

结束语 | 所谓高手，就是跨过坑和大海！

 time.geekbang.org/column/article/172730

欧创新 2019-12-02



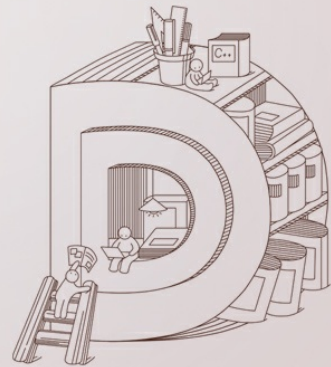
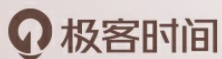
欧创新

人保高级架构师

你好，我是欧创新。

我们一起度过了 **49** 天，共学习了 **21** 篇文章，
阅读了 **85,410** 个字，收听了约 **6** 个小时的音频。

所谓高手，就是跨过坑和大海！



烁

00:00

1.0x [听](#)

讲述：欧创新 大小：5.69M 时长：08:16

你好，我是欧创新。

这是本专栏的最后一讲了，非常感谢你这两个月的陪伴，也非常感谢你的意见和建议。加上前期的专栏筹备，前前后后也有半年了，这半年其实也是自我提升的过程，通过专栏，我将原来不成体系的经验、方法和设计思想，整理成了中台和微服务设计的系统的理论和知识体系。

在撰写专栏时，我站在架构师的角度，尽力将我在实践过程中的经验、思考和体会，以及原创案例等全面详细地呈现给你。希望能够对你的 DDD 实践和架构设计有所帮助，也希望你能快速成长为具有企业级战略视角的架构师和 DDD 设计大师。

那说到成长，相信我们每个人的轨迹都是独特的，但有一点，你一定和我有同样的体会。那就是“所谓高手，就是跨过坑和大海！”每一步都是积累，每一步都是经验，每一步都算数！所以啊，在本专栏的最后，我还是要分享一些干货给你，也是我曾经踩过的一些坑。

很多人接触 DDD，可能是从 DDD 战术设计开始的，因此不知道如何开始 DDD 实践。这个专栏开启后，咱们就可以从领域建模开始了。有了领域模型，我们就可以划分出合理的微服务的逻辑和物理边界；也是因为有了它，我们才能识别出微服务内各关键对象，并建立它们之间的

依赖关系，然后开始微服务的设计和开发。

而很多 DDD 和微服务设计的书籍，大多侧重于讲述 DDD 战术设计或者一些通用的微服务设计模式。这些书籍大多没有告诉我们：如何从业务领域开始，去构建领域模型？如何用 DDD 的思想，来指导中台和微服务设计？如何将领域模型作为输入，来设计和拆分微服务？如何将 DDD 知识体系组合起来，应用到中台和微服务的设计和开发中...

这也是本专栏与这些书籍的不同点。当然，我并不是说它们不好，只是各有侧重。在真正实践的时候，强大的知识基础自然也是刚需，你可以把专栏和书籍结合起来学习，从而发挥最大效能。

下面是我推荐的几本书，这些内容是可以和本专栏互补的，如果你有意愿进一步学习 DDD，它们是非常好的学习资料。

《DDD 实战课》书籍推荐

- 《领域驱动设计：软件核心复杂性应对之道》
- 《实现领域驱动设计》
- 《微服务架构设计模式》

DDD 是一个相对复杂的方法体系，它与传统的软件开发模式或者流程存在一定的差异。在实践 DDD 时，你可能会遇到一些困难。企业需要在研发模式上有一定的调整，同时项目团队也需要提升 DDD 的设计和技术能力，培养适合 DDD 成长的土壤。拔高一点看的话，我觉得你可能会遇到这样三个大坑，下面我来说一说我的看法。

1. 业务专家或领域专家的问题

传统企业中业务人员是需求的主要提出者，但由于部门墙，他们很少会参与到软件设计和开发过程中。如果研发模式不调整，你不要奢望业务人员会主动加入到项目团队中，一起来完成领域建模。没有业务人员的参与，是不是就会觉得没有领域专家，不能领域建模了呢？其实并不是这样的。

对于成熟业务的领域建模，我们可以从团队需求人员或者经验丰富的设计或开发人员中，挑选出能够深刻理解业务内涵和业务管理要求的人员，担任领域专家完成领域建模。对于同时熟悉业务和面向对象设计的项目人员，这种设计经验尤其重要，他们可以利用面向对象的设计经

