# UI自动化

# 录制、回放测试方案

### 引言

#### 文档版本

|  |  |  |  |
| --- | --- | --- | --- |
| 版本 | 作者 | 审批 | 备注 |
| V1.0 | Ai工具自由之路 | 无 |  |

#### 项目情况

|  |  |
| --- | --- |
| 项目名称 | UI自动化录制、回放工具 |
| 项目版本 | V1.0 |
| 项目经理 | Nical |
| 测试经理 | 李宗浩 |
| 所属部门 | 管线组 |

#### 文档目的

本文档主要用于指导UI自动化录制、回放测试的开展。因原UI 自动化脚本更新频繁，消耗大量人力资源。为节约维护成本，现需要一款面向所以人员的录制回放工具。其特点是维护简单，易上手，无需编程基础。

### 问题分析

#### 2.1 项目变更频繁

目前项目新增和变更频繁，倒是原UI自动化用例代码也需要频繁的变更。

所以用例编写的方式需要由纯手敲代码改为录制回放，以减少用例的编写时间。

#### 2.2 自动化代码维护成本过高

目前自动化代码编写过慢，需要大量的时间用于调试和维护，导致测试成本过高。

代码编写用例的方式，准入门槛较高，需要对框架及编程语言有一定的水准。

### 3、自动化实施目标

#### 3.1实施原则

项目采用自动化测试，主要目的是为了应对测试过程中繁重的重复性工作任务，以期达到如下效果：

* 降低测试成本
* 提高测试效率
* 更频繁的执行覆盖
* 提供更高准确和一致性
* 更好的时间利用

需要注意到的是，自动化的高效应用，对于被测系统有更高的要求，也需要遵循合理的方法流程，总结如下：

* 系统稳定性应有所保证，太多的功能变动和界面变动，会加大自动化实施难度，增加自动化维护成本；
* 自动化的实施应该被用于解决高重复性工作，否则其本身工作量投入会大于其收益，所以不应盲目对所有工作追求自动化；
* 自动化的整体实现应采用分步进行，优先覆盖功能特性稳定的模块，进而逐步扩展到整体项目的功能回归。
* 自动化测试是一个长期的过程，后续的代码维护和调优也具有可观的工作量。

#### 3.2 自动化测试范围

**系统范围：**

|  |  |  |
| --- | --- | --- |
| 自动化实施阶段 | 被测对象 | 功能点范围 |
| 第一阶段 | *系统某模块* | *P1级功能点* |
| 第二阶段 | *系统某模块* | *P1、P2级功能点* |
| ... |  |  |
| ... |  |  |

**阶段范围：**

自动化测试应主要被应用于代码提测之前的冒烟测试，以及系统测试完成之后的回归测试阶段。

### 方案对比

|  |  |  |
| --- | --- | --- |
| **工具名称** | **优点** | **缺点** |
| UIRecorder | 录制脚本可修改，断言类型丰富、  支持公共用例、高并发测试 | 环境搭建复杂、需要编程基础、  录制回放需要命令行操作 |
| QTP | 插件多、提供excel 数据源 | 脚本文件过大、数量过多、不便于脚本批量运行、查看编写脚本不方便、维护难度大 |
| katalon | 成本低、易上手、免费 | 脚本需要二次编辑、元素定位不准确、输入框无法清空 |
| Rua | 成本低、易上手、腾讯自主研发、适用于被测系统 | 工具目前没有维护人员 |

