

35 | 如何准备测试数据？

2018-09-17 茹炳晟

软件测试52讲

[进入课程 >](#)



讲述：茹炳晟

时长 13:15 大小 6.07M



你好，我是茹炳晟，今天我和你分享的主题是：如何准备测试数据。

从今天开始，我们将一起进入测试数据准备这个新的系列了。我会用四篇文章，和你详细探讨软件测试过程中关于测试数据准备的话题。我会依次分享测试数据创建的基本方法、测试数据准备的痛点、自行开发的测试数据工具，以及目前业内最先进的统一测试数据平台。

你我都非常清楚，测试数据的准备是软件测试过程中非常重要的一个环节，无论是手工测试，还是自动化测试，无论是 GUI 测试，还是 API 测试，无论是功能测试，还是性能测试，都避不开测试数据准备的工作。

所以，如果你想要成长为一名优秀的测试工程师，那就非常有必要深入理解测试数据准备的方法，以及它们各自的优缺点、适用场景了。

今天，我们就先从测试数据准备的基本方法开始吧。

从创建测试数据的维度来看，测试数据准备方法主要可以分为四类：

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；

综合运用 API 和数据库的方式生成测试数据。

这时，相信你已经回想起我曾在第 15 篇文章 [《过不了的坎：聊聊 GUI 自动化过程中的测试数据》](#) 中从创建测试数据的维度和你分享过这些内容，这次的分享只不过是多了“通过 GUI 调用生成测试数据”的方法。

其实，我在第 15 篇文章的分享内容，只是简单的介绍了 GUI 测试数据准备的方法，并没有详细展开。事后，你可能也感觉不太过瘾，想知道一些更深入、更细节、更贴近业务场景的测试数据准备的知识。所以，也就有了我今天的这次分享。

那么，接下来我们就赶紧开始吧，一起聊聊这四种测试数据准备的方法。

基于 GUI 操作生成测试数据

基于 GUI 操作生成测试数据，是最原始的创建测试数据的方法。简单地说，它就是采用 E2E 的方式来执行业务场景，然后生成数据的方法。

比如，你想要测试用户登录功能，那么首先就要准备一个已经注册的用户，为此你可以直接通过 GUI 界面来注册一个新用户，然后用这个新创建的用户完成用户登录功能的测试。

这个方法的优点是简单直接，在技术上没有任何复杂性，而且所创建的数据完全来自于真实的业务流程，可以最大程度保证数据的正确性。但是，该方法的缺点也十分明显，主要体现在以下这四个方面：

1. **创建测试数据的效率非常低。** 一是因为每次执行 GUI 业务操作都只能创建一条数据，二是因为基于 GUI 操作的执行过程比较耗时。
2. **基于 GUI 的测试数据创建方法不适合封装成测试数据工具。** 由于测试数据的创建是通过 GUI 操作实现的，所以把这种数据创建方法封装成测试数据准备工具的过程，其实就是

在开发 GUI 自动化测试用例。无论是从开发工作量，还是从执行效率来讲，把基于 GUI 操作的测试数据创建方法封装成测试数据准备工具都不是最佳的选择。

3. **测试数据成功创建的概率不会太高。**因为，测试数据准备的成功率受限于 GUI 自动化执行的稳定性，而且任何界面的变更都有可能引发测试数据创建的失败。
4. **会引入不必要的测试依赖。**比如，你的被测对象是用户登录功能，通过 GUI 页面操作准备这个已经注册的用户，就首先要保证用户注册功能没有问题，而这显然是不合理的。

鉴于以上四方面的原因，在实际的测试过程中，我们很少直接使用基于 GUI 的操作生成测试数据。只有在万不得已的情况下，比如没有其他更好的方式可以创建正确可靠的测试数据时，我们才会使用这个方法。

而且，这里我需要说明的是，基于 GUI 操作生成测试数据的方法一般只用于手工测试，因为自动化测试中使用这种数据准备方法，基本相当于要开发一个完整的 GUI 自动化测试用例，代价太大。

那我为什么还要介绍这个方法呢？其实，这个方法更重要的应用场景是，帮助我们找到创建一个测试数据的过程中，后端调用了哪些 API，以及修改了哪些数据库的业务表，是“通过 API 调用生成测试数据”，以及“通过数据库操作生成测试数据”这两种方法的基础。

