

先写一点关于 Struts1 的材料，struts2 了解一下和 Struts1 的区别就可以了，spring 框架看老师发的材料中 spring 的代码应该就可以了。

<!--下面的总结部分来自 csdn 博客，对第 4,5,6,7 点稍加注意即可-->

- 1) 两个框架都是 MVC 的 WEB 框架，
- 2) struts1 是老牌框架，应用很广泛，有很好的群众基础，使用它开发风险很小，成本低，Struts2 核心设计思想主要源自 Webwork，实现更优雅，更容易扩展
- 3) Struts2 修改和完善了 struts1 中的一些缺陷，struts1 中一些悬而未决问题在 struts2 得到了解决。
- 4) struts1 的前端控制器是一个 Servlet，名称为 ActionServlet，struts2 的前端控制器是一个 filter，在 struts2.0 中叫 FilterDispatcher，在 struts2.1 中叫 StrutsPrepareAndExecuteFilter。
- 5) struts1 的 action 需要继承 Action 类，struts2 的 action 可以不继承任何类；struts1 对同一个路径的所有请求共享一个 Action 实例，struts2 对同一个路径的每个请求分别使用一个独立 Action 实例对象，所有对于 struts2 的 Action 不用考虑线程安全问题。
- 6) 在 struts1 中使用 formbean 封装请求参数，在 struts2 中直接使用 action 的属性来封装请求参数。
- 7) struts1 中的多个业务方法放在一个 Action 中时（即继承 DispatchAction 时），要么都校验，要么都不校验；对于 struts2，可以指定只对某个方法进行校验，当一个 Action 继承了 ActionSupport 且在这个类中只编写了 validateXxx() 方法，那么则只对 Xxx() 方法进行校验。
- 8) 与 Struts1 不同，Struts2 对用户的每一次请求都会创建一个 Action，所以 Struts2 中的 Action 是线程安全的。

在开始之前推荐花几分钟读一个博客：[\(4 条消息\) JavaWeb 学习——struts1 框架篇\\_长亭古道的博客-CSDN 博客\\_struts1](#)

Struts1 有两个控制器，一个前端控制器一个后端控制器，前端控制器主要的作用是控制页面显示，浏览器发出的请求对应到哪个类，后端控制器执行完服务后返回到哪个页面。至于后端控制器就是处理前端发送的数据，控制业务流程。

Struts1 项目构成如下，一个 web.xml 文件，用来配置前端控制器（可以配置多个前端控制器，格式都是 xml）

```
<?xml version="1.0" encoding="ISO-8859-1"?>
```

```
<!DOCTYPE web-app
```

```
    PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application 2.2//EN"
```

```
    "http://java.sun.com/j2ee/dtds/web-app_2_2.dtd">
```

```

<web-app>
  <!-- JavaServer Faces Servlet Configuration -->
  <servlet>
    <servlet-name>faces</servlet-name>
    <servlet-class>javax.faces.webapp.FacesServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
  </servlet>
  <!-- Action Servlet Configuration -->
  <servlet>
    <servlet-name>action</servlet-name>
    <servlet-class>org.apache.struts.action.ActionServlet</servlet-class>
    <init-param>
      <param-name>config</param-name>
      <param-value>/WEB-INF/struts-config.xml</param-value>
    </init-param>
  </servlet>

  <!-- JavaServer Faces Servlet Mapping -->
  <servlet-mapping>
    <servlet-name>faces</servlet-name>
    <url-pattern>*.faces</url-pattern>
  </servlet-mapping>
  <!-- Action Servlet Mapping -->
  <servlet-mapping>
    <servlet-name>action</servlet-name>
    <url-pattern>*.do</url-pattern>
  </servlet-mapping>
</web-app>

```

其中 url 请求中以.faces 结尾的请求交给 faces 的前端控制器, 以.do 结尾的交给 action 的前端控制器, 值得注意的是<init-param>要配置 action 的前端控制器的 xml 文件路径, 其中 action 前端控制器为例进行如下分析

```

<action-mappings>
  <!-- Process a user logon -->
  <action    path="/logon"
    actionClass="org.apache.struts.webapp.example.LogonAction"
    formAttribute="logonForm"
    formClass="org.apache.struts.webapp.example.LogonForm"
    inputForm="/logon.jsp">
    <forward name="success"    path="/mainMenu.jsp"/>
  </action>
</action-mappings>

```

当程序运行起来后, 浏览器访问/logon, web.xml 自动在填充.do, 将请求交给 action 的前端控制器, 前端控制器将请求发给 LogonAction 类然后在

org.apache.struts.webapp.example.logonAction 对应的 LogonAction 类继承 Action, 可以写 execute 函数 execute(ActionMapping mapping, ActionForm form, HttpServletRequest request, HttpServletResponse response) throws Exception {} 想要调用输入页面显示在前端就需要写 mapping.getInputForward(), 前端控制器自动把 /logon.jsp 显示在前端。客户端在 /logon.jsp 页面提交的 form 表单将通过 java 反射机制存入 formclass 对应路径的 logonform 中, logonform 中写的是有封装字段的 java 类, 同时还要继承 ActionForm, 相当于 mvc 中的 m。其中 mapping 中的函数访问的是 action.xml 文件对应的名值对, form 对应的是前端提交的数据就是 model, request 和 response 就不用说了。当执行 mapping.findforward("success") 当前 LogonAction 类执行结束, 前端控制器将请求分发给 <forward name="success" path="/mainMenu.jsp"/> 中的 /mainMenu.jsp。使用 form 的时候要用一下强制类型转化转换成对应的类。

先了解一下 struts 是怎样运行的, 之后是一些关于考试重点的 LookupDispatchAction 和 DispatchAction 两个类。

显然写这两个类也需要用一个类来继承他们, 先看 dispatchaction。

先看官方文档 (wm 同学已经哭晕在厕所!!!)

