

测试专栏特别放送 | 答疑解惑第六期

2018-11-09 茹炳晟

软件测试52讲

[进入课程 >](#)



你好，我是茹炳晟。

今天的“答疑解惑”系列文章，我们一起来解决测试数据准备和测试基础架构这两个系列 8 篇文章中的问题。你可以通过下面对每篇文章的简单总结回顾一下文章内容，也可以点击链接回到对应的文章复习。

这两个系列下的文章留言已经很少了，或许是你没有坚持学习，也或许是这部分内容并没有切中你现在的痛点。毕竟，广义上的软件测试，包括了测试数据平台、测试执行平台等，而我们也无法每一种都有机会接触。

但这里，我想再给你打打气，有些知识虽然你在接触时感觉自己不会用到，但随着技术发展、公司业务转型，或者是你个人的职业晋升，都会需要越来越宽广的知识面，这也正应了我在专栏里面提到的一个比喻，**测试工程师通常是“广度遍历”，关注的是“面”**。所以，坚持学习，才是我们从“小工”蜕变为“专家”的正确路径。

在第 35 篇文章 [《如何准备测试数据？》](#) 中，我从测试数据创建的维度，和你详细分享了生成测试数据的四种方法：基于 GUI 操作生成测试数据、通过 API 调用生成测试数据、通过数据库操作生成测试数据，以及综合运用 API 和数据库的方式生成测试数据。

其实，我们要创建的测试数据并不仅仅局限于数据库，很多时候还需要创建消息队列里面的数据。所以，在阅读完这篇文章后，我希望你可以思考一下如何处理这类问题？或者，请你分享一下你曾经是如何解决这个问题的。

这里，我来分享下我的方法吧。

通过模拟消息队列中的测试数据，可以实现各个被测模块之间的解耦，这个思路非常关键。至于如何来模拟消息队列中的测试数据，在技术上其实没有任何难度。

我们通常的做法是，在测试数据工具的底层封装一个工具类，这个工具类内部通过调用消息队列的 API 或者操作接口函数来实现消息队列的 CRUD 操作，然后凡是需要改变消息队列中数据的地方，都通过这个工具类来完成实际操作。

问题二：你所在公司，采用是什么测试数据策略？为什么选用了这种策略？

在专栏的第 36 篇文章 [《浅谈测试数据的痛点》](#) 中，我和你分享了选择不同时机去创建测试数据，是为了解决不同的数据准备痛点。为了解决这些痛点，我的经验是把测试数据分为“死水数据”和“活水数据”，其中：“死水数据”适合用 Out-of-box 的方式，而“活水数据”适合采用 On-the-fly 的方式。

在这篇文章最后，我希望你可以分享一下自己项目中采用的是是什么测试数据准备策略，以及会不会使用线上真实的数据进行测试。

这里我来分享下 eBay 在准备测试数据时的策略吧。

eBay 在准备测试数据时，采用的并不是单一的某种方法，而是针对实际的业务场景选取了不同方法，最后将这些方法进行组合完成整个测试的数据准备工作。可谓是多管齐下。

这里，我和你分享下 eBay 主要使用了的几种策略：

对于“活水数据”，比如订单和优惠券等，一定采用 On-the-fly 的方式实现测试数据用例内的自维护；

对于“死水数据”，比如商品类目和品牌等，一定采用 Out-of-box 的方式来提高测试数据准备的效率；

对于性能测试的背景数据，采用生产环境的实际数据（注意，这里使用的实际生产数据是经过必要的“脱敏”操作的）；

对于复杂数据，采用了预先在系统中预埋 template 数据，然后在需要使用的时候，通过复制 template 数据然后再修改的策略；

对于生产环境的测试，除了全链路压力测试，都会采用真实的数据来进行。

所以，测试数据策略的选择，最重要的是适合自己所在的公司或者项目的实际情况。这里我和你分享的 eBay 的实践，希望可以在针对特定场景选择测试策略的时候，可以感到有据可依。

问题三：如果你所在公司，也处于测试数据 1.0 时代，你们还用到过哪些测试数据准备函数的实现方法吗？

在专栏的第 37 篇文章 [《测试数据的“银弹” - 统一测试数据平台（上）》](#) 和第 38 篇文章 [《测试数据的“银弹” - 统一测试数据平台（下）》](#) 中，我从全球大型电商企业早期的测试数据准备实践谈起，和你一起分析这些测试数据准备方法在落地时遇到的问题，以及如何在实践中解决这些问题。

我希望通过这种遇到问题解决问题的思路，可以带着你去体会时代的演进，理解测试数据准备技术与架构的发展历程，并进一步掌握 3.0 时代出现的业内处于领先地位的“统一测试数据平台”的设计思路。

正如我在文中所说，目前大多数企业都还处于测试数据 1.0 时代。其实，这也很正常。因为，1.0 时代的测试数据准备函数，才是真正实现数据创建业务逻辑的部分，后续测试数据准备时代的发展都是在这个基础上，在方便用户使用的角度上进行的优化设计。所以说，测试数据 1.0 时代是实现后续发展的基础。

