# 前言

假 如 你 的 人 生 有 理 想，那 么 就 一 定 要 去 追，不 管 你 现 在 的 理 想 在 别 人 看 来是 多 么 的 可 笑 ， 你 也 不 用 在 乎 ， 人 生 蹉 跎 几 十 年 ， 如 果 年 轻 的 时 候 计 较 得 失 前 怕 狼 就 后 怕 虎 而 不 去 追 逐， 等 到 了 40， 50 岁 的 时 候 ， 你 唯 一 能 做 的 就 是 感 叹 岁 月 的 消 逝 和 无 情…

# 实验软件下载

1. 下载struts2.1.6 <http://struts.apache.org/download.cgi#struts217>   
   (选择下载full distribution)
2. 下载myeclipse，<http://www.myeclipseide.com>
3. 下载tomcat，<http://tomcat.apache.org>
4. 下载xwork，<http://www.opensymphony.com/xwork> (要注意的是下载与struts2.1.6的对应版本，即2.1.2)

# 安装与设置

1. 解压struts2.1.6，假设解压后位于d:\struts2目录
2. 安装MyEclipse，任何版本都可以
3. 假设你已经安装好jdk和tomcat了，如d:\tomcat

# HelloWorld

1. 建立项目
   1. 打开eclipse
   2. 新建项目
   3. 选择myeclipse-java enterprise projects-web project
   4. 项目名称：struts2\_0100\_introduction
   5. 选择java ee 5.0
2. 设定server
   1. window – preferences – myeclipse – servers – tomcat – 6.x
   2. 选择tomcat home directory
   3. 选择enable
   4. finish
3. 设定jdk环境
   1. window – preferences – java – installed jres
   2. 如果没有对应的JDK（不是JRE），选择add
   3. 选择standard VM -> next
   4. 选择JDK对应的Directory
   5. 将刚刚设定的JDK设为默认
4. 建立第一个struts2.1的程序
   1. 找到struts目录下对应的apps目录
   2. 解压struts2-blank-2.1.6.war
   3. Copy对应的lib的jar文件，需要除junit和spring-test之外的所有文件，其中commons-io的jar包会在文件上传和下载时需要，其他为必须
   4. Copy对应的struts.xml到src目录，在package explorer视图进行操作
   5. 注释掉struts.xml的多余内容
   6. 建立HelloStruts2\_1.jsp文件
   7. \*修改jsp文件的默认编码属性window-preferences-web-jspfiles-设为Chinese,National Standard
   8. 在struts.xml中照原配置进行对应的配置
   9. 修改对应的web.xml，建立struts2的filter（参考struts自带的项目）
5. 运行项目
   1. 可以首先部署到tomcat上项目
   2. 也可以项目右键debug as – myeclipse server app，选择刚刚建立好的server
6. 其他
   1. 学习建立struts的dev-mode，好处在于我们修改了配置文件的时候能够自动热替换
   2. 建立jar文件对应的源码D:/share/tools/struts-2.1.6/src/core/src/main/java，以及xwork对应的源码，以及对应的javadoc location，（更好的方式是建立自己的user-library）
   3. 浏览struts的目录
   4. 认识eclipse jee的项目部署目录，认识如何修改webapp的context-root，项目copy改名后可能会出现的问题
   5. package explorer – 建立jar的源文件和目标关联，navigator – 观察所有的内容
   6. 认识Action的后缀名，在struts2默认中，带不带action都可以

# 常见问题：

1. struts.xml不自动提示
   1. window – preferences – 搜索 catalog – add
   2. 选择key type为URI
   3. key: <http://struts.apache.org/dtds/struts-2.0.dtd>
   4. location: 对应的dtd文件，位于struts-core包中，解压开，指定相应位置，如：D:\share\0750\_Struts2.1.6\soft\struts-2.1.6\lib\struts2-core-2.1.6\struts-2.0.dtd
2. 导入示例项目
   1. 注意JDK的版本，应该是JDK6，否则会出class version的问题
   2. 如果总是项目出问题，可以重建项目，单独拷文件
   3. 还有一种情况是项目右键，properties – java – compiler 选择6.0
   4. 另外，在tomcat配置中也要使用jdk6
3. devMode设为true就会出问题空指针问题
   1. tomcat路径带了空格
4. 在使用ActionContext.getContext().get(“request”)取不到对象，为null值
   1. web.xml不要使用2.0的filter，用新的
5. 在action的试验中，遇到一个奇怪的bug
   1. 用default-action-ref指定一个action
   2. 用\*\_\*映射
   3. 在地址栏中访问一个不存在的action，如adfsdfsfs
   4. 结果居然会映射到\*\_\*上
   5. 如果换成\*\_\_\*就没问题了
   6. 靠，只能认为是bug
   7. 下次用\*-\*，不用\*\_\*

