课程回顾：

1. 泛型

2. 反射

3. 注解

4. 注解优化BaseDao

目标：

1. 写一个mystruts框架

2. Strtus框架

# 1. 写一个mystruts框架

MVC模式：

Model 模型

View 视图

Control 控制器

Control， 控制器

Servlet起到控制器作用！

----》 获取请求数据封装 【BeanUtils可以优化，(调用方法？)】

----》 调用Service处理业务逻辑

----》 跳转(转发/重定向) 【跳转代码写死】

传统mvc开发总结：

1. 跳转代码写死，不灵活

2. 每次都去写servlet，web.xml中配置servlet！

(配置目的： 请求, Servlet处理类)

需求：

登陆、注册

登陆成功 --🡪 首页

注册成功

--🡪 登陆

--🡪 首页

# 2. 基于mvc模式的应用矿建之struts

Struts就是基于mvc模式的框架！

(struts其实也是servlet封装，提高开发效率！)

Struts开发步骤：

1. web项目，引入struts - jar包

2. web.xml中，引入struts的核心功能

配置过滤器

3. 开发action

4. 配置action

src/struts.xml

|  |
| --- |
| 1. 引入8个jar文件 |
| 2. web.xml |
| <!-- 引入struts核心过滤器 -->  <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> |
| 3. 开发Action |
| // 开发action： 处理请求  **public** **class** HelloAction **extends** ActionSupport {    // 处理请求  **public** String execute() **throws** Exception {  System.*out*.println("访问到了action，正在处理请求");  System.*out*.println("调用service");  **return** "success";  }  } |
|  |
| 4. 配置action : src/struts.xml |
| <?xml version=*"1.0"* encoding=*"UTF-8"* ?>  <!DOCTYPE struts PUBLIC  "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"  "http://struts.apache.org/dtds/struts-2.0.dtd">  <struts>  <package name=*"xxxx"* extends=*"struts-default"*>  <action name=*"hello"* class=*"cn.itcast.action.HelloAction"* method=*"execute"*>  <result name=*"success"*>/success.jsp</result>  </action>  </package>  </struts> |