

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

2018-08-03 茹炳晟

软件测试52讲

[进入课程 >](#)



讲述：茹炳晟

时长 09:54 大小 4.54M



在前面的几篇文章中，我介绍了 GUI 自动化测试的数据驱动测试、页面对象（Page Object）模型、业务流程封装，以及测试数据相关的内容。

今天这篇文章，我将从页面对象自动生成、GUI 测试数据自动生成、无头浏览器三个方面展开，这也是 GUI 测试中三个比较有意思的知识点。

页面对象自动生成

在前面的文章中，我已经介绍过页面对象（Page Object）模型的概念。页面对象模型，是以 Web 页面为单位来封装页面上的控件以及控件的部分操作，而测试用例基于页面对象完成具体操作。最典型的模式就是：XXXPage.YYYComponent.ZZZOperation。

基于页面对象模型的伪代码示例，如图 1 所示。

```
1  Class loginPage{
2      username_input = findElementByName("username");
3      password_input = findElementByName("password");
4      login_ok_button = findElementByName("login_ok_button");
5      login_cancel_button = findElementByName("login_cancel_button");
6  }
7
8
9  login(username,password)
10 {
11     loginPage.username_input.input(username);
12     loginPage.password_input.input(password);
13     loginPage.login_ok_button.click();
14 }
```

图 1 基于页面对象模型的伪代码示例

如果你在实际项目中已经用过页面对象模型，你会发现开发和维护页面对象的类（Page Class），是一件很耗费时间和体力的事儿。

你需要打开页面，识别出可以唯一确定某元素的属性或者属性集合，然后把它们写到 Page Class 里，比如图 1 的第 2 行代码

username_input=findElementByName("username")，就是通过控件的名字 (username) 来定位元素的。

更糟糕的是，GUI 的页面会经常变动，如果开发人员开发前端代码时没有严格遵循可测试性的要求，Page Class 的维护成本就会更高。

那么，什么方法能够解决这个问题呢？答案就是，页面对象自动生成技术，它非常适用于需要维护大量页面对象的中大型 GUI 自动化测试项目。

页面对象自动生成技术，属于典型的“自动化你的自动化”的应用场景。它的基本思路是，你不用再手工维护 Page Class 了，只需要提供 Web 的 URL，它就会自动帮你生成这个页面上所有控件的定位信息，并自动生成 Page Class。

但是，需要注意的是，那些依赖于数据的动态页面对象也会被包含在自动生成的 Page Class 里，而这种动态页面对象通常不应该包含在 Page Class 里，所以，往往需要以手工的方式删除。

目前，很多商用自动化工具，比如 UFT，已经支持页面对象自动生成功能了，同时还能够对 Page Class 进行版本管理。

但是，开源的自动化方案，页面对象自动生成功能一般需要自己开发，并且需要与你所用的自动化测试框架深度绑定。目前，中小企业很少有自己去实现这一功能的。

不过，有个好消息是，目前国内应用还不算多、免费的 Katalon Studio，已经提供了类似的页面对象库管理功能，如果感兴趣的话，你可以去试用一下。

GUI 测试数据自动生成

GUI 测试数据自动生成，指的由机器自动生成测试用例的输入数据。

乍一听上去是不是感觉有点玄乎？机器不可能理解你的业务逻辑，怎么可能自动生成测试数据呢？

你的这个想法完全合理，并且也是完全正确的。所以，我在这里说的“测试数据自动生成”，仅仅局限于以下两种情况：

1. **根据 GUI 输入数据类型，以及对应的自定义规则库自动生成测试输入数据。** 比如，GUI 界面上有一个“书名”输入框，它的数据类型是 string。

那么，基于数据类型就可以自动生成诸如 Null、SQL 注入、超长字符串、非英语字符等测试数据。

同时，根据自定义规则库，还可以根据具体规则生成各种测试数据。这个自定义规则库里面的规则，往往反映了具体的业务逻辑。比如，对于“书名”，就会有书名不能大于多少个字符、一些典型的书名（比如，英文书名、中文书名等）等业务方面的要求，那么就可以根据这些业务要求来生成测试数据。