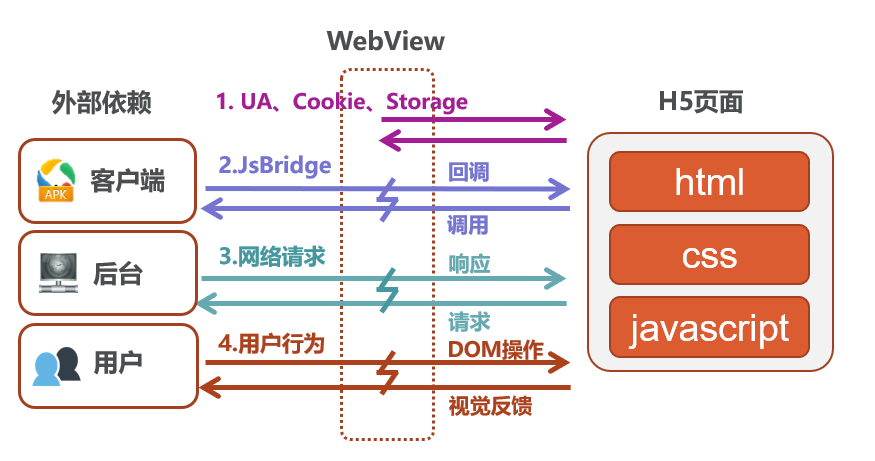
## 注：从表格中可以看出，Rua自动化工具，更适合工作台中各系统的自动化执行。

### Rua工具介绍

RUA是一款基于pupeteer实现的前端录制回放自动化测试框架。核心能力在于能够采用录制回放的方式，解偶前端页面与后台、客户端等的交互，实现稳定的前端页面。以较低的脚本录制成本，录制较稳定的自动化用例。

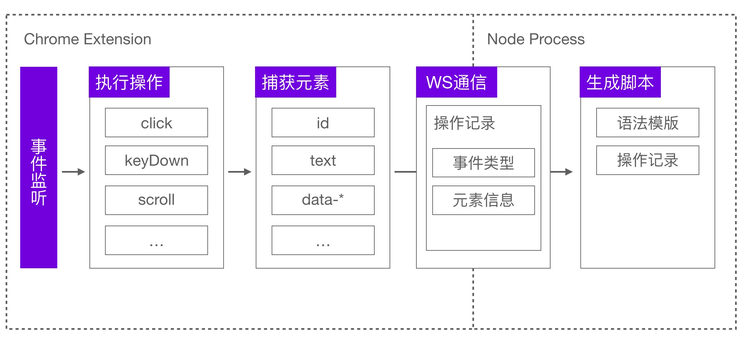
#### 5.1 整体体系

Rua自动化工具层级划分为：应用层、核心能力、外部依赖。并选取HTML、CSS、JavaScript做为编程语言实现。



#### 5.2 录制回放流程

##### 5.2.1录制流程

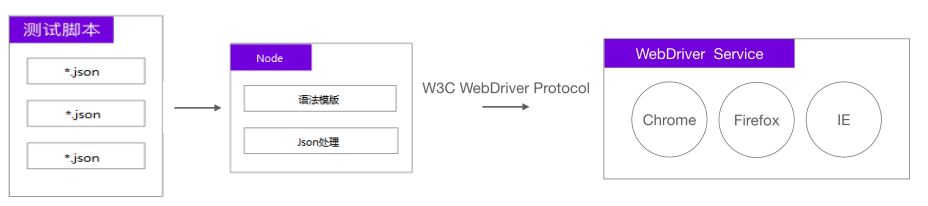


**录制功能**主要由 Chrome Extension 和 Node Process 两部分组成：

       1、Chrome Extension 会对当前浏览器进行全局事件监听，当触发某个事件（比如 click）时，即获取当前操作元素的节 点信息，同时通过 WebSocket 发送指令（开始录制、录制、结束录制）、数据（事件类型、节点信息）至 Node Process。

      2、 Node Process 接收到 Chrome Extension 的请求指令、数据，将其记录在一个数组中，当 Chrome Extension 发 起结束录制指令时，结合记录队列和配置的脚本模版，生成测试用例脚本至指定目录。