通过 API 调用生成测试数据

通过 API 调用生成测试数据，是目前主流的测试数据生成方法。其实，当我们通过操作 GUI 界面生成测试数据时，实际的业务操作往往是由后端的 API 调用完成的。所以，我们完全可以通过直接调用后端 API 生成测试数据。

还是以用户登录功能的测试为例，我们通过 GUI 界面注册新用户时，实际上就是调用了 createUser 这个 API。既然知道了具体要调用哪个 API，那么我们就可以跳过在 GUI 界面的操作，直接调用 createUser 生成“已经注册的用户”这个测试数据了。

为了规避在创建测试数据时过于在乎实现细节的问题，在实际工程实践中，我们往往会把调用 API 生成测试数据的过程封装成测试数据准备函数。那问题是，我怎么才能知道前端新用户注册这个操作到底调用了哪些后端 API 呢？这里，我推荐三种方式：

1. 直接询问开发人员，这是最直接的方法；
2. 如果你有一定的代码基础，可以直接阅读源代码，这个方法也可以作为直接询问方法的补充；

3. 在一个你可以独占的环境上执行 GUI 操作创建测试数据，与此同时监控服务器端的调用日志，分析这个过程到底调用了哪些 API。

通过 API 调用生成测试数据的方法，优点主要体现在以下几个方面：

可以保证创建的测试数据的准确性，原因是使用了和 GUI 操作同样的 API 调用；

测试数据准备的执行效率更高，因为该方法跳过了耗时的 GUI 操作；

把创建测试数据的 API 调用过程，封装成测试数据函数更方便，因为这个调用过程的代码逻辑非常清晰；

测试数据的创建可以完全依赖于 API 调用，当创建测试数据的内部逻辑有变更时，由于此时 API 内部的实现逻辑也会由开发人员同步更新，所以我们依旧可以通过调用 API 来得到逻辑变更后的测试数据，而这个过程对使用来说是完全透明的。

但是，该方法也不是完美无瑕的，其缺点主要表现在：

1. 并不是所有的测试数据创建都有对应的 API 支持。也就是说，并不是所有的数据都可以通过 API 调用的方式创建，有些操作还是必须依赖于数据库的 CRUD 操作。那么，这时，我们就不得不在测试数据准备函数中加入数据库的 CRUD 操作生成测试数据了。
2. 有时，创建一条业务线上的测试数据，往往需要按一定的顺序依次调用多个 API，并且会在多个 API 调用之间传递数据，这也无形中增加了测试数据准备函数的复杂性。
3. 虽然相比于 GUI 操作方式，基于 API 调用的方式在执行速度上已经得到了大幅提升，并且还可以很方便地实现并发执行（比如，使用 JMeter 或者 Locust），但是对于需要批量创建海量数据的场景，还是会力不从心。

因此，业界往往还会通过数据库的 CRUD 操作生成测试数据。

通过数据库操作生成测试数据

通过数据库操作生成测试数据，也是目前主流的测试数据生成方法。这个方法的实现原理很简单，就是直接通过数据库操作，将测试数据插入到被测系统的后台数据库中。

常见的做法是，将创建数据需要用到的 SQL 语句封装成一个个的测试数据准备函数，当我们需要创建数据时，直接调用这些封装好的函数即可。

还是以用户登录功能测试为例，当我们通过 GUI 界面注册新用户时，实际上是在后端调用了 createUser 这个 API，而这个 API 的内部实现逻辑是，将用户的详细信息插入到了 userTable 和 userRoleTable 这两张业务表中。

那么此时，我们就可以直接在 userTable 和 userRoleTable 这两张业务表中插入数据，然后完成这个新用户的注册工作。

这样做的前提是，你需要知道前端用户通过 GUI 操作注册新用户时，到底修改了哪些数据库的业务表。这里，我也推荐三种方式：

1. 直接向开发人员索要使用到的 SQL 语句；
2. 直接阅读产品源代码；
3. 在一个你可以独占的环境上执行 GUI 操作产生测试数据，与此同时，监控独占环境的数据库端业务表的变化，找到哪些业务表发生了变化。

