

# vo、po、dto、bo、pojo、entity、mode如何区分

---

**Java Bean:** 一种可重用组件，即“一次编写，任何地方执行，任何地方重用”。  
满足三个条件

- 类必须是具体的和公共的
- 具有无参构造器
- 提供一致性设计模式的公共方法将内部域或暴露成员属性

## 认识

- 1、它们都是javabean的一种扩展。这种扩展就让我们程序员自己便于理解和认识。
- 2、在企业开发过程中，javabean有对内的（dao----service---controller），对外（对页面视图，对远程接口）
- 3、一般防止数据的污染

## pojo、entity、mode、model、bean

说的一回事，和数据表直接挂钩。--- 表的列有多少，pojo,bean,model,entity对应多少属性，

- setter / getter
- toString
- 有参
- 无参
- 序列化

--- jsp/servlet --- ssh -- ssm -- 其实很多时候并不需要下面的vo,dto,bo等这些具体数据封装。为什么？

因为项目不大，业务不复杂不需要搞这么罗里吧嗦的东西。如果小项目怎么方便怎么来。没不要写这么多数据封装。但是一定到项目到了的规模，开始就进行数据的隐藏和安全性以及对内和对外都有严格界限的时候，就一定考虑使用严格数据封装比如：vo dto bo 等等。

## VO

**value object**：值对象

通常用于业务层之间的数据传递，由new创建，由GC回收

和PO一样也是仅仅包含数据而已，但应是抽象出的业务对象，可以和表对应，也可以不是

## PO

**persistant object**：持久层对象

是ORM(Object Relational Mapping)框架中Entity，PO属性和数据库中表的字段形成一一对应关系

VO和PO，都是属性加上属性的get和set方法；表面看没什么不同，但代表的含义是完全不同的

## DTO

**data transfer object**: 数据传输对象

是一种设计模式之间传输数据的软件应用系统，数据传输目标往往是数据访问对象从数据库中检索数据

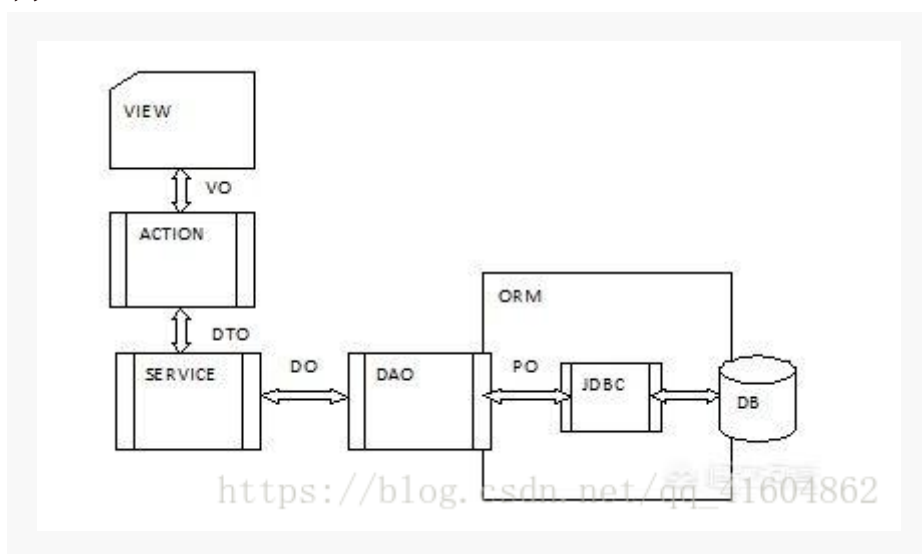
数据传输对象与数据交互对象或数据访问对象之间的差异是一个以不具任何行为除了存储和检索的数据（访问和存取器）

简而言之，就是接口之间传递的数据封装

表里面有十几个字段：id, name, gender (M/F), age.....

页面需要展示三个字段：name, gender(男/女), age

DTO由此产生，一是能提高数据传输的速度(减少了传输字段)，二能隐藏后端表结构



## BO

**business object**: 业务对象

BO把业务逻辑封装为一个对象，通过调用DAO方法，结合PO或VO进行业务操作  
PO组合，如投保人是一个PO，被保险人是一个PO，险种信息是一个PO等等，他们组合起来是第一张保单

```
1 public class OrderDetail{}
2 public class Order{}
3
4 public UserOrderBo(){
5     private OrderDetail orderDetail;
```

```

6     private Order order;
7     private int total;
8 }
9
10 public UserOrderBo selectOrder(Long userId){
11
12     // 1: 订单明细
13     OrderDetail orderDetail = new OrderDetail();
14     // 2: 订单支付信息
15     Order order = new Order();
16     // 3: 钉钉总金额
17     int total = 1111;
18
19     UserOrderBo userOrderBo = new UserOrderBo();
20     userOrderBo.setOrderDetail(orderDetail);
21     userOrderBo.setOrder(order);
22     userOrderBo.setTotal(total);
23
24     return userOrderBo;
25 }

```

## POJO

**plain ordinary java object**: 简单无规则java对象

纯的传统意义的java对象，最基本的Java Bean只有属性加上属性的get和set方法

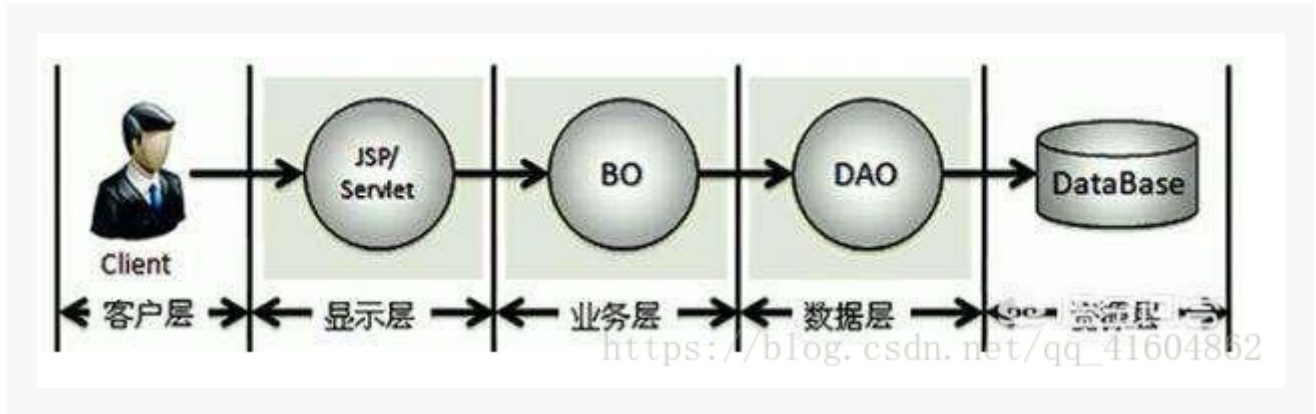
可以额转化为PO、DTO、VO；比如POJO在传输过程中就是DTO

## DAO

**data access object:** 数据访问对象

是sun的一个标准j2ee设计模式，这个模式中有一个接口就是DAO，负责持久层的操作

主要用来封装对数据的访问，注意，是对数据的访问，不是对数据库的访问



其他的还有model/module/domain/entity

## Entity

实体，和PO的功能类似，和数据表一一对应，一个实体一张表