

自动化神器Playwright快速上手指南

行者AI

16 人赞同了该文章

本文首发于：[行者AI](#)

+ 关注

赞同 16

分享

Playwright是由微软公司2020年初发布的新一代自动化测试工具，相较于目前最常用的Selenium，它仅用一个API即可自动执行Chromium、Firefox、WebKit等主流浏览器自动化操作。作为针对 Python 语言纯自动化的工具，在回归测试中可更快的实现自动化。

1. 为什么选择Playwright

1.1 Playwright的优势

- (1)

Selenium需要通过WebDriver操作浏览器；Playwright通过开发者工具与浏览器交互，安装简洁，不需要安装各种Driver。
- (2)

Playwright几乎支持所有语言，且不依赖于各种Driver，通过调用内置浏览器所以启动速度更快。
- (3)

Selenium基于HTTP协议（单向通讯），Playwright基于Websocket（双向通讯）可自动获取浏览器实际情况。
- (4)

Playwright为自动等待。
- 等待元素出现（定位元素时，自动等待30s，等待时间可以自定义，单位毫秒）
- 等待事件发生

1.2 已知局限性

- (1)

Playwright不支持旧版Microsoft Edge或IE11。支持新的Microsoft Edge（在Chromium上）；所以对浏览器版本有硬性要求的项目不适用。
- (2)

需要SSL证书进行访问的网站可能无法录制，该过程需要单独定位编写。
- (3)

移动端测试是通过桌面浏览器来模拟移动设备（相当于自带模拟器），无法控制真机。

2. Playwright使用

2.1 安装

- (1)

安装Playwright依赖库（Playwright支持AsyncAwait语法，故需要Python3.7+）

```
pip install playwright
```

- (2)

安装Chromium、Firefox、WebKit等浏览器的驱动文件（内置浏览器）

```
python -m playwright install
```

2.2 自动录制

- (1)

命令行键入 --help 看到所有可后接选项

```
python -m playwright codegen --help
```

- (2)

从起始页为xingzheai.cn开始录制

```
python -m playwright codegen https://xingzheai.cn/
```

- (3)

打开xingzheai.cn，用Chromium驱动，将结果保存为my.py的python文件

```
python -m playwright codegen --target python -o 'my.py' -b chromium https://xingzheai.cn/
```

- --target：规定生成脚本的语言，有JS和Python两种，默认为Python
- -b: 指定浏览器驱动
- -o: 将录制的脚本保存到一个文件

2.3 定制化编写

- (1)

元素定位

- 选择单个元素：querySelector(engine=body)
- 选择多个元素：querySelectorAll(engine=body)
- 选择单个元素，并且自动等待：waitForSelector(engine=body)

By的8种定位方式，实际为4种

- id、name、tag name、class name（java和pythona将该4种都归为CSS）
- xpath、link text、partial link text、css selector

W3C标准规定的webdriver协议为5种定位方式

- CSS、Link text、Partial link text、Tag name、XPath

Playwright将选择器汇总为3种

- CSS、XPATH（支持逻辑表达式和函数）、TEXT

- (2)

选择器规则

- CSS：ID选择器、类选择器、元素选择器、属性选择器、通配选择器、层次选择器。
- XPath：XML路径语言，通过“路径标识符”，导航XML文档的，在类XML种（HTML）也可以使用。
- Text：结构化内容（html、xml、json）使用模糊匹配（忽略大小写，忽略前后空格，搜索子字符串）及精确匹配、非结构化内容使用正则匹配。

- (3)

元素常用操作

- 下拉选择框：selectOption、value、label、index
- 文件上传：setInputFiles、单个文件、多个文件、拖放上传
- 鼠标点击：click、dblclick
- 鼠标拖动：down、up
- 鼠标移动：move
- 触摸屏幕：tag
- 键盘按键：press
- 截屏、录屏：screenshot、recordVideo

2.4 网络拦截（Mock接口），示例如下：

```
page = context.newPage()
def Whether_intercept() -> bool:
    return True    #进行拦截
# return False  #不进行拦截

def handler(route:Route):
    print(route.request.url)
    #正常访问
    # route.continue_()
    #拒绝访问
    # route.abort("网络拦截")
    # 重定向到非目标地址
    route.fulfill(
        status=302,
        headers={
            'Location' : "https://xingzheai.cn/"
        }
    )
page.route(Whether_intercept,handler)
```

