

# 面向服务编程复习

2019年10月26日 13:38

业务服务的设计原则：

- (1) 无状态。
- (2) 单一实例，避免功能的冗余
- (3) 明确定义的接口
- (4) 自包含和模块化
- (5) 粗粒度
- (6) 服务之间的松耦合性
- (7) 重用能力，服务应该是可以重用的
- (8) 互操作性、兼容和策略声明

使用SOA系统的主要角色包括：

设计人员、开发人员、管理人员、操作人员

SOA结构中共有三种角色：

- (1) 服务提供者，可以发布自己的服务，并且对使用自身服务的请求进行相应
- (2) 服务注册中心，也经常被称为服务代理，用于注册已经发布的Service Provider,对其进行分类，并提供搜索服务。
- (3) 服务请求者，利用服务注册中心查找所需要的服务，然后使用该服务。

SOAP的消息格式并介绍每个元素：

SOAP消息是包含3个元素的XML文档信息项，  
<Envelope>、<Header>和<Body>，Envelope是SOAP消息的根元素，  
包含一个可选的Header元素和一个必需的Body元素。Header元素是一种以非集中的方式  
增加SOAP消息功能的通用手法，其每个子元素都被称为一个Header block。

UDDI的数据结构包含的元素：

- 1 商业实体：businessEntity元素
- 2 商业服务信息：businessService元素
- 3 技术绑定信息：bindingTemplate元素
- 4 技术规范信息：tModel元素

ESB相对于EAI的进化特征：消息  
转换、流程编排、服务管理

避免发生循环依赖的方法：依赖倒  
置原则、自动检测、抽离底层、使  
用中间层/基础层

PEST的思想的关键点：

- 1.面向资源的接口设计
- 2.抽象操作为基础的CRUD
- 3.URI，一般的每个资源至少有一个URI与之对应
- 4.无状态

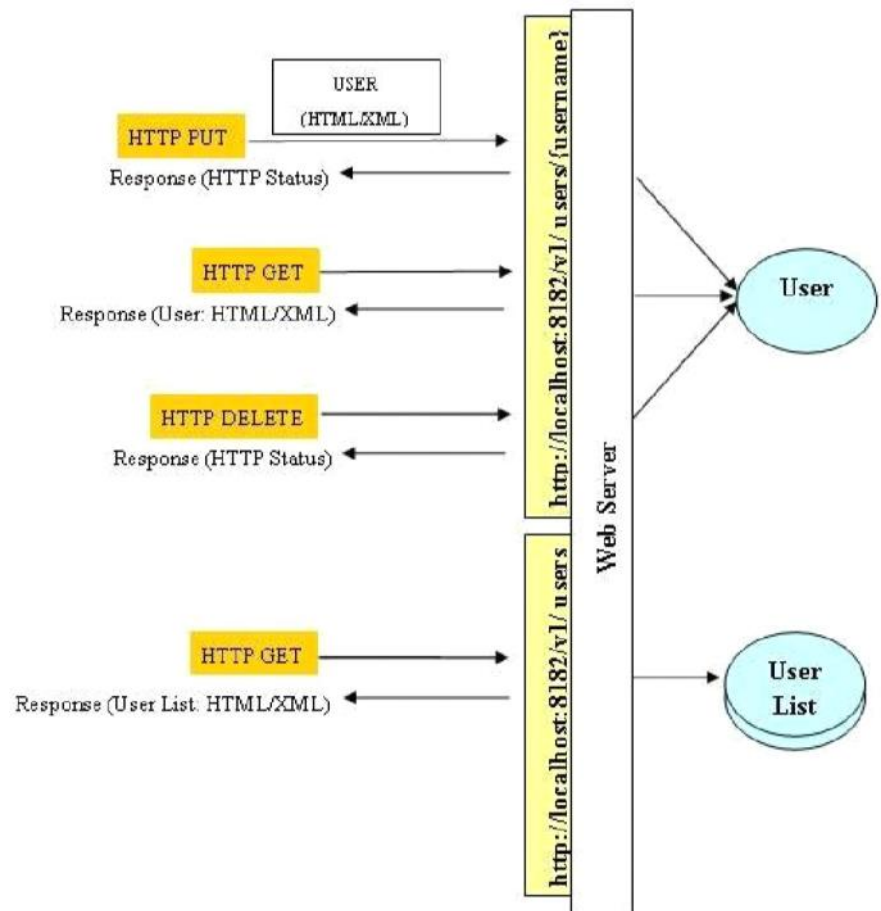
MVC模式的组成:

MVC是模型 (Model) —视图 (View) —控制器 (Controller) 的缩写, 是一种软件设计模式, 目的是将M (业务逻辑) 和V (显示) 分离, 其最大的优点是耦合性低和可维护性高。

Model (模型): 负责业务逻辑的处理以及数据库的交互, Service、Dao、Entity都属于模型层

View (视图): 负责显示界面并与用户交互, 不包含业务逻辑和控制逻辑的Web前端属于视图层。

Controller(控制器): 负责接收请求并控制请求转发, 是连接模型层M和视图层V的桥梁。



Spring MVC的组件:

DispatcherServlet:前端控制器, 请求入口  
HandlerMapping:控制器, 请求派发 (处理器映射)

Controller:控制器, 请求处理流程  
ModelAndView:模型, 封装业务处理结果和视图名称;

ViewResolver:视图, 视图显示处理器

Spring框架的模块组成:

- 1.Spring Core (提供Spring框架的基本功能)
- 2.Spring AOP (面向切面编程)
- 3.Spring ORM (Spring 框架插入了若干个 ORM 框架, 从而提供了 ORM 的对象关系工具)
- 4.Spring DAO (提供对 JDBC 和 DAO 的支持)
- 5.Spring Web (建立在应用程序上下文模块之上, 为基于 Web 的应用程序提供上下文)
6. Spring Context (向 Spring 框架提供上下文信息)
- 7.Spring Web MVC (构建 Web 应用程序的 MVC 实现)