通过数据库操作生成测试数据的方法，主要优点是测试数据的生成效率非常高，可以在较短的时间内创建大批量的测试数据。

当然，这个方法的缺点也非常明显，主要体现在以下几个方面：

很多时候，一个前端操作引发的数据创建，往往会修改很多张表，因此封装的数据准备函数的维护成本要高得多；

容易出现数据不完整的情况，比如一个业务操作，实际上在一张主表和一张附表中插入了记录，但是基于数据库操作的数据创建可能只在主表中插入了记录，这种错误一般都会比较隐蔽，往往只在一些特定的操作下才会发生异常；

当业务逻辑发生变化时，即 SQL 语句有变化时，需要维护和更新已经封装的数据准备函数。

综合运用 API 和数据库的方式生成测试数据

目前，在实际的工程实践中，很少使用单一的方法生成测试数据，基本都是采用 API 和数据库相结合的方式。**最典型的应用场景是，先通过 API 调用生成基础的测试数据，然后使用数据库的 CRUD 操作生成符合特殊测试需求的数据。**所以，你经常会看到很多的数据准备函数中，既有 API 操作，又有数据库操作。

我以创建用户为例，和你分享一下如何综合运用 API 和数据库两种方式创建测试数据吧。

假设，我们需要封装一个创建用户的函数，这个函数需要对外暴露“用户国家”和“支付方式”这两个参数。由于实际创建用户是通过后台 createUser API 完成的，但是这个 API 并不支持指定“用户国家”和“支付方式”，所以我们就需要自己封装一个创建用户的函数。

自己封装用户创建函数的方法，你可以通过下面这个思路实现：

首先，调用 createUser API 完成基本用户的创建；

然后，调用 paymentMethod API 实现用户对于不同支付方式的绑定，其中 paymentMethod API 使用的 userID 就是上一步中 createUser API 产生的用户 ID；

最后，通过数据库的 SQL 语句更新“用户国家”。

在这个例子中，createUser API 和 paymentMethod API 只是为了说明如何综合运用 API 的顺序调用，而其具体参数并不是我要阐述的关键内容，所以我并没有和你详细说明这两个 API 的参数、实现方式等问题。另外，我在最后一步综合运用了数据库的 CRUD 操作，完成了创建测试数据的全部工作。

这，就是一个封装测试数据准备函数的典型例子了。

总结

今天，我从测试数据创建的角度，和你分享了准备测试数据的四种方法。

其中，基于 GUI 操作生成测试数据是最原始的方法，但是效率很低，而且会引入不必要的依赖；通过 API 调用以及数据库操作的方式生成测试数据是目前主流的做法，通过 API 调用的方式具有数据准确度高但是创建效率较低的特点，而通过数据库的方式具有创建效率高但是维护复杂度也高的特点。

所以，在实际项目中，业界往往会综合采用 API 和数据库的方式生成测试数据，即通过 API 调用生成基础数据，然后使用数据库的 CRUD 操作进一步生成符合特殊测试需求的数据。

思考题

目前，我们需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。你在实际工作中遇到过这类测试数据吗？你又是如何处理的呢？

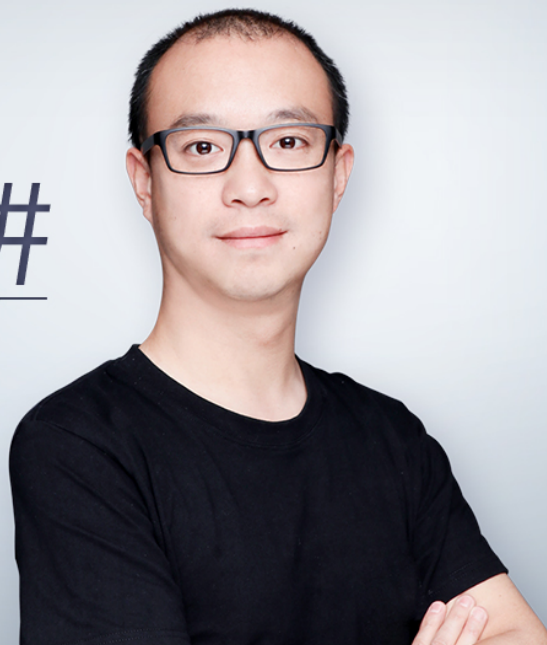
感谢你的收听，欢迎你给我留言。



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 34 | 站在巨人的肩膀：企业级实际性能测试案例与经验分享

