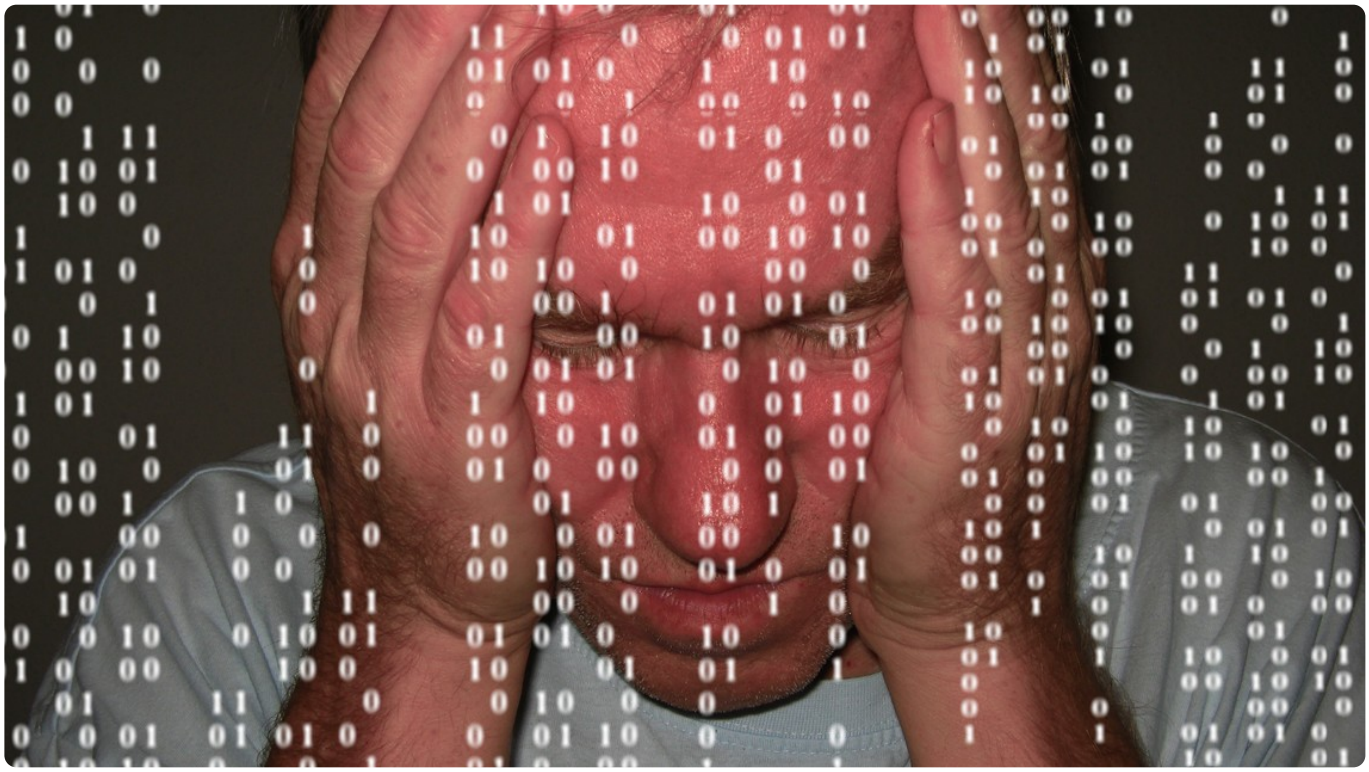
15 | 过不了的坎：聊聊GUI自动化过程中的测试数据

2018-08-01 茹炳晟

软件测试52讲 进入课程 >



讲述：茹炳晟
时长 13:26 大小 6.16M

在前面几篇文章中，我从页面操作的角度介绍了 GUI 自动化测试，讲解了页面对象模型和业务流程封装，今天我将从测试数据的角度再来谈谈 GUI 自动化测试。

为了顺利进行 GUI 测试，往往需要准备测试数据来配合测试的进行，如果不采用事先数据准备的方式，测试效率将会大打折扣，而且还会引入大量不必要的依赖关系。

以“用户登录”功能的测试为例，如果你的目的仅仅是测试用户是否可以正常登录，比较理想的方式是这个用户已经存在于被测系统中了，或者你可以通过很方便的方式在测试用例中生成这个用户。否则，难道你要为了测试用户登录功能，而以 GUI 的方式当场注册一个新用户吗？显然，这是不可取的。

其实从这里，你就可以看出测试数据准备是实现测试用例解耦的重要手段，你完全不必为了测试 GUI 用户登录功能而去执行用户注册，只要你能够有方法快速创建出这个登录用户就可以了。

