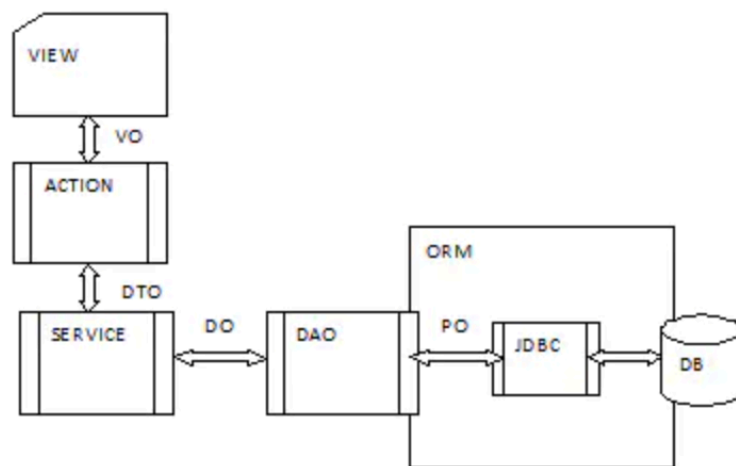


0714项目梳理-VO&DTO&DO

VO对应于页面上需要显示的数据（表单），DO对应于数据库中存储的数据（数据表），DTO对应于除二者之外需要进行传递的数据。

领域模型中的实体类

领域模型中的实体类分为四种类型：VO、DTO、DO、PO。业务中对应的层次关系如下：



VO (View Object)：视图对象，用于展示层，它的作用是把某个指定页面（或组件）的所有数据封装起来。（前端展示的数据）

DTO (Data Transfer Object)：数据传输对象，这个概念来源于J2EE的设计模式，原来的目的是为了EJB的分布式应用提供粗粒度的数据实体，以减少分布式调用的次数，从而提高分布式调用的性能和降低网络负载，但在这里，我泛指用于展示层与服务层之间的数据传输对象。（展示层将VO转换为服务层对应方法要求的DTO）

DO (Domain Object)：领域对象，就是从现实世界中抽象出来的有形或无形的业务实体。（服务层根据DTO构造/重建DO，调用其业务方法完成具体任务）

PO (Persistent Object)：持久化对象，它跟持久层（通常是关系型数据库）的数据结构形成一一对应的映射关系，如果持久层是关系型数据库，那么，数据表中的每个字段（或若干个）就对应PO的一个（或若干个）属性。（服务层把DO转换为持久层对应的PO，调用持久层持久化方法）

区别与应用

VO与DTO

一个DTO可以转换成多个VO，对应于多个页面展示。在只有一个页面格式时可以简化转换为VO的一步。

DTO与DO

DTO是View层和Service层之间的数据传输对象（可以认为是两者之间的协议），而DO是对现实世界各种业务角色的抽象。

设计的DTO需要兼容，所有Service层需要的数据，无论View层有没有传入都需要包括，而不应被返回给View层的数据，无论其是否设置都应忽略。

DO具有一些不应被View层知道的数据DO具体方法，直接将DO传递给View层会导致跨层依赖，造成不必要的耦合。