

自动化测试概述(六)-隐藏的后端工具

执行工具的本质

工具人的要求

工具的要求

工具的结构

结尾

上文说道了接口工程师的职责和内容，本章则是阐述工具开发工程师的职责和内容

执行工具的本质

当你完成了脚本的结构定义，脚本接口的定义和扩充，本质上，脚本已经是一个成品了，只是无法被执行而已。这是就需要执行工具上场了。

所谓的执行工具就是沟通脚本解释器，将脚本转为一个个单体对象并调用执行器执行脚本获取结果，说白了就是非常有价值的工具人

工具人的要求

- 丰富的开发经验，语言体系和脚本、接口体系相同或兼容
- 丰富的测试经验，知道测试人员要什么，不要什么

实际上，测试工具的开发真的就只要这两条要求。

不管执行工具的最终形态如何，它都是服务于脚本的，脚本的基础又是脚本接口体系，因此语言的选择基本遵从于脚本接口，两者需要相同或兼容。例如C/C++编写脚本(虽然没人真会这么干)，使用JAVA或C#可以充当执行工具语言。因为这几类语言都有很好的兼容性，可以通过直接调用的方式执行，诸如Python也可以通过此方法;只是稍显复杂。

而作为一个有数的工具人也应该知道测试人员究竟要的是什么。这一点非常重要，因为执行工具的最终使用者是所有的测试人员，不仅仅是脚本执行工程师

工具的要求

- 傻瓜化

怎么方便怎么来，怎么脑残怎么来，一键启动，无脑执行;作为软件就要有作为软件的B数，执行人员的技术先天性的会差点，更不会有闲心去配置环境，诸如Python(说的就是你)之类的语言属于下限极低，上限极高的存在，各种恼人的环境配置，缺失的库，包；万一脚本开发、接口开发、工具开发任何一位不小心

配错了包，可能会占用了80%的测试执行时间去一一解决。**C/C++**也一样，side-by-side的存在，让Debug和Release用的都是不同的CRT版本。说到底，大家都把自动化测试当作了炫技的舞台(包括我)，总想彰显自己的牛逼之处。但你只是个工具啊，工具有工具的本分，GUI必须要有，CMD刷屏一般的告诉我哪出错了，我是傻瓜不会直接看log嘛？

执行无视环境，自动配置，显示要大，够显眼，操作够傻瓜；

你方便人家，人家才觉得你是牛逼的。你写个python脚本，跑一遍，就丢给执行人员，你是在恶心自己还是恶心别人？

不要说什么环境为什么不一次配置，一直不改？其实很简单，你无法确保每一次脚本的更新都不会附带一堆新增的引用库、引用包。更不要提普通测试人员是否真的有自己单独的测试环境和测试集群。财大气粗的可以这么搞，但没钱的怎么办？钱一般的怎么办？测试环境经常切换怎么办？你不能用一成不变去要求开发、主管、运维等等一堆人，既然如此，作为工具开发者，你为什么不学着让自己多思考一点？工具傻瓜化是必然，你不能以自己的技术水平要求别人也跟你一样

- **日志系统**

没有日志的工具可以重新开发了

- **稳定性**

执行工具要求一直工作或长时间工作，因此稳定性一定要高，各种意外情况都要考虑，如果跑着跑着跑挂了，算谁的责任？

- **快速修复**

测试工具也是软件，不可能没有bug，因此一旦发现bug，必然要求能够快速修复，因此工具要支持快速修复

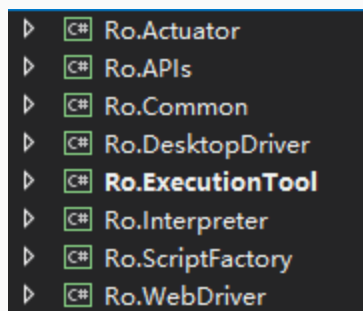
- **可拓展性**

需求永远都在加，谁知道下回你的需求者会开什么脑洞，因此工具需要能够拓展新的功能

工具的结构

从上面两篇文章中，我们已经描述了脚本结构和接口分层，其实工具也是有结构的，单纯的工具不止一个工作内容，也必然有着其他方面的工作。

从脚本到执行，是一个被赋予生命的过程，工作在其中扮演者非常重要的角色，就是赋予生命和其活力。因此工具也会按照分工具具有以下的功能，甚至更多



1. 入口

可以是一个GUI，或者是命令行式的入口，用以载入脚本库，配置一些执行参数等，并最终实时呈现结果

2. 集成解析器与扩展

集成式的脚本解析器，将脚本解析为执行器可被理解的类、属性或对象，并在入口执行相应方法；脚本不仅是步骤需要被解析，脚本结构也需要被解析，读取步骤宏、读取参数等。

3. 集成脚本接口与扩展

不同的脚本语言下，脚本接口具有的不同的意义；

- 定义脚本步骤属性
- 定义脚本步骤方法
- 定义控件元素属性
- 初始化脚本

4. 集成执行器

处于工具最底层，直接驱动各类库、架构做程序的操作，并反馈每一个步骤结果至解释器扩展

5. 日志与结果保存

时刻记录工具和脚本执行情况并输入文本和GUI

6. 通用库

存储工具所有的通用性质的对象；在条件允许的情况下，如果工具支持多语言，可以在通用库内集成部分脚本解析方法；程序配置信息等

上文可以看出，其实自动化测试工具的要求很高，并不是只是一个单纯的读取-输出的工具或几行命令行。在这个分工日趋明确的年代，自动化测试已经不仅仅会selenium+python 写两个py文件就能秒全家的年代了

结尾

自动化测试工程师要求有自己的思路、想法，不要一味的看着教程就上手做，做再多也就是个码农！

同时不要觉得领导说的都对，任何一位走上管理道路的测试工程师的理念很多时候就会被冻结在他的那个年代！

每一个自动化测试工程师都会有自己的自动化测试架构理念！这条"道"路注定充满了理念之争(大道之争)，所以一定要自信！唯有天下第一的霸气，才有独闯天涯的勇气！

