Atitit 开发模式简化 请求驱动 事务脚本 sql

脚本之间不要互相依赖，，，

后端直接对接db

领域层可以分为三种主要的模式：事务脚本、领域模型、表模块。 名称的 ... 业务的一个请求将触发一系列的业务处理逻辑，而我们在代码中通常

　　业务逻辑层的模式的发展历史

　　历史上，事务脚本是第一个广泛应用的业务逻辑模式。  
　　后来出现了基于表数据的表模块模式，仍然属于过程式模式，但是加入了一些面向对象思维。  
　　在面向对象开发兴起之后，出现了基于对象的业务逻辑模式，最简单的对象模型就像是数据库表的数据模型，这里的对象就是数据库中的记录，并加了一些额外的方法，这种模式通常叫做活动记录模式。  
　随着业务逻辑的复杂性越大，软件的抽象程度越高，这时就应该从领域着眼，创建一个领域驱动的对象模型，这种模式通常叫做领域模型。

# 事务脚本概念

　　业务逻辑层是一系列过程的集合，每个集合都用来处理来自于表现层的一个请求。业务逻辑层被看做是一系列的相关的操作，系统执行的每个步骤都会被分割成更小的步骤，每个步骤都用一个操作表示，叫做事务。在这个上下文中，事务是一个不可分割的逻辑操作，但这个事务与数据库中的事务没有关系，这个模式叫做事务脚本。

## **使用事务脚本模式设计业务逻辑**

虽然我一直积极地倡导使用面向对象的方式，但在某些情况下使用面向对象的设计方法会有一种“杀鸡用牛刀”的感觉，例如在开发简单的业务逻辑时。在这种情况下，更好的方法是编写面向过程的代码，并使用Martin Fowler在《Patterns of EnterpriseApplication Architecture》一书中提到的事务脚本模式。你可以编写一个称为事务脚本的方法来处理来自表示层的每个请求，而不是进行任何面向对象的设计。如图2所示，这种方法的一个重要特征是实现行为的类与存储状态的类是分开的。

　　事务脚本的优缺点

　　事务脚本就是一个简单的过程式模型，简单是事务脚本最值得一提的优势，对于逻辑不多，时间紧迫且依赖于强大的IDE的项目，事务脚本是其理想的选择。简单既是事务脚本的最大优势，同时也成为了它最大的劣势。事务脚本有造成代码重复的潜质，你会很容易的得到一系列完成类似任务的事务，最终应用程序变成了一团混乱的子程序组合。当然这时，重构闪亮登场。重构可以在很大程度上缓解事务脚本天生的劣势，不过重构也有其作用的范围。