# 学习方法说明

1. 先阅读老师的代码
2. 再建立自己的项目进行动手学习

# Action

1. 实现一个Action的最常用方式：从ActionSupport继承
2. DMI动态方法调用 ！
3. 通配符配置 \* {1} {2} …
   1. \*\_\*
4. 接收参数的方法（一般用属性或者DomainModel来接收）
5. 简单参数验证addFieldError
   1. 一般不使用Struts2的UI标签
6. 访问Web元素
   1. Map类型
      1. IoC
      2. 依赖Struts2
   2. 原始类型
      1. IoC
      2. 依赖Struts2
7. 包含文件配置
8. 默认action处理

# Result

1. 常用四种类型：
   1. dispatcher（默认）
   2. redirect
   3. chain
   4. redirectAction
2. 全局结果集
   1. global-results | extends
3. 动态结果（了解）
   1. 在action中保存一个属性，存储具体的结果location
4. 传递参数
   1. 客户端跳转才需要传递
   2. ${}表达式（不是EL）

# OGNL

1. Object Graph Navigation Language
2. 想初始化domain model，可以自己new，也可以传参数值，但这时候需要保持参数为空的构造方法
3. 其他参考ognl.jsp
4. **什么时候在stack中会有两个Action？chain**

# Struts-Tags

1. 通用标签：
   1. property
   2. set
      1. 默认为action scope，会将值放入request和ActionContext中
      2. page、request、session、application
   3. bean
   4. include(对中文文件支持有问题，不建议使用，如需包含，改用jsp包含)
   5. param
   6. debug
2. 控制标签
   1. if elseif else
   2. iterator
      1. collections map enumeration iterator array
   3. subset
3. UI标签
   1. theme
      1. simple xhtml(默认) css\_xhtml ajax
4. AJAX标签
   1. 补充
5. $ # %的区别
   1. $用于i18n和struts配置文件
   2. #取得ActionContext的值
   3. %将原本的文本属性解析为ognl，对于本来就是ognl的属性不起作用
      1. 参考<s:property 和 <s:include

# 定义自己的theme

1. css（覆盖struts2原来的css）
2. 覆盖单个文件
3. 定义自己的theme
4. 实战：
   1. 把所有主题定义为simple
   2. fielderror特殊处理
   3. 自己控制其他标签的展现

# 作业

1. 读doc文档：struts-tags
2. 设计约定（编码规定）
   1. 原则：简单就是美
   2. 库名：项目名
   3. 表的命名：\_Model名
   4. 字段：保持和属性名一致（尽量不要起名和数据库命名冲突）
   5. 用层来划分包com.bjsxt.bbs.action model(bean) service dto(vo)
   6. Action XXXXAction
   7. \*-\*
   8. /
   9. /admin
   10. package “action” adminAction

# 项目开发顺序-以BBS2009的名义

1. 建立界面原型
2. 建立Struts.xml
   1. 确定namespace
   2. 确定package
   3. 确定Action的名称，空的方法
   4. 确定Result
   5. 将界面原型页面进行修改，匹配现有设置
   6. 测试
   7. 做好规划！！！！！
3. 建立数据库（或者实体类）
4. 建立Model层
5. 建立Service层（后面讲了Hibernate后再完善）
   1. 此时可以使用JUnit进行单元测试了
6. 着手开发

# 声明式异常处理

1. 在Actoin中进行异常映射
2. 在package中进行全局异常映射
3. 使用继承共用异常映射
4. Struts2中异常处理由拦截器实现（观察struts-default.xml）
   1. 实际上Struts2的大多数功能都由拦截器实现

# I18N

1. I18N原理
   1. ResourceBundle和Locale的概念
   2. 资源文件
   3. native2ascii
2. Struts的资源文件
   1. Action – Package – App级
   2. 一般只用APP
      1. struts.xml custom.i18n
   3. PropertiesEditor插件
      1. 解压
      2. features plugin 覆盖到myeclipse中的eclipse目录里
   4. 动态语言切换
      1. request\_locale=en\_US