在正式讨论测试数据的创建方法前，我先来分析一下 GUI 测试中两种常见的数据类型：

> 第一大类是，测试输入数据，也就是 GUI 测试过程中，通过界面输入的数据。比如“用户登录”测试中输入的用户名和密码就就属于这一类数据；再比如，数据驱动测试中的测试数据，也是指这一类。
>
> 第二大类是，为了完成 GUI 测试而需要准备的测试数据。比如，“用户登录”测试中，我们需要事先准备好用户账户，以便进行用户的登录测试。今天我分享的测试数据创建的方法，也都是围着这一部分的数据展开的。

那么接下来，我就带你一起去看看创建测试数据的方法都有哪些，以及它们各自的优缺点，和适用场景。

从创建的技术手段上来讲，创建测试数据的方法主要分为三种：

1. API 调用；
2. 数据库操作；
3. 综合运用 API 调用和数据库操作。

从创建的时机来讲，创建测试数据的方法主要分为两种：

1. 测试用例执行过程中，实时创建测试数据，我们通常称这种方式为 On-the-fly。
2. 测试用例执行前，事先创建好“开箱即用”的测试数据，我们通常称这种方式为 Out-of-box。

在实际项目中，对于创建数据的技术手段而言，最佳的选择是利用 API 来创建数据，只有当 API 不能满足数据创建的需求时，才会使用数据库操作的手段。

实际上，往往很多测试数据的创建是基于 API 和数据库操作两者的结合来完成，即先通过 API 创建基本的数据，然后调用数据库操作来修改数据，以达到对测试数据的特定要求。

而对于创建数据的时机，在实际项目中，往往是 On-the-fly 和 Out-of-box 结合在一起使用。

对于相对稳定的测试数据，比如商品类型、图书类型等，往往采用 Out-of-box 的方式以提高效率；而对于那些只能一次性使用的测试数据，比如商品、订单、优惠券等，往往采用 On-the-fly 的方式以保证不存在脏数据问题。

接下来，我就先从测试数据创建的技术手段开始今天的分享吧。

基于 API 调用创建测试数据

先看一个电商网站“新用户注册”的例子，当用户通过 GUI 界面完成新用户注册信息填写后，向系统后台递交表单，系统后台就会调用 createUser 的 API 完成用户的创建。

而互联网产品，尤其是现在大量采用微服务架构的网站，这个 API 往往以 Web Service 的形式暴露接口。那么，在这种架构下，你完全可以直接调用这个 API 来创建新用户，而无须再向后台递交表单。

由于 API 通常都有安全相关的 token 机制来保护，所以实际项目中，通常会把对这些 API 的调用以代码的形式封装为测试数据工具（Test Data Utility）。

这种方式最大的好处就是，测试数据的准确性直接由产品 API 保证，缺点是并不是所有的测试数据都有相关的 API 来支持。

另外，对需要大量创建数据的测试来说，基于 API 调用方式的执行效率，即使采用了并发机制也不会十分理想。为了解决执行效率的问题，就有了基于数据库操作的测试数据创建手段。

基于数据库操作创建测试数据

实际项目中，并不是所有的数据都可以通过 API 的方式实现创建和修改，很多数据的创建和修改直接在产品代码内完成，而且并没有对外暴露供测试使用的接口。

那么，这种情况下，你就需要通过直接操作数据库的方式来产生测试数据。

同样地，我们可以把创建和修改数据的相关 SQL 语句封装成测试数据工具，以方便测试用例的使用。但是，如果你正尝试在实际项目中运用这个方法，不可避免地会遇到如何才能找到正确的 SQL 语句来创建和修改数据的问题。

因为，创建或修改一条测试数据往往会涉及很多业务表，任何的遗漏都会造成测试数据的不准确，从而导致有些测试因为数据问题而无法进行。

那么，现在我就提供两个思路来帮你解决这个问题：

1. 手工方式。查阅设计文档和产品代码，找到相关的 SQL 语句集合。或者，直接找开发人员索要相关的 SQL 语句集合。
2. 自动方式。在测试环境中，先在只有一个活跃用户的情况下，通过 GUI 界面操作完成数据的创建、修改，然后利用数据库监控工具获取这段时间内所有的业务表修改记录，以此为依据开发 SQL 语句集。