下一篇 36 | 浅谈测试数据的痛点

精选留言 (12)

写留言



wenjun

2018-11-01

3

测试数据这个问题困扰了我很久，基本上成为团队进一步发展的瓶颈之一。想请教一下老师，对于前台和中台团队，下面难点怎么突破：

- 1.最终数据，部分数据不是实时生成的，例如需要后台清算才能产生。
- 2.依赖于多个后台系统，他们分别产生不同的数据，存储在不同的数据库中。中台需要整个这些数据。...

展开 ▾



楚耳

2018-10-31

👍 2

老师，你文中提到的在一个你可以独占的环境上执监控服务器端的调用日志，和数据库日志，这其中分别用的是什么工具进行监控的



sylan215

2018-09-17

👍 2

1.如果使用非 GUI 操作获取测试数据，一定要保证使用的方法和 GUI 操作路径/逻辑是一致的，不然没法保证测试的一致性了；

2.如果是一次性操作，其实 GUI 操作的成本并不高(或者使用 GUI 操作的接口，而不是额外开发的接口)，其他方法可能会涉及单独的开发工作量吧，特别是数据库，大部分业务...

展开 ▾



夏洛克的救...

2018-10-01

👍 1

“基于 GUI 的测试数据创建方法不适合封装成测试数据工具”

测试数据工具？用JavaScript实现？

展开 ▾

作者回复: 如果基于gui来封装测试数据工具，就会变成用gui来创建测试数据，这样的效率非常差。测试数据工具一般采用java或者python实现居多，当然用js也是可以的，这要取决于你们公司的技术架构和技术栈



口水窝

2019-05-13

👍

消息队列就是两台机器之间的通讯协议，消息存储于一个容器中。没有真正深入到消息队列，所以里面的创建原理并不熟悉，记录，下次学习。



Sphinx

👍



2018-11-19

开发在做UT测试的时候也会需要准备测试数据，若是底层服务，会直接在测试用例中通过ORM来创建所需数据；若是上层服务，通常会调用底层服务的API来创建所需数据，微服务架构下，我觉得使用API方式来构建测试数据会比较理想，这样更利于测试代码的抽象与实现。



静静张

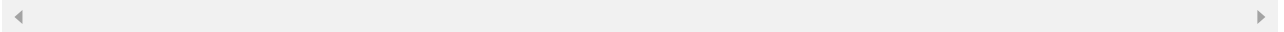
2018-11-18



我们是在测试用例执行前通过api创建数据,执行后清除数据的方式。

展开 ∨

作者回复: 完全可行，取决被测系统的特性



小老鼠

2018-11-07



测试数据

- 1、在测试执行前创建完毕
 - 2、在测试前通过setup来创建
- 这两个方法各有什么利弊？



小老鼠

2018-11-04



各位作自动化测试的时候，测试数据是先存在系统数据库中还是在运行测试程序开始生成，完成以后再删除，有没有必要一批测试用例执行完毕，不侵犯系统中原有的数据。今天我发现我以前写的一些测试用例只接操纵产品数据库，但是忘了启动web server，但启动了DB server，然后运行测试代码往DB server里写数据了，但由于web server没有启动，测试用例运行失败，结果写进数据库中的数据没有及时处理干净，形成了垃圾测试...

展开 ∨



胖虫子

2018-11-01



遇到过一次测试，很多的测试数据需要操作很多步骤才能得到，只是为了测试方便，也为了节约时间，采用katalon提供的web录制回访，虽然也是gui的，但因为是录制回访，所以也没花太多的时间

展开 ▾



雨注金池

2018-10-25



茹老师，请问如果通过api更新数据库业务上需要发消息给上游，那么通过CRUD数据库的方法直接构造特殊数据并不会发消息，那是需要再构造Q么？这样成本会不会很高

展开 ▾



深邃

2018-09-17



我现在还是不懂怎么用GUI测试

展开 ▾