

# Spring MVC 笔记

---

三层架构：

- 1、表现层：常说的 web 层。负责接收客户端请求，向客户端响应结果  
表现层包括展示层和控制层。控制层负责接收请求，展示层负责结果的展示。  
表现层依赖业务层，接收到客户端请求一般会调用业务层进行业务处理，并将处理结果响应给客户端。MVC 是表现层的设计模型，和其他层没有关系
- 2、业务层：常说的 service 层。负责业务逻辑处理  
web 层依赖业务层，但是业务层不依赖 web 层。
- 3、持久层：常说的 dao 层。负责数据持久化，包括数据层即数据库和数据访问层。  
持久层就是和数据库交互，对数据库表进行增删改查的。

MVC 设计模式：

Model（模型）：包含业务模型和数据模型，数据模型用于封装数据，业务模型用于处理业务。

View（视图）：通常指的就是我们的 jsp 或者 html。作用一般就是展示数据的。通常视图是依据模型数据创建的。

Controller（控制器）：是应用程序中处理用户交互的部分。作用一般就是处理程序逻辑的。

MVC 提倡：每一层只编写自己的东西，不编写任何其他的代码；分层是为了解耦，解耦是为了维护方便和分工协作。

请求参数绑定：

参数绑定：取出参数值绑定到 handler 方法的形参上

可支持简单数据类型：八种基本数据类型及其包装类型

Restful 基础知识：

基本定义：Restful 是一种 web 软件架构风格，它不是标准也不是协议，它倡导的是一个资源定位及资源操作的风格。

它本身并没有什么实用性，其核心价值在于如何设计出符合 REST 风格的网络接口。

优点：结构清晰、符合标准、易于理解、扩展方便，所以正得到越来越多网站的采用。

GET 用来获取资源，POST 用来新建资源，PUT 用来更新资源，DELETE 用来删除资源。

Ajax Json 交互：

1) 前端到后台：前端 ajax 发送 json 格式字符串，后台直接接收为 pojo 参数，使用注解

@RequestBody

2) 后台到前端：后台直接返回 pojo 对象，前端直接接收为 json 对象或者字符串，使用注解

@ResponseBody

@ResponseBody 注解的作用是将 controller 的方法返回的对象通过适当的转换器转换为指定的格式之

后，写入到 response 对象的 body 区，通常用来返回 JSON 数据或者是 XML 数据。

拦截器基本概念：

Servlet：处理 Request 请求和 Response 响应

过滤器（Filter）：对 Request 请求起到过滤的作用，作用在 Servlet 之前

监听器（Listener）：随 Web 应用的启动而启动，只初始化一次，然后会一直运行监视，随 Web 应用的停止而销毁。1、初始化工作 2、监听 web 中的特定事件

拦截器（Interceptor）：只会拦截访问的控制器方法（Handler）

serlvet、filter、listener 是配置在 web.xml 中的，而interceptor 是配置在表现层框架自己的配置文件中

- 1、在 Handler 业务逻辑执行之前拦截一次
- 2、在 Handler 逻辑执行完毕但未跳转页面之前拦截一次
- 3、在跳转页面之后拦截一次

Mybatis 整合 Spring:

- 1、数据库连接池以及事务管理都交给 Spring 容器来完成
- 2、SqlSessionFactory 对象应该放到 Spring 容器中作为单例对象管理
- 3、Mapper 动态代理对象交给 Spring 管理，我们从 Spring 容器中直接获得 Mapper 的代理对象

Spring Data JPA :

概念：是 Spring 基于 JPA 规范的基础上封装的一套 JPA 应用框架

作用：Spring Data JPA 极大简化了数据访问层代码

构建步骤：

- 1、创建工程导入坐标（Java 框架于我们而言就是一堆 jar）
- 2、配置 Spring 的配置文件（配置指定框架执行的细节）
- 3、编写实体类 Resume，使用 JPA 注解配置映射关系
- 4、编写一个符合 Spring Data JPA 的 Dao 层接口（ResumeDao 接口）