

pyppeteer- 配置及常见问题

作者: [AlwaysBeFriday](#)

原文链接: <https://ld246.com/article/1575602772243>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

官方文档

- <https://miyakogi.github.io/pyppeteer/reference.html>

读取 option

```
# options是dict类型, 初始化时使用  
await launch(self.options)
```

options 选项

```
# 是否忽略https错误, 默认为False  
ignoreHTTPSErrors
```

```
# 将pyppeteer操作减慢指定的毫秒数  
slowMo(int | float)
```

```
# 是否以无头模式运行浏览器。除非appMode或devtools选项为True, 默认为 True  
headless
```

```
# 运行而不是默认捆绑的Chromium的Chromium或Chrome可执行文件的路径  
executablePath
```

```
# 将pyppeteer操作减慢指定的毫秒数  
slowMo (int | float)
```

```
# 是否将浏览器进程stdout和stderr传递到process.stdout和中process.stderr,默认为False  
dumpio
```

```
# 脚本完成后自动关闭浏览器进程, 默认为True  
autoClose
```

```
# 传递给浏览器进程的其他参数 (标志)
```

```
args  
'args': [  
    '--no-sandbox',  
    f'--window-size={width},{height}' #设置浏览器窗口大小, 保持和页面大小一致  
    --proxy-server=127.0.0.1:1080 # 使用代理  
]
```

```
# 是否为每个选项卡自动打开DevTools面板。如果是此选项True, headless则将设置该选项 False  
devtools (布尔)
```

```
# Path to a user data directory, 如果需要多个浏览器并行, 则使用不同的data地址  
userDataDir
```

option 实例

```
self.options = {  
    "headless": False, # 设置模式, 默认无头  
    'dumpio': True,
```

```
'autoClose': False,
'args': [arg2, arg4, arg5, arg6],
"devtools": True,
'userDataDir': os.getcwd() + os.sep + 'userData' + os.sep + username,
}
```

页面大小

```
import tkinter

tk = tkinter.Tk()
width = tk.winfo_screenwidth()
height = tk.winfo_screenheight()
tk.quit()

width, height # 是页面得宽和高
await self.page.setViewport(viewport={'width': width, 'height': height})

#设置浏览器页面得高度和宽度
#设置后可能会出现页面没有铺满得情况，需要调整浏览器大小
```

问题

1. Most likely the page has been closed

1. 在 pyppeteer 中得 connection.py 文件中得 44 行得参数中添加参数，可解决页面自动关闭，开连接问题

```
self._ws = websockets.client.connect
ping_interval = None, ping_timeout = None
```

2. 被检测到驱动浏览器的问题

1. 在chrome的console中输入：window.navigator.webdriver，隐藏的浏览器该值为 undefine，没有隐藏的浏览器该值为 true

```
window.navigator.webdriver = true # 被识别
```

```
window.navigator.webdriver = undefined# # 未被识别
```

2. 隐藏浏览器标识，0.0.25版本pyppeteer 中，修改 launch 中的源码，注释 AUTOMATION_ARGS 中的 --enable-automation

```
AUTOMATION_ARGS = [
    # '--enable-automation',
    '--password-store=basic',
    '--use-mock-keychain',
]
```

3. 设置函数参数，避免被检测

```
async def PageEvaluate(self, page):
    await page.evaluate(""" =>{ Object.defineProperties(navigator,{ webdriver:{ get: () => u
defined } }) }""")
    await page.evaluate(""" =>{ window.navigator.chrome = { runtime: {}, }; }""")
    await page.evaluate(""" =>{ Object.defineProperty(navigator, 'languages', { get: () => ['e
-US', 'en'] }); }""")
    await page.evaluate(""" =>{ Object.defineProperty(navigator, 'plugins', { get: () => [1, 2, 3
4, 5,6], }); }""")
    await self.PageEvaluate(self.page)

await self.PageEvaluate(self.page)
```