根据自定义规则生成测试数据的核心思想，与安全扫描软件 AppScan 基于攻击规则库自动生成和执行安全测试的方式，有异曲同工之处。

2. **对于需要组合多个测试输入数据的场景，测试数据自动生成可以自动完成多个测试数据的笛卡尔积组合，然后再以人工的方式剔除掉非法的数据组合。**

但是，这种方式并不一定是最高效的。对于输入参数比较多，且数据之间合法组合比较少或者难以明确的情况，先自动化生成笛卡尔积组合，再删除非法组合，效率往往还不如人为组合来得高。所以，在这个场景下是否要用测试数据自动生成方法，还需要具体问题具体分析。

更常见的用法是，先手动选择部分输入数据进行笛卡尔积，并删除不合法的部分；然

后，在此基础上，再人为添加更多业务上有意义的输入数据组合。

比如，输入数据有 A、B、C、D、E、F 六个参数，你可以先选取最典型的几个参数生成笛卡尔积，假设这里选取 A、B 和 C；然后，在生成的笛卡尔积中删除业务上不合法的组合；最后，再结合 D、E 和 F 的一些典型取值，构成更多的测试输入数据组合。

无头浏览器

无头浏览器，即 Headless Browser，是一种没有界面的浏览器。

什么？浏览器没有界面，还叫什么浏览器啊？别急，我将为你——道来。

无头浏览器，其实是一个特殊的浏览器，你可以把它简单地想象成是运行在内存中的浏览器。它拥有完整的浏览器内核，包括 JavaScript 解析引擎、渲染引擎等。

与普通浏览器最大的不同是，无头浏览器执行过程中看不到运行的界面，但是你依然可以用 GUI 测试框架的截图功能截取它执行中的页面。

无头浏览器的主要应用场景，包括 GUI 自动化测试、页面监控以及网络爬虫这三种。在 GUI 测试过程中，使用无头浏览器的好处主要体现在四个方面：

1. **测试执行速度更快。** 相对于普通浏览器来说，无头浏览器无需加载 CSS 以及渲染页面，在测试用例的执行速度上有很大的优势。
2. **减少对测试执行的干扰。** 可以减少操作系统以及其他软件（比如杀毒软件等）不可预期的弹出框，对浏览器测试的干扰。
3. **简化测试执行环境的搭建。** 对于大量测试用例的执行而言，可以减少对大规模 Selenium Grid 集群的依赖，GUI 测试可以直接运行在无界面的服务器上。
4. **在单机环境实现测试的并发执行。** 可以在单机上很方便地运行多个无头浏览器，实现测试用例的并发执行。

但是，无头浏览器并不完美，它最大的缺点是，不能完全模拟真实的用户行为，而且由于没有实际完成页面的渲染，所以不太适用于需要对于页面布局进行验证的场景。同时，业界也一直缺乏理想的无头浏览器方案。

在 Google 发布 Headless Chrome 之前，PhantomJS 是业界主流的无头浏览器解决方案。但是，这个项目的维护一直以来做得都不够好，已知未解决的缺陷数量多达 1800 多

个，虽然支持主流的 Webkit 浏览器内核，但是依赖的 Chrome 版本太低。所以，无头浏览器一直难以在 GUI 自动化测试中大规模应用。

但好消息是，2017 年 Google 发布了 Headless Chrome，以及与之配套的 Puppeteer 框架，Puppeteer 不仅支持最新版本的 Chrome，而且得到 Google 官方的支持，这使得无头浏览器可以在实际项目中得到更好的应用。

也正是这个原因，PhantomJS 的创建者 Ariya Hidayat 停止了它的后续维护，Headless Chrome 成了无头浏览器的首选方案。

那什么是 Puppeteer 呢？Puppeteer 是一个 Node 库，提供了高级别的 API 封装，这些 API 会通过 Chrome DevTools Protocol 与 Headless Chrome 的交互达到自动化操作的目的。

