

彬纷享礼积分兑换商城平台开发（APP，小程序，公众号）

产品名称	彬纷享礼积分兑换商城平台开发（APP，小程序，公众号）
公司名称	恒探软件网络科技（6年开发公司）
价格	.00/件
规格参数	
公司地址	广州市天河区东英科技园
联系电话	WX：916966649 13729039903

产品详情

业务开发包括哪些工作？

实际上，我们平时做业务系统的设计与开发，无外乎有这样三方面的工作要做：接口设计、数据库设计和业务模型设计（也就是业务逻辑）。

数据库和接口的设计非常重要，一旦设计好并投入使用之后，这两部分都不能轻易改动。改动数据库表结构，需要涉及数据的迁移和适配；改动接口，需要推动接口的使用者作相应的代码修改。这两种情况，即便是微小的改动，执行起来都会非常麻烦。因此，我们在设计接口和数据库的时候，一定要多花点心思和时间，切不可过于随意。相反，业务逻辑代码侧重内部实现，不涉及被外部依赖的接口，也不包含持久化的数据，所以对改动的容忍性更大。

彬纷享礼兑换软件开发，彬纷享礼兑换APP开发，彬纷享礼兑换小程序开发，彬纷享礼兑换公众号开发，彬纷享礼兑换模式开发，彬纷享礼兑换源码搭建，彬纷享礼兑换平台开发。

针对彬纷享礼兑换系统，我们先来看，如何设计数据库。

数据库的设计比较简单。实际上，我们只需要一张记录彬纷享礼兑换流水明细的表就可以了。表中记录彬纷享礼兑换的赚取和消费流水。用户彬纷享礼兑换的各种统计数据，比如总彬纷享礼兑换、总可用彬纷享礼兑换等，都可以通过这张表来计算得到。

接下来，我们再来看，如何设计彬纷享礼兑换系统的接口。

接口设计要符合单一职责原则，粒度越小通用性就越好。但是，接口粒度太小也会带来一些问题。比如

，一个功能的实现要调用多个小接口，一方面如果接口调用走网络（特别是公网），多次远程接口调用会影响性能；另一方面，本该在一个接口中完成的原子操作，现在分拆成多个小接口来完成，就可能会涉及分布式事务的数据一致性问题（一个接口执行成功了，但另一个接口执行失败了）。所以，为了兼顾易用性和性能，我们可以借鉴 facade（外观）设计模式，在职责单一的细粒度接口之上，再封装一层粗粒度的接口给外部使用。

对于彬纷享礼兑换系统来说，我们需要设计如下这样几个接口。

前面我们讲到，从代码实现角度来说，大部分业务系统的开发都可以分为 Controller、Service、Repository 三层。Controller 层负责接口暴露，Repository 层负责数据读写，Service 层负责核心业务逻辑，也就是这里说的业务模型。

除此之外，前面我们还提到两种开发模式，基于贫血模型的传统开发模式和基于充血模型的 DDD 开发模式。前者是一种面向过程的编程风格，后者是一种面向对象的编程风格。不管是 DDD 还是 OOP，开发模式的存在一般都是为了应对复杂系统，应对系统的复杂性。对于我们要开发的彬纷享礼兑换系统来说，因为业务相对比较简单，所以，选择简单的基于贫血模型的传统开发模式就足够了。

从开发的角度来说，我们可以把彬纷享礼兑换系统作为一个独立的项目，来独立开发，也可以跟其他业务代码（比如营销系统）放到同一个项目中进行开发。从运维的角度来说，我们可以将它跟其他业务一块部署，也可以作为一个微服务独立部署。具体选择哪种开发和部署方式，我们可以参考公司当前的技术架构来决定。

实际上，彬纷享礼兑换系统业务比较简单，代码量也不多，我更倾向于将它跟营销系统放到一个项目中开发部署。只要我们做好代码的模块化和解耦，让彬纷享礼兑换相关的业务代码跟其他业务代码之间边界清晰，没有太多耦合，后期如果需要将它拆分成独立的项目来开发部署，那也并不困难。

相信这样一个简单的业务功能的开发，对你来说并没有太大难度。所以，具体的代码实现我就不在专栏中给出了。感兴趣的话，你可以自己实现一下。接下来的内容，才是我们这一节的重点。