

# 北邮自动请假小程序

By Lambert Lang

前言：为应用之前学的 Python 和了解一下 selenium，结合目前的防疫操作，整一个出校请假小程序，目前是雏形初具，可以实现简易的打卡，后续完善的点在第三部分。仅供参考。

## 一.基于 selenium 自动化测试

- 1.安装 Python, pip, 安装 selenium 包  
此处可能会遇到未知原因的错误，请关掉代理即可。
- 2.Chromedriver 与浏览器版本号一致
- 3.使用 selenium+pytest 框架
- 4.使用远程主机定时打卡或者本地 Windows 计划任务定时

每天 7: 50 自动执行一次脚本

```
50 7 * * * /usr/bin/python3 /home/lxy/auto.py
```

## 二.问题及解决

序号	遇到的问题	对应解决方法
1	下拉框元素定位问题	输入元素搜索，点击第一个元素确认
2	时间控件元素难以输入	删除只读属性，直接 js 格式化输入
3	弹窗的确定按钮无法定位	被遮挡定位不到，使用键盘 ENTER
4	诡异元素无法定位	切换定位方法，键鼠配合操作
5	元素不在页面内	滑动页面至元素出现
6	直接使用框架运行定时程序出错	抛开框架   配置完整的 PyCharm 外部运行环境

## 三.不足之处及未来计划：

- 1.未判断已登录场景
- 2.未对网络连通性进行判断，健壮性一般
- 3.未使用显式等待和隐式等待，单纯使用 sleep 延时，效率较低
- 4.后续可以升级邮件&发送操作日志
- 5.后续可以对程序健壮性进行升级（加入登录判断和网络判断）
- 6.后续加入异常捕获和断言
- 7.后续可以对多个用户账号进行自动打卡请假
- 8.后续可以数据驱动，独立用户账号，进行加密处理

# 自动请假小程序思路

By Lambert Lang

## 前言：

由于学校防疫规定，每次外出都需要同学进行“临时出入校园申请”，最近两个月又一直实习，需要天天外出，所以结合目前学习的 python 基础，发挥人类的惰性思维，写了一个简单的定时执行的自动请假脚本。分享出来，仅供学习交流。

## 一.环境配置

环境配置比较简单，主要注意 Chrome 驱动需要跟浏览器版本匹配：

- 1.安装 python 环境
- 2.安装 pycharm
- 3.安装 selenium 包
- 4.下载安装 ChromeDriver 驱动
- 5.配置环境变量

## 二.代码解释

1	<pre>import time from selenium import webdriver from selenium.webdriver.common.by import By from selenium.webdriver.common.keys import Keys</pre>
	导包
2	<pre>class TestCase6():     def setup_method(self, method):         self.driver = webdriver.Chrome()         self.driver.implicitly_wait(10) # seconds         self.vars = {}      def teardown_method(self, method):         self.driver.quit()      def test_case6(self):         self.driver.get(url)         self.driver.set_window_size(1550, 832)         self.driver.find_element(By.ID, "username").click()         time.sleep(1)         self.driver.find_element(By.ID, "username").send_keys(usr)         self.driver.find_element(By.ID, "password").send_keys(psd)</pre>
	Pytest 框架，设置浏览器启动&关闭和方法的执行

3	<pre> self.driver.find_element(By.CSS_SELECTOR, ".btnsubmit").click() time.sleep(3) self.driver.find_element(By.CSS_SELECTOR, ".header_bot .zl-button").click() time.sleep(2) self.driver.find_element(By.CSS_SELECTOR, ".el-dialog__footer:nth-child(3) .zl-line").click() time.sleep(2) self.driver.find_element(By.CSS_SELECTOR, "tr:nth-child(3) &gt; td:nth-child(4) .dvalue").click() self.driver.find_element(By.CSS_SELECTOR, "tr:nth-child(3) &gt; td:nth-child(4) .dvalue").send_keys(     "17862954087") self.driver.find_element(By.CSS_SELECTOR, ".el-select__input").click() self.driver.find_element(By.CSS_SELECTOR, ".el-select__input").send_keys("西土城") time.sleep(2) self.driver.find_element(By.CSS_SELECTOR, ".el-select-dropdown__item &gt; span").c </pre>
	结合使用多种定位方式+js 命令对表单进行自动填充

### 三.任务定时

1.使用 Windows: 任务计划程序, 创建基本任务, 设定时间, 输入要启动的程序和参数即可。

2.Linux 下执行 Windows 下编辑的脚本可能会报错:

- 1) 在 windows 上, 用 UltraEdit 或 EditPlus 等工具先将脚本编码转换, 再放到 Linux 中执行。转换方式如下 (UltraEdit): File-->Conversions-->DOS->UNIX 即可或者是命令: dos2unix test.py
- 2) 在 Linux 上直接替换把不可见字符去掉 sed -i 's/^M//g' filename (注意^M 在 Linux 下写法 按^M 是回车换行符,输入方法是按住 CTRL+v,松开 v,按 m)
- 3) 在 Linux 上转换文件的格式 vim filename :set ff可以查看文件格式会显示 fileformat=mac dos 或者 Unix 等等 直接 :set ff=unix :wq 保存退出即可

附录:

Pytest 官方文档: <https://docs.pytest.org/en/latest/contributing.html>

Selenium 官网: <https://www.selenium.dev/>