##### 5.2.2 回放流程

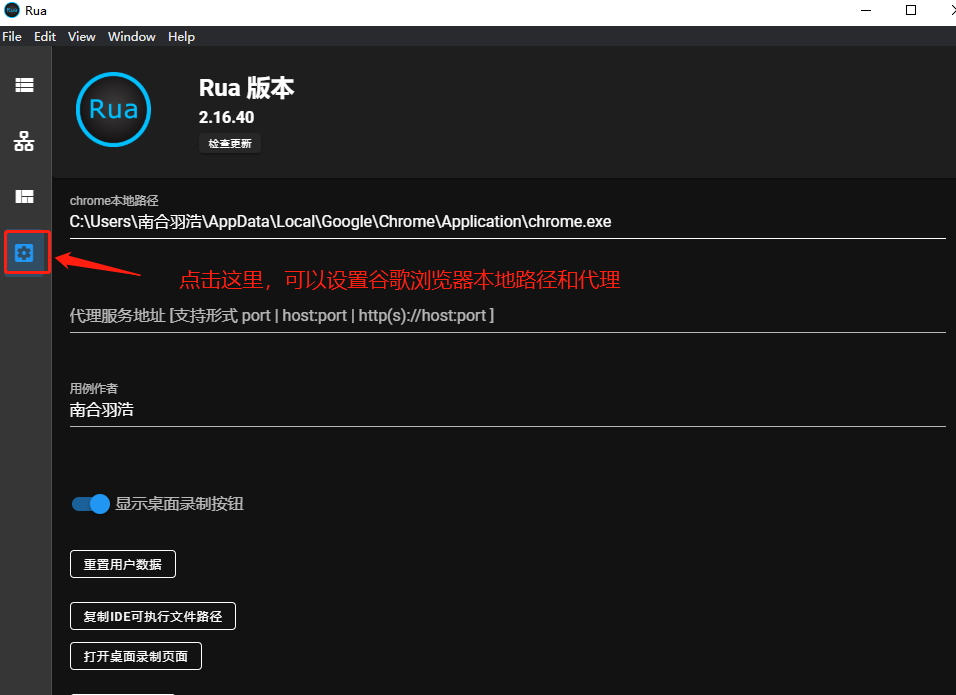


**回放功能**主要由 Node Process和webDriver 两部分组成：

1. Node Process 接收到 测试脚本数据，对其就行处理和加工，使其生成浏览器可以识别的数据形式。
2. webDriver 浏览器驱动就行对脚本的解析和执行，使录制的数据得以回放。

### Rua操作展示

#### 6.1环境设置



#### 6.2环境检查

### 

打开 **Rua** 桌面应用，会自动进入环境检测步骤，以下两项没问题即可开始录制。