API 支持的复杂测试数据场景。

这种方法就是，事先在数据库中插入一条所谓的模板数据，在下一次需要创建这类数据的时候，就直接从模块数据种复制一份出来，然后再根据具体的要求来修改相应的字段。

这里需要特别注意的是，对于 1.0 时代的测试数据准备函数来说，我们还需要建立工具的版本管理机制，使其能够应对多个不同的数据版本。

问题四：目前 Selenium Grid 已经有 Docker 的版本了，你有没有考虑过可以在云端搭建 Selenium Grid 呢？

在专栏的第 39 篇文章 [《从小作坊到工厂：什么是 Selenium Grid？如何搭建 Selenium Grid？》](#) 中，我从测试基础架构的概念讲起，并和你分享了传统 Selenium Grid 和基于 Docker 的 Selenium Grid 的搭建方法。

不知道，你有没有在课后去尝试搭建 Selenium Grid 呢，这其中是否遇到了什么问题？如果你遇到了问题，欢迎你给我留言，我将帮助你一起解决。

在这篇文章最后，我希望你可以畅想一下是否可以在云端搭建 Selenium Grid。这里我结合 eBay 的探索，来谈谈我的看法吧。

对于一些大公司来说，在云端来搭建 Selenium Grid 已经被证明是切实可行的，而且也已经呈现出逐渐向云端过度的趋势。这主要得益于云端部署的易维护性和上云本身的便利性。

比如，eBay 已经实现了在 PCF 上基于 Docker 来运行 Selenium Grid 的方案，其中落地的难点在于配置 Docker 的网络和 IP 地址。主要原因是，PCF 会为部署的应用提供统一 App URL 的命名转换。

从本质上讲，只要 Selenium Grid 是基于 Docker 实现的，那么上不上云本身并没有本质区别。但是，考虑到将来的 App 以及所有的后台服务都会逐渐向云端过度，所以测试基础架构这块必然也会遵循这个趋势，和 App 以及后台服务的环境保持在一个技术栈上，将会减少公司整体基础架构的多样性，从而提高研发效能。

在专栏的第 40 篇文章 [《从小工到专家：聊聊测试执行环境的架构设计（上）》](#) 和第 41 篇文章 [《从小工到专家：聊聊测试执行环境的架构设计（下）》](#) 中，我首先和你分享了测试执行环境的概念，然后为你剖析了测试基础架构的演进历程，希望可以帮助你理解测试基础架构的设计，最终可以定制一套适合自己的测试基础架构。

学习完这两篇文章，我希望你思考的是，你所在团队，还可以再测试执行环境的架构设计上，进行哪些优化和创新？

其实，测试基础架构的优化和创新都是由问题本身驱动的。如果不是为了解决实际遇到的问题，企业是不会为了追求更新的技术而去寻求测试架构的改进方案的。所以，通常情况下，我们都是在测试或者 DevOps 的过程中，遇到了问题或者瓶颈才会考虑新的技术与方法。

