Day10_jsp & 案例知识点

- 三层体系结构
  - 包结构
    - 公司域名的倒写.项目名称[.子模块名称]...
    - 例如：用户管理系统 user manager system
      - com.itheima.ums
      - （包名称一般都是小写）
  - 结构分类
    - com.itheima.ums.dao — dao接口
    - com.itheima.ums.dao.impl — dao实现类
    - com.itheima.ums.service — service接口
    - com.itheima.ums.service.impl — service实现类
    - com.itheima.ums.web.servlet — servlet
    - com.itheima.ums.web.filter — 过滤器
    - com.itheima.ums.domain — javabean （domain、bean）
    - com.itheima.ums.utils — 工具包
    - com.itheima.ums.exception — 自定义异常
    - com.itheima.ums.constant — 常量
- JSP模式
  - 模式1 — JavaBean + JSP
    - 开发特点适合小项目
    - 开发速度快
    - 但维护成本高
    - 可扩展性差
  - 模式2 — JavaBean + JSP + Servlet
    - 适合平时开发
    - Servlet
      - 逻辑处理
      - 用户数据搜集
      - 选择相应jsp
    - JavaBean — 数据封装
    - JSP — 数据展示
- JSP指令
  - 介绍
    - tomcat将jsp转成servlet，将依据指令所设置内容。
      - 用于指示JSP执行某些步骤
      - 用于指示JSP表现特定行为
      - 用来设定JSP网页的整体配置信息
    - 格式：<%@指令名称 属性=值 属性2=值2 ....%>
    - 分类：page、include、taglib
  - page指令
    - 格式：<%@page 属性=值 ..... %>
  - include指令
    - 格式
      - <%@include file="需要的页面"%>
    - 静态包含（include指令）
      - tomcat在处理jsp时，将1.jsp页面和包含2.jsp页面，
      - 整合在一起生产一个servlet源码，并编译，最后运行。（合并后结果）
    - 动态包含（jsp 动作标签<jsp:include>）
      - tomcat分析处理1.jsp 和 2.jsp，各自进行生成源码和编译成字节码，
      - 在运行时，将1和2 结果合并输出给浏览器。（合并后结果）
  - taglib指令
    - taglib 在jsp页面引用标签库，标签作用是取代jsp脚本的。<%%> <%= %>
    - 格式：<%@taglib uri="标签库的名称" prefix="前缀（别名）"%>
      - uri — 标签库名称，一般都是规定好的。
      - prefix
        - 前缀，表示如何使用标签库
        - 例如：c --> <c:标签名称>
    - 标准标签库：javaee 规定的，有apache实现。
- JSP内置对象
  - JSP提供9个内置对象。其实就是service方法中提供9个变量。默认情况只能使用8个
  - jsp就是servlet，servlet（init/service）是类，类需要被new，就有this
    - 1.page 表示当前页面（this），一般没用。在开发中使用page作为一个标记，jsp中存在一个page作用域。
    - 2.config 类型ServletConfig，就是init参数。表示当前servlet配置。
    - 3.application 类型 ServletContext，servletConfig.getServletContext() 表示一个web项目（应用app） — ServletContext servlet上下文对象（管理者），管理servlet
    - 4.request 类型HttpServletRequest 表示一次请求
    - 5.response 类型HttpServletResponse 表示一次响应
    - 6.session 类型HttpSession 表示一次会话 就是 request.getSession()
    - 7.out 类型 JspWriter 表示发送流数据，等效response.getWriter()
    - 8.exception 类型Throwable ，表示servlet出现异常
    - 9.pageContext 类型 PageContext ，表示page（页面）上下文对象（管理者），jsp管理者
- JSP作用域
  - servlet 有3个作用域
    - request
    - session
    - servletContext
  - jsp 有4个作用域：page、request、session、application
    - page — 表示当前一个页面。在页面上面定义，在页面下面使用。应用常见：标签
    - request — 表示一次请求，默认一个页面，如果请求转发多个页面
    - session — 表示一次会话，可以是多个请求
    - application — 表示一个应用，可以有多个会话
- JavaBean
  - 介绍
    - POJO（Plain Ordinary Java Object） — 简单的Java对象（普通的类）
    - JavaBean
      - 特殊的POJO，具有一定规范，一般称为组件开发。常用数据封装
      - （面向对象：封装）
  - BeanUtils工具包 导入jar包
    - apache 提供commons组件（工具包），beanUtils用于处理javabean。
  - api
    - BeanUtils.setProperty(javabean对象，属性名称，值)
    - ConvertUtils.register 进行自定义类型转换器的注册
    - BeanUtils.populate 给表单封装数据
    - 要求：表单元素name的值，必须与javabean属性名称 保持一致
- jsp动作标签（了解）
  - jsp 动作标签（action tag） — jsp自己提供一组标签，用于取代jsp脚本的。不需要使用它taglib进行引入。
  - 格式
    - <jsp:标签名称>
  - 动态包含
    - <jsp:include>
    - 当前1jsp在运行时，包含另一个2jsp
  - <s:param> 设置参数
- PageContext对象
  - jsp界面管理者（上下文对象）
  - 对其他8个对象进行引用
    - 语法 — pageContext.getXxx()
    - 其它8个对象
      - getPage()
      - getServletConfig()
      - getServletContext()
      - getRequest()
      - getResponse()
      - getSession()
      - getOut()
      - getException()
  - 简化4个作用域操作
    - 默认作用域操作
      - setAttribute(name, value) — 给page 作用域设置内容
      - getAttribute(name) — 从page作用域获得内容
      - removeAttribute(name) — 将所有作用域将指定名称内容移除了
    - 对指定作用域操作
      - setAttribute(name, value, int scope) — 给指定作用域设置内容
      - getAttribute(name, int scope) — 从指定作用域获得内容
      - removeAttribute(name, int scope) — 将指定作用域的内容移除
    - 指定作用域
      - public static final int PAGE_SCOPE 1
      - public static final int REQUEST_SCOPE 2
      - public static final int SESSION_SCOPE 3
      - public static final int APPLICATION_SCOPE 4
    - 快捷方式
      - findAttribute(name) 依次从page、request、session、application 作用域获得指定名称的值，如果没有返回null