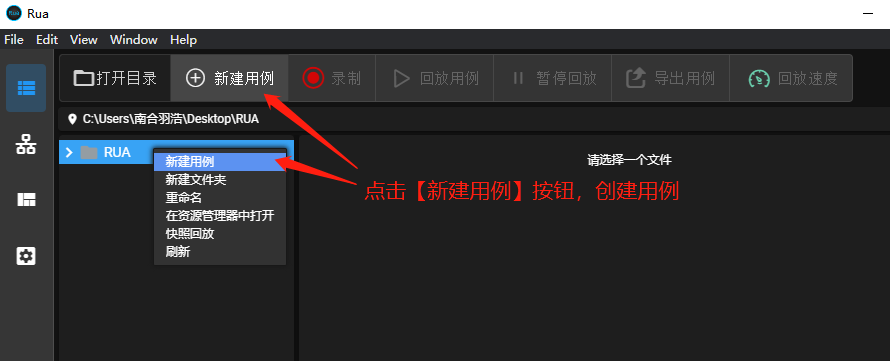
#### 6.3用例（工作）目录选择

点击**打开目录**或**请选择用例目录**按钮选择本地文件目录

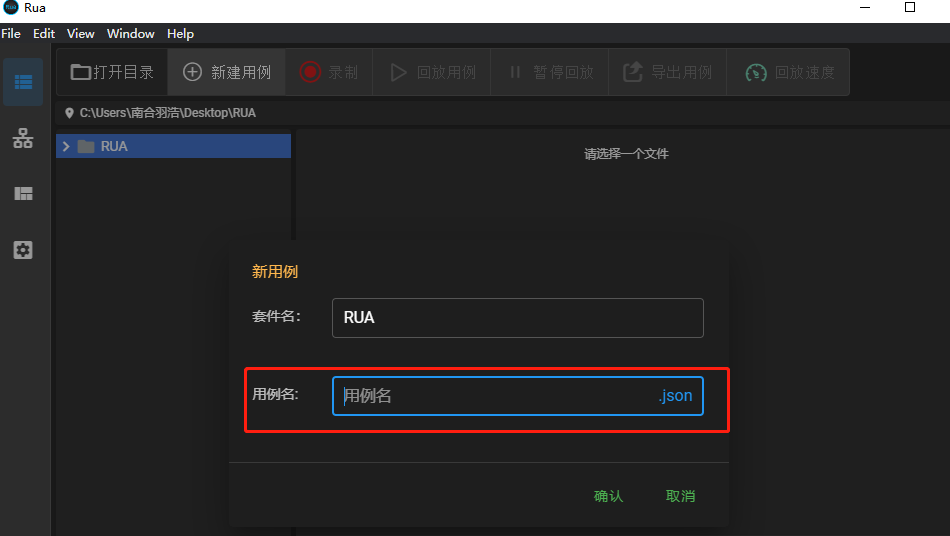


#### 6.4用例新建

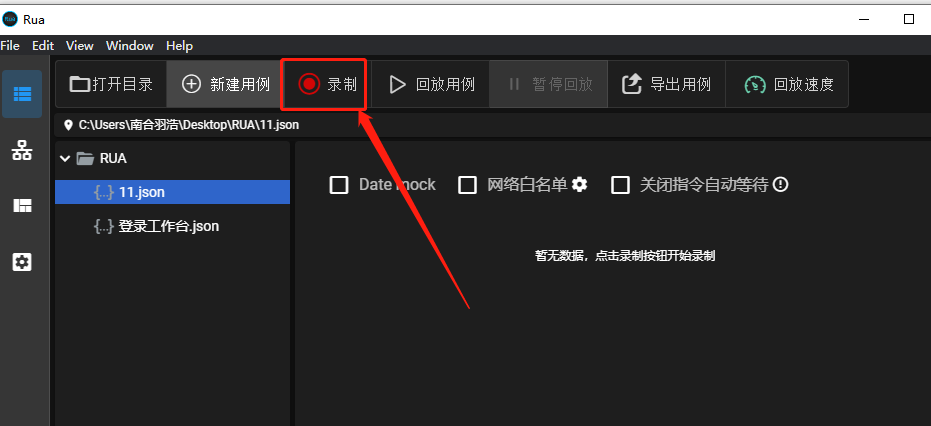
（1） 点击顶部菜单新建用例，或在文件列表中右键文件夹选择新建用例



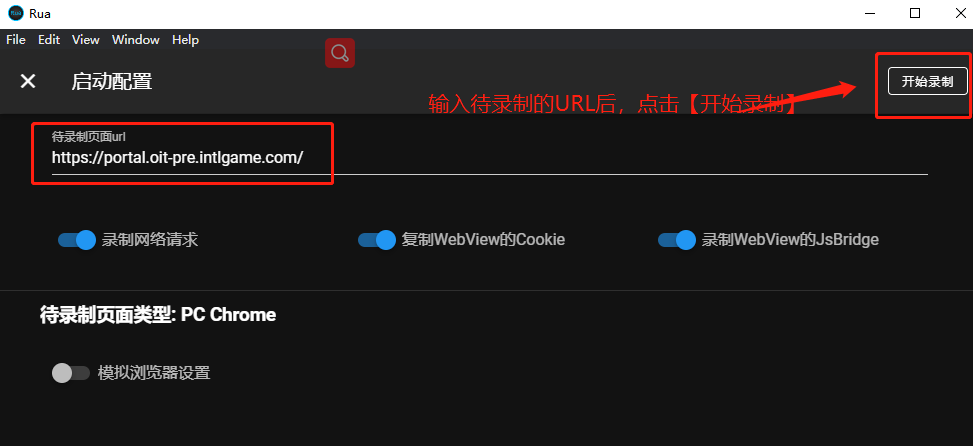
1. 输入用例文件名



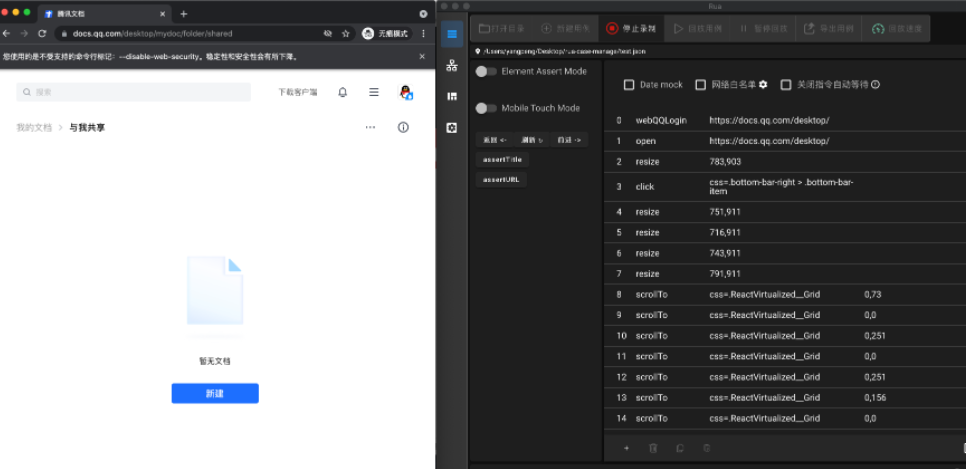
