception="java.lang.NullPointerException"/>
            <result name="error3">/WEB-INF/jsp/error3.jsp</result>
            <result name="success">/WEB-INF/jsp/index.jsp</result>
        </action>
    </package>
</struts>
```

DemoAction.java

```
package online.qsx.action;

public class DemoAction {

    public String to_index() {

        System.out.println("DemoAction.....");

        // 模拟异常
        Object obj = null;

        obj.toString();

        // int i = 0 / 0;

        return "success";

    }

}
```

error3.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<%@taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
<!DOCTYPE html >
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
</head>
<body>

    java.lang.NullPointerException[局部异常]

</body>
</html>
```