2.5 同步执行，示例如下：

```
#依次打开三个浏览器，前往行者官网，截图后退出。
from playwright import sync_playwright with sync_playwright() as p:
    for browser_type in [p.chromium, p.firefox, p.webkit]:
        # 指定为有头模式，Ture为无头模式
        browser = browser_type.launch(headless=False)
        page = browser.newPage()
        page.goto('https://xingzheai.cn/')
        # 等待页面加载完全后截图
        page.waitForSelector("text=智能内容审核")
        page.screenshot(path=f'example-{browser_type.name}.png')
        browser.close()
```

2.6 异步执行，示例如下：

```
#同时进行三个浏览器操作
import asyncio
from playwright import async_playwright
async def main():
    async with async_playwright() as p:
        for browser_type in [p.chromium, p.firefox, p.webkit]:
            browser = await browser_type.launch()
            page = await browser.newPage()
            await page.goto('https://xingzheai.cn/')
            await page.waitForSelector("text=智能内容审核")
            await page.screenshot(path=f'example-{browser_type.name}.png')
            await browser.close()
    asyncio.get_event_loop().run_until_complete(main())
```

2.7 Pytest结合，示例如下：

安装：pip install pytest-playwright

```
def test_playwright(page):
    page.goto("https://xingzheai.cn/")
    with page.expect_popup() as popup_info:
        page.click('text="智能内容审核"')
    assert "智能内容审核" == element.textContent()
```

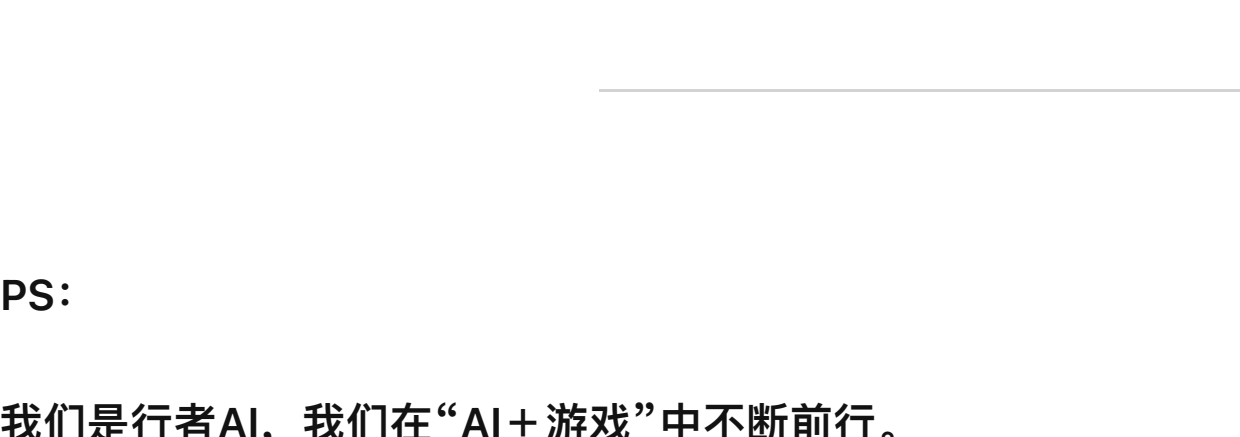
2.8 移动端操作，示例如下：

目前支持模拟机型较少，参照：[仿真设备列表](#)

```
from time import sleep
from playwright import sync_playwright
with sync_playwright() as p:
    GalaxyS5 = p.devices['Galaxy S5']
    browser = p.chromium.launch(headless=False)
    context = browser.newContext(**GalaxyS5)
    page = context.newPage()
    page.goto('https://xingzheai.cn/')
    page.click('text="智能内容审核"')
    # 截图
    # page.screenshot(path='colosseum-GalaxyS5.png')
    sleep(10)
    browser.close()
```

3. 总结

Playwright作为新一代自动化测试工具，相较于Selenium不论是易用性，还是实用性都得到了全方位的提升。做到了简约而不简单，相信使用该工具可以帮助我们提升做自动化的工作效率。



PS:

我们是行者AI，我们在“AI+游戏”中不断前行。

快来【公众号 | xingzhe_ai】，和我们讨论更多技术问题吧！

编辑于 2021-02-03 15:56

桌面浏览器

网页浏览器

Chromium

写下你的评论...

还没有评论，发表第一个评论吧

推荐阅读

C# 自动化神器 Playwright

壹零贰零

别找了，Android常用自动化工具全在这儿了！

新梦想IT

版本控制可视化神器Gource：简单易上手，效果恰似烟花秀

量子位

发表于量子位

软件SU | 软件简介及基本设置

小吴懂点室...

发表于CAD室内...