需要注意的是，这两种思路的前提都是，假定产品功能正确，否则就会出现“一错到底”的尴尬局面。

基于数据库操作创建测试数据的最大好处是，可以创建和修改 API 不支持的测试数据，并且由于是直接数据库操作，执行效率会远远高于 API 调用方法。

但是，数据库操作这种方式的缺点也显而易见，数据库表操作的任何变更，都必须同步更新测试数据工具中的 SQL 语句。

但很不幸的是，在实际项目中，经常出现因为 SQL 语句更新不及时而导致测试数据错误的问题，而且这里的数据不准确往往只是局部错误，因此这类问题往往比较隐蔽，只有在特定的测试场景下才会暴露。

所以，在实际工程项目中，需要引入测试数据工具的版本管理，并通过开发流程来保证 SQL 的变更能够及时通知到测试数据工具团队。

综合运用 API 调用和数据库操作创建测试数据

你如果已经理解了基于 API 调用和基于数据库操作创建测试数据这两类方法，那么综合运用这两类方法，就是使得测试数据工具能够提供更多种类的业务测试数据。

具体来讲，当你要创建一种特定的测试数据时，你发现没有直接 API 支持，但是可以通过 API 先创建一个基本的数据，然后再通过修改数据库的方式来更新这个数据，以此来达到创建特定测试数据的要求。

比如，你需要创建一个已经绑定了信用卡的新用户，如果创建新用户有直接的 API，而绑定信用卡需要操作数据库，那这种情况下就需要综合运用这两种方式完成测试数据工具的开发。

实时创建数据：On-the-fly

GUI 测试脚本中，在开始执行界面操作前，我们往往会通过调用测试数据工具实时创建测试数据，也就是 On-the-fly 方式。

这种方式不依赖被测试系统中的任何原有数据，也不会对原有数据产生影响，可以很好地从数据层面隔离测试用例，让测试用例实现“自包含”。

从理论上讲，On-the-fly 是很好的方法，但在实际测试项目中却并不是那么回事儿，往往会存在三个问题：

1. **在用例执行过程中实时创建数据，导致测试的执行时间比较长。** 我曾经粗略统计过一个大型 Web GUI 自动化测试项目的执行时间，将近 30% 的时间都花在了测试数据的准备上。
2. **业务数据的连带关系，导致测试数据的创建效率非常低。** 比如，你需要创建一个订单数据，而这个订单必然会绑定买家和卖家，以及订单商品信息。
   如果完全基于 On-the-fly 模式，你就需要先实时创建买家和卖家这两个用户，然后再创建订单中的商品，最后才是创建这个订单本身。
   显然，这样的测试数据创建方式虽然是“自包含”的，但创建效率非常低，会使得测试用例执行时间变得更长，而这恰恰与互联网产品的测试策略产生冲突。
3. **更糟糕的情况是，实时创建测试数据的方式对测试环境的依赖性很强。** 比如，你要测试用户登录功能，基于 On-the-fly 方式，你就应该先调用测试数据工具实时创建一个用户，然后再用这个用户完成登录测试。
   这时，创建用户的 API 由于各种原因处于不可用的状态（这种情况在采用微服务架构的系统中很常见），那么这时就会因为无法创建用户，而无法完成用户登录测试。

基于这三种常见问题，实际项目中还会引入 Out-of-box 方式（即在执行测试用例前，预先创建好测试数据）准备测试数据。

事先创建测试数据：Out-of-box

Out-of-box 的含义是开箱即用，也就是说，已经在被测系统中预先创建好了充足的、典型的测试数据。这些数据通常是在搭建测试环境时通过数据库脚本“预埋”在系统中的，后续的测试用例可以直接使用。

Out-of-box 的方式有效解决了 On-the-fly 的很多问题，但是这种方法的缺点也很明显，主要体现在以下三个方面：

1. **测试用例中需要硬编码（hardcode）测试数据，额外引入了测试数据和用例之间的依赖。**
2. **只能被一次性使用的测试数据不适合 Out-of-box 的方式。** 测试用例往往会需要修改测试数据，而且有些测试数据只能被一次性使用。比如，一个商品被买下一次后就不能再用了；再比如，优惠券在一个订单中被使用后，就失效了，等等。所以如果没有很好的全局测试数据管理，很容易因为测试数据失效而造成测试失败。
3. **“预埋”的测试数据的可靠性远不如实时创建的数据。** 在测试用例执行过程中，经常会出现测试数据被修改的情况。比如，手动测试，或者是自动化测试用例的调试等情况。

On-the-fly 和 Out-of-box 的互补

基于 On-the-fly 和 Out-of-box 的优缺点和互补性，在实际的大型测试项目中，我们往往会采用两者相结合的方式，从测试数据本身的特点入手，选取不同的测试数据创建方式。