验，更深刻地理解和识别出领域模型的领域对象和业务行为，有助于推进领域模型的设计。

而对于新的创业企业，他们面对的是从来没人做过的全新的业务和领域，没有任何可借鉴的经验，更不要提什么领域专家。对于这种情况，就需要团队一起经过更多次更细致的事件风暴，才能建立领域模型。当然建模过程离不开产品愿景分析，这个过程是确定和统一系统建设目标以及项目的核心竞争力在哪里。这种初创业务的领域模型往往需要经过多次迭代才能成型，不要奢望一次就可以建立一个完美的领域模型。

2. 团队 DDD 的理念和技术能力问题

完成领域建模和微服务设计后，就要投入开发和测试了。这时你可能会发现一些开发人员，并不理解 DDD 设计方法，不知道什么是聚合、分层以及边界？也不知道服务的依赖以及层与层之间的职责边界是什么？

这样容易出现设计很精妙，而开发很糟糕的状况。遇到这种情况，除了要在项目团队普及 DDD 的知识和设计理念外，你还要让所有的项目成员尽早地参与到领域建模中，事件风暴的过程除了统一团队语言外，还可以让团队成员提前了解领域模型、设计要点和注意事项。

3. DDD 设计原则问题

DDD 基于各种考虑，有很多的设计原则，也用到了很多的设计模式。条条框框多了，很多人可能就会被束缚住，总是担心或犹豫这是不是原汁原味的 DDD。其实我们不必追求极致的 DDD，这样做反而会导致过度设计，增加开发复杂度和项目成本。

DDD 的设计原则或模式，是考虑了很多具体场景或者前提的。有的是为了解耦，如仓储服务、边界以及分层，有的则是为了保证数据一致性，如聚合根管理等。在理解了这些设计原则的根本原因后，有些场景你就可以灵活把握设计方法了，你可以突破一些原则，不必受限于条条框框，大胆选择最合适的方法。

以上就是我对这三个问题的理解了。

用好 DDD 的关键，首先要领悟 DDD 的核心设计思想和理念，了解它为什么适合微服务架构，然后慢慢体会、消化、吸收和实践。DDD 体系虽然复杂，但也是有矩可循的，照着样例多做几个事件风暴，完成领域建模和微服务设计，体会 DDD 的整个设计过程。相信你很快就能领悟到 DDD 的核心设计理念了，这样就可以做到收放自如，趟出一条适合自己的 DDD 实践之路。

好了，到了该说再见的时候了。再次感谢你的陪伴，期待再相遇！愿我们都能跨过坑和大海，开辟出一片广阔新天地！

优惠充值推荐

极客时间充值卡

— 充值享优惠，学习更高效 —



© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。



wallace

Command + Enter 发表

0/2000字

提交留言

精选留言(36)



Miss 置顶

git地址以及示例代码讲解预计元旦前后更新，敬请期待！感谢等待、追更！

2019-12-26

瑞 4

娟 8



阿玛铭

故不积跬步，无以至千里。不积小流，无以成江河。建议老师会说话就出书。这个课程比较全面，包含价值观和方法论两个层面的内容。一是课程订阅者可以作为工具书温习复习，二是私人癖好想收藏记录一下。

作者回复: 谢谢你的建议。等好了告诉你哈。

2019-12-02

猫

姐 6



南山

从专栏出来一篇没有落下的跟到现在，时间真的好快！
收获良多，算是入了ddd的门，重术(战术)更要重道(战略)，后续打算把ddd分享给身边的人，至少一起码的人要有所了解，有相同的语言，才能一起聊下去
感谢老师，江湖再见！！

作者回复：江湖再见！

2019-12-02

瑞 1

娟 5

•



David

讲的很不错，想了解一下幂等和事务方面在ddd实现中有什么思路或经验

作者回复: 就DDD来说，它是没有幂等的方案的，需要我们通过设计来实现。原本想在第20节加一个幂等的议题的。

你可以在不同阶段进行幂等性处理，如使用Token（UUID）、分布式锁、去重表等方式。

可通过Token或全局唯一ID确定请求的唯一性：根据业务生成一个全局唯一ID,在调用接口时会传入该ID,接口提供方会从相应的存储系统比如Redis中去检索这个全局ID是否存在,如果存在则说明该操作已经执行过了,将拒绝本次服务请求;否则将相应服务请求并将全局ID存入存储系统中,之后包含相同业务ID参数的请求将被拒绝。