1. 开始录制
2. 选中用例文件点击**录制**



1. 输入待录制页面url，这里使用文档的页面进行录制演示**https://docs.qq.com/desktop/**



1. 点击**开始录制**，页面详细的操作流程会在右侧面板中记录

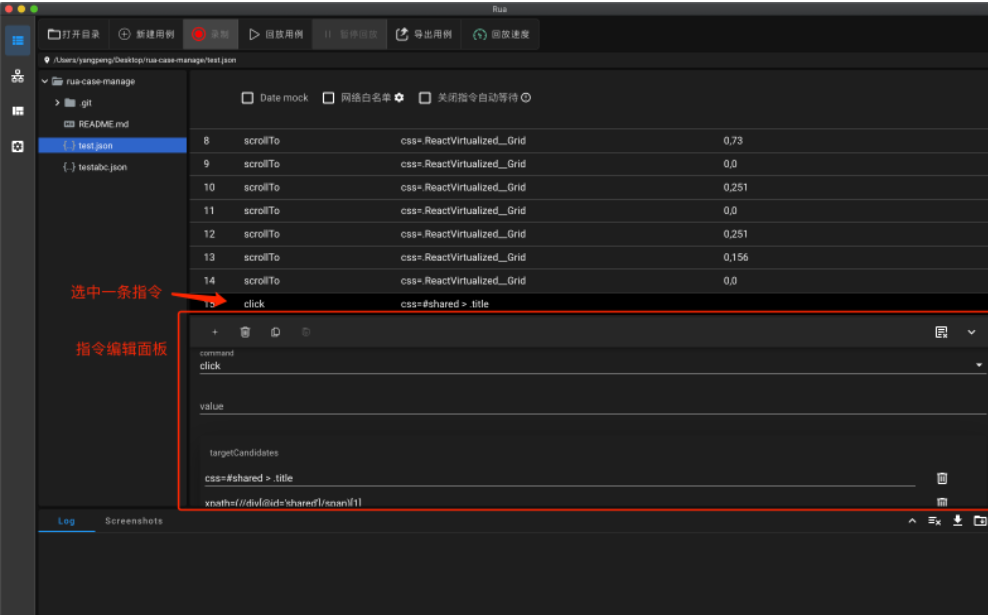


1. 结束录制

点击**停止录制**即结束录制并自动保存用例

#### 6.5 用例维护及修改

如果录制的用例有问题，可以对用例手动二次编辑



#### 6.6用例回放

选择脚本后，点击【回放用例】按钮。用例可自动执行。

