模拟12306登录

首先, 我们通过selenium打开12306的登录页面. 并进入到账号登录模块

```python
web = Chrome()

web.get("https://kyfw.12306.cn/otn/resources/login.html")

time.sleep(1)
web.find_element_by_xpath('/html/body/div[2]/div[2]/ul/li[2]/a').click()
time.sleep(1)
```

找到验证码位置, 并截图, 对验证码进行解析.

```python
verify_img = web.find_element_by_xpath('//*[@id="J-loginImg"]')
result = chaojiying.PostPic(verify_img.screenshot_as_png, 9004)
```

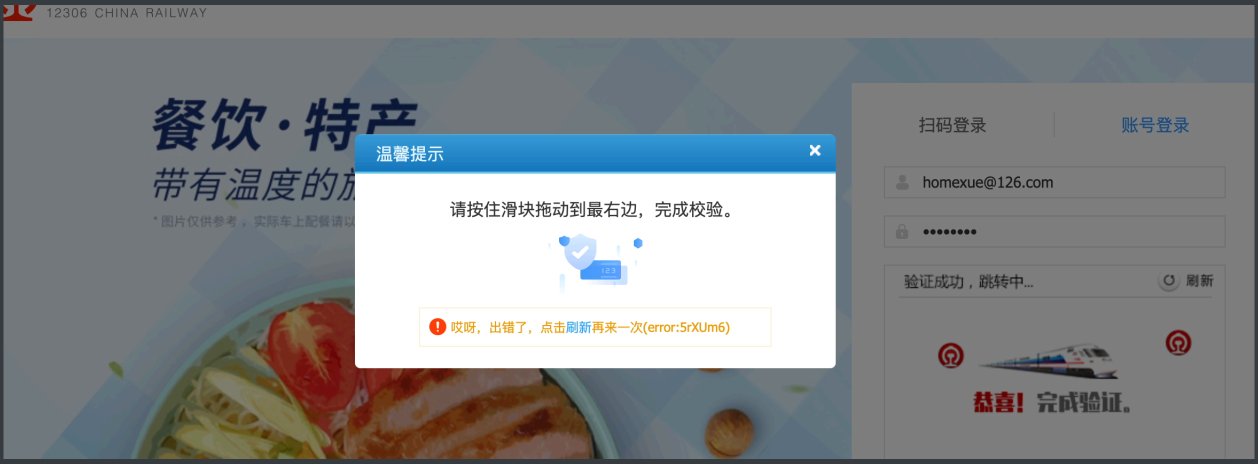
在验证码区域模拟用户点击

```python
points_str_lst = result['pic_str'].split("|")
for p in points_str_lst:
    p_temp = p.split(",")
    p_x = int(p_temp[0])
    p_y = int(p_temp[1])
    
 ActionChains(web).move_to_element_with_offset(verify_img, p_x, p_y).click().perform()
```

然后输入用户名和密码, 并点击登录按钮

```python
time.sleep(1)
web.find_element_by_xpath('//*[@id="J-userName"]').send_keys("homexue@126.com")
web.find_element_by_xpath('//*[@id="J-password"]').send_keys('q6035945')
time.sleep(1)
web.find_element_by_xpath('//*[@id="J-login"]').click()
```

最后一步, 进入滑块环节, 这里其实看似容易, 操作起来就恶心了. 因为这里才是本节真正要讲的. 12306会在这里进行浏览器验证, 验证你是否是通过自动化工具启动的浏览器. 而且, 这个问题非常恶心, 常规的浏览器检测处理方案都是没用的.



请按住滑块拖动到最右边，完成校验。

哎呀，出错了，点击刷新再来一次(error:5rXUm6)

homexue@126.com

••••••••

验证成功，跳转中...

恭喜! 完成验证。

注意, 此时我们要了解一下这个插件的工作机制. 它是在整个页面加载的时候就开始对浏览器参数进行读取. 所以我们常规的对Chrome设置是无效的. 此时, 需要添加以下一段代码来规避检测.

```
# 亲测, 88版本以后可以用.
option = Options()
# option.add_experimental_option('excludeSwitches', ['enable-automation'])
option.add_argument('--disable-blink-features=AutomationControlled')
web = Chrome(options=option)

# 亲测, 88版本之前可以用.
```

```python
# web = Chrome()
#
# web.execute_cdp_cmd("Page.addScriptToEvaluateOnNewDocument", {
#   "source": """
#   navigator.webdriver = undefined
#     Object.defineProperty(navigator, 'webdriver', {
#       get: () => undefined
#     })
#   """
# })
```

最后给出完整代码

```python
from selenium.webdriver import Chrome
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.common.action_chains import ActionChains
from chaojiying import Chaojiying_Client
import time

```

```python

chaojiying = Chaojiying_Client('xxxxxxx', 'xxxxxx', '912488')
# result = chaojiying.PostPic(png_code_img, 9004)


# 亲测，88版本以后可以用.
option = Options()
# option.add_experimental_option('excludeSwitches', ['enable-automation'])
option.add_argument('--disable-blink-features=AutomationControlled')
web = Chrome(options=option)

# 亲测，88版本之前可以用.
# web = Chrome()
#
# web.execute_cdp_cmd("Page.addScriptToEvaluateOnNewDocument", {
#   "source": """
#   navigator.webdriver = undefined
#     Object.defineProperty(navigator, 'webdriver', {
#       get: () => undefined
#     })
#   """
# })

web.get("https://kyfw.12306.cn/otn/resources/login.html")

time.sleep(1)
web.find_element_by_xpath('/html/body/div[2]/div[2]/ul/li[2]/a').click()
time.sleep(1)

verify_img = web.find_element_by_xpath('//*[@id="J-loginImg"]')
result = chaojiying.PostPic(verify_img.screenshot_as_png, 9004)

points_str_lst = result['pic_str'].split("|")
for p in points_str_lst:
    p_temp = p.split(",")
    p_x = int(p_temp[0])
    p_y = int(p_temp[1])

    ActionChains(web).move_to_element_with_offset(verify_img, p_x, p_y).click().perform()
```

```

time.sleep(1)
web.find_element_by_xpath('//*[@id="J-userName"]').send_keys("xxxxxxx@126.com")
web.find_element_by_xpath('//*[@id="J-password"]').send_keys('xxxxxxx')
time.sleep(1)
web.find_element_by_xpath('//*[@id="J-login"]').click()

time.sleep(3)
btn = web.find_element_by_xpath('//*[@id="nc_1_n1z"]')
ActionChains(web).drag_and_drop_by_offset(btn, 300, 0).perform()


```