可使用Redis分布式锁解决资源并发竞争的情况，获取唯一请求；

可使用去重表保证数据库数据唯一：适用于在业务中有唯一标识的插入场景。比如在支付场景中,一个订单只会支付一次,可以建立一张去重表,将订单ID作为唯一索引。把支付并且写入支付单据到去重表放入一个事务中,这样当出现重复支付时,数据库就会抛出唯一约束异常,操作就会回滚。这样保证了订单只会被支付一次。

2019-12-03

瑞

娟 3

•



墨名次

第一次学习这种几乎纯理论的课程确实很考验耐心，幸运的是老师这种讲课方式很适合我，全部学习了，收获很大，感谢！

作者回复: 谢谢你的耐心陪伴。

2019-12-02

瑞

娟 2



quietwater

talk is cheap show me the code

作者回复: 马上就有代码详解上新了。

2020-01-01

滿

娛 1



达文西

粗劣看完了,实在都是干货,值得反复多看几遍.等老师的代码demo出来再对照着看相信收获更大.

2019-12-17

猫

娟 1



Geek_aa8017

老师，git项目地址什么时候可以整理好发出来啊

作者回复: 正在准备中，等整理好了发出来。

2019-12-11

满

娵 1



瓜瓜

老师的代码什么时候放出来啊。

作者回复: 最近比较忙哈，还需要一点时间。等好了告诉你。

2019-12-10

满

娵 1



marker

很希望老师能多出一些领域分析实战的相关课程，四色原型，用例，事件风暴这些相关设计

作者回复：谢谢建议。后续考虑考虑。

2019-12-09

瑞

姐 1



祥敏

欧老师的课程内容富有层次，注重整体。

不能一下子都吸收，后序要反复实践、归纳整理，如此循环才能跨过坑和大海。

作者回复: 谢谢，建议来回多看几遍。

2019-12-04

瑞

娟 1



coke7up

一路追下来，没迷路。谢谢老师。

作者回复: 谢谢

2020-03-31

瑞

娟



stg609

请教老师，关于DDD业务方面的配置如何处理？比如有 保险, 银行 两个领域，及一个配置中心。那和保险紧密相关的业务方面的配置参数，如一些保费费率，是由配置中心统一维护？

这些带有业务意义的配置如果直接有该领域自己维护是否更合适？

作者回复: 配置信息属于弱领域模型，不好建立领域模型，但是他们大多是查询，而且实体之间独立性强，如果考虑复用，建议采用CQRS模式，或者也可将他们放在跟领域模型在一起的微服务内，用一个虚拟的聚合将他们聚在一起。

2020-03-26

瑞

娟

-



狮锅艺

学习打打卡，课程学习结束了，实战才刚刚开始。

2020-03-18

猫

娘



zk_207

新哥，代码demo整理好了吗？GitHub地址贴一下呗

编辑回复：看专栏里1月2日的加餐哈～

2020-03-18

瑞

姐



WING

学习这么的正确姿势，先整门课程通读，前面的理论看不懂先放一放。再通过欧老师后面的案例，回头去理解前面的理论。最后结合自己做过的项目拿来练习。书读百遍其义自见。谢谢欧老师。

作者回复: 是的，DDD主要体现的是设计思想，需要琢磨和慢慢理解。

2020-03-17

满

娒



maomaostyle

非常好的专栏，感谢老师答疑解惑

作者回复: 谢谢😊

2020-03-01

满

娒



小宇

老师PersonDomainService的findByld方法是先从数据库里查一遍，然后getPerson方法中又查询了一遍吗？这个不太理解。这个地方是不是应该用personFactory.createPerson

作者回复：复杂的聚合的数据初始化是可以用工厂的。

2020-02-16

瑞

娟



吴凌华

这个很重要是跨过！

2019-12-24

猫

姐



瓜瓜

另一个问题帮忙回复下？战术设计完成映射，战略设计完成领域建模，这个回答太笼统了，你文章里面写过了，占用您点时间，帮忙看一下另一个问题呢，感谢

作者回复: 已回复。因为不知道你的聚合是如何得来的，不知是否回答清楚？

2019-12-07

猫2

娛

收起评论祇