# Struts拦截器以及源码解析

1. Struts架构图
   1. 见文档
2. Struts执行过程分析
3. Interceptor拦截器过程模拟
4. 定义自己的拦截器
   1. acegi – spring security
5. 使用token拦截器控制重复提交（很少用）
6. 类型转换
   1. 默认转换
      1. 日期处理
   2. 写自己的转换器：

**public** **class** MyPointConverter **extends** DefaultTypeConverter{

@Override

**public** Object convertValue(Object value, Class toType) {

**if**(toType == Point.**class**) {

Point p = **new** Point();

String[] strs = (String[])value;

String[] xy = strs[0].split(",");

p.x = Integer.*parseInt*(xy[0]);

p.y = Integer.*parseInt*(xy[1]);

**return** p;

}

**if**(toType == String.**class**) {

**return** value.toString();

}

**return** **super**.convertValue(value, toType);

}

}

**public** **class** MyPointConverter **extends** StrutsTypeConverter{

@Override

**public** Object convertFromString(Map context, String[] values, Class toClass) {

Point p = **new** Point();

String[] strs = (String[])values;

String[] xy = strs[0].split(",");

p.x = Integer.*parseInt*(xy[0]);

p.y = Integer.*parseInt*(xy[1]);

**return** p;

}

@Override

**public** String convertToString(Map context, Object o) {

// **TODO** Auto-generated method stub

**return** o.toString();

}

}

* 1. 三种注册方式：
     1. 局部：XXXAction-conversion.properties
        1. p（属性名称） = converter
     2. 全局：xwork-conversion.properties
        1. com.xxx.XXX（类名）= converter
     3. Annotation
  2. 如果遇到非常麻烦的映射转换
     1. request.setAttribute();
     2. session

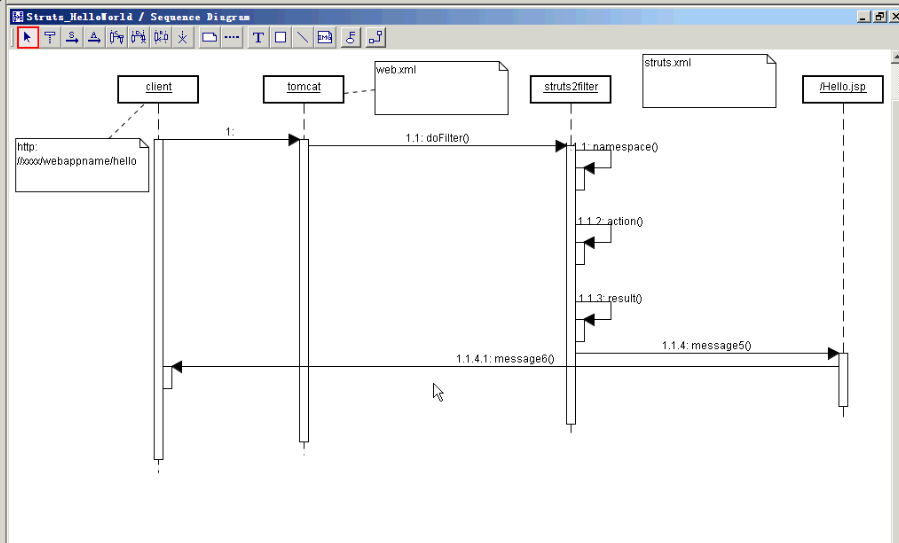
# 补充视频中讲解的内容

1. Lambda表达式
2. 验证框架
3. UI标签
4. 类型转换中的异常处理
5. 上传与下载文件
6. *Struts2注解*

# Struts2总结

1. Action
   1. namespace（掌握）
   2. path（掌握）
   3. DMI（掌握）
   4. wildcard（掌握）
   5. 接收参数（掌握前两种）
   6. 访问request等（掌握Map IOC方式）
   7. 简单数据验证（掌握addFieldError和<s:fieldError）
2. Result
   1. 结果类型（掌握四种，重点两种）
   2. 全局结果（掌握）
   3. 动态结果（了解）
3. OGNL表达式（精通）
   1. # % $
4. Struts标签
   1. 掌握常用的
5. 声明式异常处理（了解）
6. I18N（了解）
7. CRUD的过程（最重要是设计与规划）（精通）
8. Interceptor的原理（掌握）
9. 类型转换（掌握默认，了解自定义）

# 初始流程



# Action执行流程