Puppeteer 也是由 Google 开发的，所以它可以很好地支持 Headless Chrome 以及后续 Chrome 的版本更新。

如果你也迫不及待地想要尝试把 Headless Chrome 应用到自己的 GUI 测试中，那还等什么，赶紧下载并开始吧。

总结

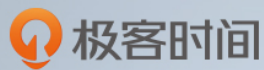
我分别介绍了无头浏览器、页面对象自动生成，以及 GUI 测试数据自动生成，这三个 GUI 测试中比较有意思的知识点，包括它们的概念、应用场景等内容。

1. 对于页面对象自动生成，商用测试软件已经实现了这个功能。但是，如果你选择开源测试框架，就需要自己实现这个功能了。
2. GUI 测试数据自动生成，主要是基于测试输入数据的类型以及对应的自定义规则库实现的，并且对于多个测试输入数据，可以基于笛卡尔积来自动组合出完整的测试用例集合。
3. 对于无头浏览器，你可以把它简单地想象成运行在内存中的浏览器，它拥有完整的浏览器内核。与普通浏览器最大的不同是，它在执行过程中看不到运行的界面。目前，Headless Chrome 结合 Puppeteer 是最先进的无头浏览器方案，如果感兴趣，你可以下载试用。

思考题

在你的工作中，还有哪些好的方法和实践可以提高 GUI 自动化测试的效率吗？

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 15 | 过不了的坎：聊聊GUI自动化过程中的测试数据

下一篇 17 | 精益求精：聊聊提高GUI测试稳定性的关键技术

精选留言 (19)

写留言



Geek_84a77...

2018-08-03

14

老师，是每篇文章有时间限制吗？可否细致的讲解一下，如何自动化生成页面对象，如何自动化生成测试数据？我们现在的理解很大部分是停留在概念上，来这儿学习的理由也是老师能够给我们一个直观的认识关于这些技术，或者每篇文章能否提供个链接供我们看看源码，切身体验一下封装等其他文章中提到的技术，多谢

展开 ∨

作者回复: 关于页面对象自动生成的具体实现细节还是比较复杂的, 因为采用不同前端开发框架的页面的生成还都不太一样, 我看看是否有可能搞个git给点示例代码。至于测试数据自动生成主要是思想方法, 用excel + vba就能轻易完成大部分的功能, 当然用代码实现也是可以的, 但是文章本身还是以讲方法为主, 不会去讲具体怎么写代码来实现这个功能。



sylan215

2018-08-03

👍 13

- 1.如果使用 selenium + xpath, 应该可以解决大部分的控件识别, chrome 支持右键一键拷贝 xpath, 贼方便;
- 2.关于测试数据的自动生成, 个人感觉目前测试中, 场景测试重要性是最高的, 但是自动生成可能解决不了场景覆盖的问题, 不过如果借鉴最新的机器学习算法, 说不定有新发现; ...

展开 ▾

作者回复: 高质量的留言。👍! 自动测试数据生成的应用场景主要是覆盖边界值和典型易出错场景, 应用价值还是比较大的。关于无头浏览器, 目前的应用领域主要在爬虫和devops中的环境健康检查, 就是去看一下网站基本的页面是否可以打开, 最最基本的smoke用例是否可以通过, 如果不行, 那就会把环境健康状态标红, 并从可用列表中移除



猪猪

2018-08-03

👍 4

老师讲一下如何自动生成页面模型的原理, 不要一带而过

展开 ▾

作者回复: 这个具体的我实现细节还是有点复杂的, 如果是静态页面, 那就好解析和分类页面元素的源代码, 如果页面是react页面, 那就要要基于react来做解析, 不同前端框架的页面解析还都不同, 有很多细节要处理。