比如我在文中提到的，为了解决大量测试用例在短时间内执行完成的要求，才出现了测试执行集群的架构；

再比如，当测试用例数量非常多，每次测试结束，需要分析所有失败的用例时，就必须要考虑基于机器学习的缺陷分类方法；

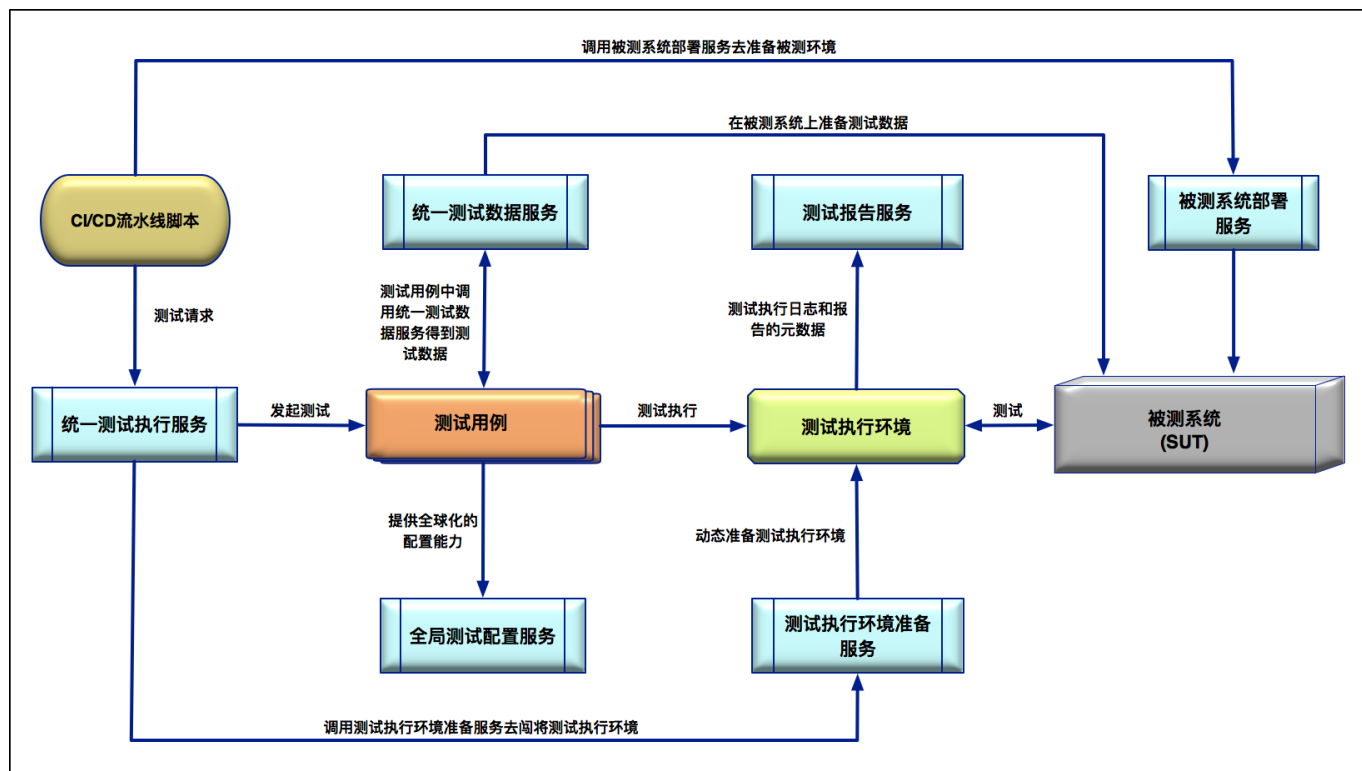
再比如，为了增加测试环境的有效使用时间，避免开发人员要在发生缺陷的环境上实时 Debug 而造成的测试环境占用问题，我们就会考虑在测试用例执行失败的时候自动获取全链路的日志；

再比如，如果我们的大量测试执行请求都是 API 调用，我们就可以实现完全基于 Docker 的并发执行环境。

除了上面这些技术驱动的原因外，还有些是由企业的组织结构驱动的。比如，eBay 的某些产品线，CI/CD 团队在印度，而测试架构团队在中国。大家都知道 CI/CD 的流水线脚本和测试执行是强耦合的，而要求两个异地团队实时合作并保证完全同步很困难。为此，我们就需要考虑为 CI/CD 提供统一的测试接口，这样流水线脚本就可以以解耦的方式与测试集成了。

问题六：你觉得在我给出的全球化电商的测试基础架构中，还可以增加哪些与测试相关的服务？

为了便于你回顾文章内容，我把文中用到的大型全球化电商网站的全局测试基础架构设计图，放到了这里。



其实，除了图中的各类测试服务以外，你完全可以根据你的业务场景来搭建更多的测试服务，以提高团队的测试效率和研发效能。

比如，你可以提供完整和详细被测系统所有版本信息的服务，通过一个简单的 Restful API 调用，就可以得到被测系统中所有组件、各个微服务的详细版本信息，还可以自动比较多个被测环境的版本差异。

再比如，为了解决微服务测试过程中的依赖性，你可以考虑提供基于消费者契约的统一 Mock 服务，以契约文件作为输入，来提供模拟真实服务的 Mock 服务，从而方便微服务的测试。

再比如，你还可以提供压力测试服务，由这个服务统一管理所有的 Load Generator 来发起测试，这个服务还可以进一步细分为前端性能测试服务，以及后端压力测试服务。

这样的例子还有很多，但在实际工作中，我们还是要根据测试本身的上下文以及被测产品的性质，来决定将哪些需求实现成测试服务会。

而且还不止一个团队会需要这种类型测试的时候，你就应该考虑将前端性能测试服务化了。

最后，感谢你能认真阅读第 35~42 这 8 篇文章的内容，并写下了你的想法和问题。期待你能继续关注我的专栏，继续留下你对文章内容的思考，我也在一直关注着你的留言、你的学习情况。

感谢你的支持，我们下一期答疑文章再见！

 极客时间

软件测试52讲

从小工到专家的实战心法



茹炳晟 eBay中国研发中心
测试基础架构技术主管

新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 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。

上一篇 测试专栏特别放送 | 答疑解惑第五期

下一篇 测试专栏特别放送 | 答疑解惑第七期

精选留言 (2)

写留言



Wanderlust

2018-11-10

2

您对测试的理解，以便于类似我这样的读者日后能提高掌握的回响。



Rain 

2019-01-16



内容非常棒。我常常在想，如果所有的课程，都有模拟项目，让学习人员能参与每个细节，代码的具体实现等，那么一轮下来，成长是飞快的。理论的学习，与实际操作，还是有巨大鸿沟，相信在ebay工作的小伙伴们，更能体会到文章的博大精深，因为他们参与其中，有机会与老师沟通交流具体的工作细节。而其它小伙伴，就只能自己探索。

展开 ∨

作者回复: 很高的建议，其实我们也有考虑过，但是一旦细节到代码实现的层面，就会内容过多，过于具体，而且容易失去核心原理

