



链滴

阿里云监控自定义大盘自动切换

作者: [fish2018](#)

原文链接: <https://ld246.com/article/1560923452166>

来源网站: [链滴](#)

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



项目地址: https://github.com/fish2018/aliyun_monitor_switch.git

功能

阿里云监控大盘自动切换页面, 配合监控大屏(电视、显示器)使用。通过selenium自动登录阿里云监控, 批量打开监控大盘页面, 自动切换chrome标签

使用说明

使用阿里云-RAM用户登录 <https://signin.aliyun.com/login.htm> 监控子账号分配只读权限

- URLS:

自定义监控大盘的url

- SWITCHTIME:

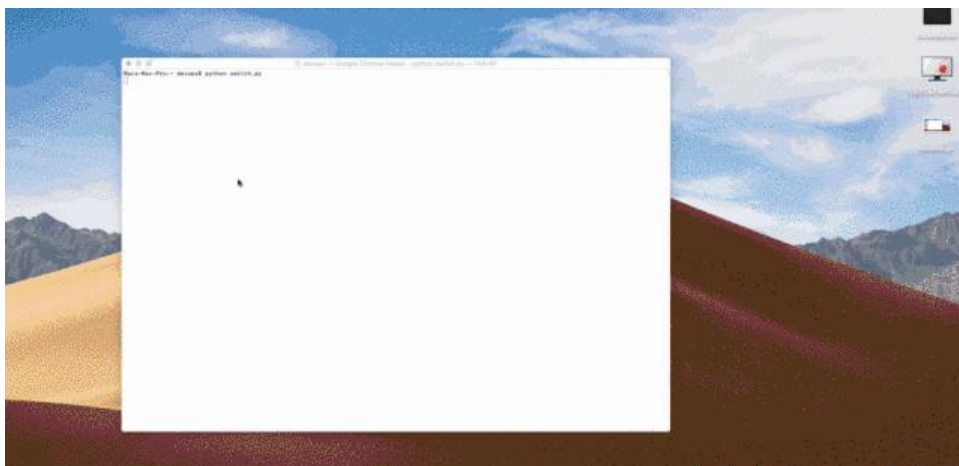
切换标签的间隔时间

- USERNAME:

用户名

- PASSWORD:

密码



开发

mac下需要安装chrome的驱动chromedriver

否则报错:

WebDriverException: 'chromedriver' executable needs to be in PATH

下载对应版本的驱动, 加入到path即可

```
$ sudo mkdir /devops
```

```
$ sudo cp chromedriver /devops
```

```
$ sudo vi /etc/path  
/devops
```

核心代码

```
# -*- coding: utf-8 -*-  
from selenium import webdriver  
import time  
from retry import retry
```

```
# 阿里云监控大盘urls  
URLS = ["URL1", "URL2", "URL3"]  
# 用户名  
USERNAME = "USERNAME"  
# 密码  
PASSWORD = "PASSWORD"  
# 切换间隔时间  
SWITCHTIME = 120
```

```
class Monitor(object):  
    def __init__(self):  
        self.urls = URLS  
        self.switchtime = SWITCHTIME  
    # 加启动配置,隐藏"Chrome正在受到自动软件的控制"  
    option = webdriver.ChromeOptions()  
    option.add_argument('disable-infobars')
```

```

self.browser = webdriver.Chrome(chrome_options=option)
self.username = USERNAME
self.password = PASSWORD

# 子账号登录阿里云监控
def login(self):
    self.browser.get("https://signin.aliyun.com/login.htm")
    # 通过css_selector找到账号, 密码的输入框, 并输入相应的账号/密码
    self.browser.find_element_by_id("user_principal_name").send_keys(self.username)
    self.browser.find_element_by_class_name("next-btn").click()
    time.sleep(1)
    self.browser.find_element_by_id("password_ims").send_keys(self.password)
    self.browser.find_element_by_class_name("submit-btn").click()

# 打开监控大盘页面
def openurls(self):
    for index, u in enumerate(self.urls):
        if index is 0:
            self.browser.get(u)
        else:
            url = "window.open('%s');" % u
            self.browser.execute_script(url)

# 关闭浮动框
@retry(tries=10, delay=1)
def closebox(self):
    try:
        self.browser.find_element_by_css_selector("div.help-guide-step-header > i.topbar-sid
bar-no").click()
    except Exception:
        pass

# 切换全屏
@retry(tries=10, delay=1)
def fullscreen(self):
    try:
        self.browser.find_element_by_class_name("cms4service-hidden").click()
    except Exception:
        pass

# 循环切换浏览器标签
def loopswitch(self):
    handles = self.browser.window_handles # 获取当前窗口句柄集合(列表类型)
    i = 0
    while True:
        for h in handles:
            self.browser.switch_to.window(h)
        if i is 0:
            self.closebox()
            self.fullscreen()
            time.sleep(self.switchtime if self.switchtime >= 30 else 30)
        i += 1

# 启动

```

```
def start(self):
    self.login()
    self.openurls()
    self.loopswitch()

if __name__ == '__main__':
    Monitor().start()
```