先读官方文档为敬

An abstract **Action** that dispatches to a public method that is named by the request parameter whose name is specified by the parameter property of the corresponding ActionMapping. This Action is useful for developers who prefer to combine many similar actions into a single Action class, in order to simplify their application design.

一种抽象操作, 它分派给由请求参数命名的公共方法, 请求参数的名称由相应 ActionMapping 的 parameter 属性指定。此操作对于喜欢将许多类似操作组合到单个操作类中以简化其应用程序设计的开发人员非常有用。

To configure the use of this action in your struts-config.xml file,  
create an entry like this:

```
<action path="/saveSubscription"
type="org.apache.struts.actions.DispatchAction"
name="subscriptionForm" scope="request" input="/subscription.jsp"
parameter="method"/>
```

同样先配置前端控制器, scope 是作用域, 要写上 parameter="method", 这个是用来决定你的请求是调用哪个函数的。

and call one of the methods with a URL like this:

http://localhost:8080/myapp/saveSubscription.do?method=update

地址栏重点是 method=update

然后我们在类里面写一两个函数 (不可以写 excute 函数, 因为 excute 这家伙厉害, 其他人

干不过他，写了就执行他了，自己写的函数就没用了)

## dispatchMethod

```
protected ActionForward dispatchMethod(ActionMapping mapping,
                                           ActionForm form,
                                           HttpServletRequest request,
                                           HttpServletResponse response,
                                           String name)
    throws Exception
```

### Dispatch to the specified method.

在 DispatchAction 类中写如下两函数

```
Public ActionForward update ( ActionMapping mapping , ActionForm form ,
HttpServletRequest request, HttpServletResponse response) {
    XXXXXX;
}
Public ActionForward add ( ActionMapping mapping , ActionForm form ,
HttpServletRequest request, HttpServletResponse response) {
    XXXXXX;
}
```

http://localhost:8080/myapp/saveSubscription.do?method=update

当 url 中 method=update 就执行第一个函数，=add 就执行第二个，这个 dispatchAction 的作用就是在采用标准的 Action 时每个单独的操作都要写一个 Action 的类，这样会使我们的 Action 类膨胀起来，为了解决这个问题，可以使用 Struts 给我们提供的 DispatchAction。

瞟一眼 lookupDispatchAction

An abstract **Action** that dispatches to the subclass mapped execute method. This is useful in cases where an HTML form has multiple submit buttons with the same name. The button name is specified by the parameter property of the corresponding ActionMapping. To configure the use of this action in your struts-config.xml file, create an entry like this:

分派给子类映射的执行方法的抽象操作。这在 HTML 表单具有多个同名提交按钮的情况下非常有用。按钮名称由相应 ActionMapping 的参数属性指定。在 struts 配置中配置此操作的使用。xml 文件，创建如下项（怪不得 struts 没人用呢，搁这搁这呢，配置起来麻烦的一批还想整点花火）

config.xml file, create an entry like this:

```
<action path="/test"
        type="org.example.MyAction"
        name="MyForm"
        scope="request"
        input="/test.jsp"
        parameter="method"/>
```

which will use the value of the request parameter named "method" to locate the corresponding key in ApplicationResources. For example, you might have the following ApplicationResources.properties:

```
button.add=Add Record
button.delete=Delete Record
```

And your JSP would have the following format for submit buttons:

```
<html:form action="/test">
  <html:submit property="method">
    <bean:message key="button.add"/>
  </html:submit>
  <html:submit property="method">
    <bean:message key="button.delete"/>
  </html:submit>
</html:form>
```

Your subclass must implement both getKeyMethodMap and the methods defined in the map. An example of such implementations are:

```
protected Map getKeyMethodMap() {
    Map map = new HashMap();
    map.put("button.add", "add");
    map.put("button.delete", "delete");
    return map;
}

public ActionForward add(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response)
    throws IOException, ServletException {
    // do add
    return mapping.findForward("success");
}

public ActionForward delete(ActionMapping mapping,
    ActionForm form,
    HttpServletRequest request,
    HttpServletResponse response)
    throws IOException, ServletException {
    // do delete
    return mapping.findForward("success");
}
```

**Notes** - If duplicate values exist for the keys returned by getKeys, only the first one found will be returned. If no corresponding key is found then an exception will be thrown. You can override the method unspecified to provide a custom handler. If the submit was cancelled (a html:cancel button was pressed) the custom handler cancelled will be used instead.

## Field Summary

protected	<a href="#">keyMethodMap</a>
<a href="#">Map</a>	Resource key to method name lookup.



原理一样，自行阅读嗷，我实在顶不住了!!!  
祝考试顺利!