

1.创建Maven Web 项目，并解决默认创建时的一些错误

## 2.添加Struts2依赖

```
<!-- 添加struts2核心包 -->
<dependency>
  <groupId>org.apache.struts</groupId>
  <artifactId>struts2-core</artifactId>
  <version>2.5.16</version>
  <!-- 如果需要整合 Hibernate需要排除 javassist -->
  <!--
  <exclusions>
    <exclusion>
      <groupId>org.javassist</groupId>
      <artifactId>javassist</artifactId>
    </exclusion>
  </exclusions>
  -->
</dependency>
<!-- 添加struts2 注解包 -->
<dependency>
  <groupId>org.apache.struts</groupId>
  <artifactId>struts2-convention-plugin</artifactId>
  <version>2.5.16</version>
</dependency>
```

3.在 src/main/resources目录下，创建struts.xml  
在struts2-core-x.x.x.jar 包中找到对应 DTD约束复制到 struts.xml 中

## 3.配置 web.xml --添加struts过滤器

```
<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>/*\.action</url-pattern>
</filter-mapping>
```

## 4.样例页面模板

```
<%@ 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://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>新增类别界面</title>
</head>
<body>
  <h1>新增类别</h1>
  <s:actionmessage/>
  <s:form action="category!save" method="post" namespace="/" theme="simple">
    <table width="600px">
      <tr>
        <th>类别名称</th>
        <td><s:textfield name="pname"/></td>
        <td><font color="red"><s:fielderror fieldName="pname"/></font></td>
      </tr>
      <tr>
        <th colspan="2">
          <input type="submit" value="保存"/>
        </th>
      </tr>
    </table>
  </s:form>
</body>
</html>
```

```
### set log levels ###
log4j.rootLogger = INFO , C , D , E
```

```
### console ###
log4j.appender.C = org.apache.log4j.ConsoleAppender
log4j.appender.C.Target = System.out
log4j.appender.C.layout = org.apache.log4j.PatternLayout
log4j.appender.C.layout.ConversionPattern = [shop][%p] [%-d{yyyy-MM-dd HH:mm:ss}] %C.%M(%L) | %m%n
```

```
### log file ###
log4j.appender.D = org.apache.log4j.DailyRollingFileAppender
log4j.appender.D.File = ../logs/shop.log
log4j.appender.D.Append = true
log4j.appender.D.Threshold = INFO
log4j.appender.D.layout = org.apache.log4j.PatternLayout
log4j.appender.D.layout.ConversionPattern = [shop][%p] [%-d{yyyy-MM-dd HH:mm:ss}] %C.%M(%L) | %m%n
```

```
### exception ###
log4j.appender.E = org.apache.log4j.DailyRollingFileAppender
log4j.appender.E.File = ../logs/shop_error.log
log4j.appender.E.Append = true
log4j.appender.E.Threshold = ERROR
log4j.appender.E.layout = org.apache.log4j.PatternLayout
log4j.appender.E.layout.ConversionPattern = [shop][%p] [%-d{yyyy-MM-dd HH:mm:ss}] %C.%M(%L) | %m%n
```

对于Struts2.5以上的版本如果需要url+!+ 方法访问Action某个方法的话需要在struts.xml加入如下语句

```
<constant name="struts.enable.DynamicMethodInvocation" value="true"/>
<constant name="struts.devMode" value="true"></constant>
```

## convention-plugin和config-browser-plugin插件介绍

以前用struts2框架开发项目时，每次编写好一个Action，就需要在struts.xml文件中配置Action，而convention-plugin这个插件的出现出现后，就不再需要在struts.xml文件中配置Action了，convention-plugin提供了一种非常方便的注解方式来配置Action类。

convention-plugin采用“约定大于配置”的思想，只要我们遵守约定，完全可以少写配置甚至不写配置；config-browser-plugin插件则用于方便的浏览项目中的所有action及其与jsp view的映射。这二个插件结合起来学习，能很方便的搞定struts2中各种复杂的action-view映射需求，所以现在使用Struts2框架开发Web应用时，一般都会配合这两个插件一起使用。

http://localhost:8080/项目名/config-browser/index.action来访问config-browser-plugin插件提供的视图页面