杜艳

2018-08-03

👍 3

建议可以不可以不要伪代码。写一个真实可用的java代码

展开 ▾

作者回复: 采用伪代码的目的是为了更简单直观的说明问题的本质以及代码的实现思路, 而尽量避免过多的不相关细节, 如果使用实际java代码, 放在文档中你看到的就是大段的代码, 说明问题本质的代码就会混在其中, 不容易说明问题。另外, 现在还有很多自动化测试是基于python和ruby的, 所以说明原理的部分我还是倾向于使用伪代码。后续文章会有一些实际的例子, 那时候我会采用真实可用的java代码。

,



我的summer

2018-08-07

👍 2

GUI测试数据的自动生成中第二种情况, 可以使用PICT小工具。使用一定限制条件对笛卡尔积结果筛选出已经剔除的部分组合, 然后再人工确认, 可以提高一些些效率

展开 ∨



口水窝

2019-03-29

👍 1

没有深入实践自动化测试, 更无从谈起提高测试效率。

展开 ∨



胖虫子

2018-08-31

👍 1

为什么大家用类似katalon这样的工具热情不高, 但自己去写自动化框架的热情高

作者回复: 蛮同意你的观点的, 很多也是处于测试人员自身技能发展的考虑



Geek_007

2019-05-05

👍

你好 请问关于GUI测试 还有一些桌面应用软件(非浏览器), 应该怎么来做自动化测试呢?



johnny

2018-11-20

👍

老师，希望能在git提供部分章节的代码实现。比如第14节的内容看完还是只停留在概念，如果有示例代码就比较方便理解

作者回复: 很好的建议，下次一定提供完整可运行的代码示例，但是讲解还是采用伪代码可以帮助更好的理解



小老鼠

2018-10-24



1, " 无头浏览器 "、" 页面对象自动生成 "，以及 " GUI 测试数据自动生成 " 这三个技术现在在企业中用得有多吗？

2、" selenium for Java中的htmlunit是不是属于无头浏览器？

展开 ▾



subona

2018-10-22



希望老师能深入地讲解下怎么去做，感觉这几节课都是概念性的知识点，太表面了。听完对于怎样运用到项目中，把工作做得更好没有太多帮助



kaiserin

2018-10-09



Katalon Studio现在的普及率怎么样？感觉大多数人还是用的robotframework

作者回复: 现在来看，对于互联网企业，你提的这两个框架的普及率都不算太高。



欧晓鸥

2018-08-13



关于数据自动化生成有例子吗？

展开 ▾



欧晓鸥

2018-08-13



对于一个即将被替代的项目，要做自动化，覆盖率是最主要的吗？

展开 ▾



雪哥

2018-08-07



茹老师，请问自己实现页面元素自动生成，一般用什么技术，指点一下大概方向就可以



星

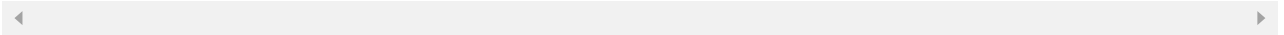
2018-08-05



如何生成页面对象？

展开 ▾

作者回复: 这个问题不是一两句话可以说清楚的，回头我去搞个git的代码示例。



soul

2018-08-04



Katalon Studio 已经操作 不错

展开 ▾



橄榄

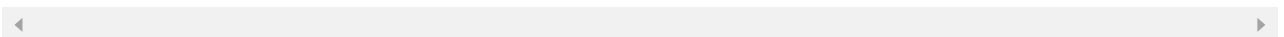
2018-08-03



没有实际应用过

展开 ▾

作者回复: 没关系，知道有这些概念和方法就好，uft等商业gui自动化工具都已经实现了部分功能，使用起来还是很简单



Robert小七

2018-08-03



企业实战中，无头浏览器的应用程度如何？是否可以用多线程来替代grid集群

展开 ▾

作者回复: 无头浏览器的应用并不是太多, 很多是用来做爬虫, 自动化测试的应用相对比较少, 是的, 完全可以用selenium grid来代替。但是最近由于google官方发布了headless chrome和对应的测试框架, 所以估计后面会有更多的应用。