针对应该选择什么时机创建测试数据，结合多年的实践经验，我为你总结了以下三点：

1. 对于相对稳定、很少有修改的数据，建议采用 Out-of-box 的方式，比如商品类目、厂商品牌、部分标准的卖家和买家账号等。
2. 对于一次性使用、经常需要修改、状态经常变化的数据，建议使用 On-the-fly 的方式。
3. 用 On-the-fly 方式创建测试数据时，上游数据的创建可以采用 Out-of-box 方式，以提高测试数据创建的效率。以订单数据为例，订单的创建可以采用 On-the-fly 方式，而与订单相关联的卖家、买家和商品信息可以使用 Out-of-box 方式创建。

其实，为了更好地解决测试数据本身组合的复杂性和多样性，充分发挥测试数据工具的威力，还有很多大型企业的最佳实践值得讨论，在本专栏后面的测试数据章节，我会再为你详细介绍。

总结

今天我从创建测试数据的技术手段和时机两个方面，介绍了 GUI 测试数据的准备。

在实际测试项目中，往往需要综合运用 API 调用和数据库操作来创建测试数据，并且会根据测试数据自身的特点，分而治之地采用 On-the-fly 和 Out-of-box 的方式，以寻求数据稳定性和数据准备效率之间的最佳平衡。

思考题

你所在的公司是如何准备 GUI 测试的测试数据的？遇到了哪些问题，对应的有哪些解决方案呢？

欢迎你给我留言。

© 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　14 | 更接近业务的抽象：让自动化测试脚本更好地描述业务

下一篇　16 | 脑洞大开：GUI测试还能这么玩（Page Code Gen + Data Gen + Headless）？

精选留言 (15)

写留言

假装乐
2018-08-01

24

数据库监控工具有推荐的吗

展开

sylan215
2018-08-01

6

是不是可以这么理解：
API 调用和数据库操作创建，本质上都是操作数据库，不过 API 调用是做了一层封装，保证了操作的可控性（避免胡乱写数据库操作语句）。
实时创建数据和事先创建测试数据，其实也是不冲突的，我理解他俩并不是互斥的关系，而是互为补充，在 API 调用逻辑内部，先检查数据库中是否存在需要的测试数据，存在...

展开

年轻人的瞎...
2018-11-08

2

我们是out the box 脚本预制 然后on the fly 接口调用，API测试，经常因为接口变动大，数据库也有变化 这样脚本经常容易改动 有什么方法可以设置变量方面，灵活性的脚本?

展开

arthur
2018-08-06

2

我们的产品有一个best practice的包，里面包含了很多数据，对测试非常有用

展开

FamilyLi
2018-08-02

2

最近几张讲的GUI测试，听起来主要是基于浏览器的业务测试，对于APP的测试如何应用

叶夏立
2018-08-01

2

我的做法是备份还原整个数据库😂，当然也是看业务场景的

展开

口水窝

2019-03-29

1

小公司，没有做GUI自动化测试，更无从测试数据的准备谈起，只能自己摸索，不断尝试，总结更多的实践经验。

任大树

2018-09-21

1

老师讲的很清楚～～我有个小问题想请教一下：自动化做完 要进行数据还原，老师有没有什么数据还原的方法推荐呢？比如数据库快照什么的。或者说有哩些类型的自动化测试根本不用还原？

展开

Geek_AX1

2019-06-04

老师，我以前做过一个项目，测试数据我们直接copy一些现网数据来创造数据，尤其是性能测试高并发的时候，请问这个是不是一种好方法呢？

孙建伟

2019-04-24

并不矛盾，GUI自动化测试是基于功能稳定的情况下进行的！

展开

Lynn

2018-12-13

数据库监控工具有推荐的吗

展开

小老鼠

2018-10-24

1，本文中需要注意的是，这两种思路的前提都是，假定产品功能正确，否则就会出现“一

错到底”的尴尬局面。——- 前题是产品功能正确，测试的目的是找到产品中的Bug，没觉得这有矛盾吗?
2，在自动化测试中，teatdown方法中往往作的最重要的事情是清除脏数据。但是自动化测试往往出现的状况是测试程序在测试过程中遇到问题，挂掉了，这样造成的结果是执...
展开

晴天
2018-08-10

hui测试的两类数据感觉没有什么区别，老师能详细说下嘛
展开

> 作者回复: 其实这里是从两个不同的角度来描述测试数据，一种是测试输入数据，也就是你的数据驱动中用到的数据，另一种是讲你怎么去创建这个测试数据。

涅槃Ls
2018-08-03

打卡15
展开

hi！girl
2018-08-02

在准备测试数据中，我觉得应该尽量减少第三方的依赖，避免脚本的不稳定性，也就是说能预先设定的就先考虑，不能的再采取实时产生的方式