


本文主要介绍登陆博客园的流程，使用创建的Base进行实战

1-博客园

登陆URL: <https://account.cnblogs.com/signin>

博客园用户登录



代码改变世界

密码登录

短信登录

登录用户名 / 邮箱

请输入登录用户名或邮箱




密码

请输入密码

☒ 记住我

登录

第三方登录/注册



没有账户, [立即注册](#)

2-登陆过程

查看登陆页面，分析登陆过程如下：

1. 登陆: <https://account.cnblogs.com/signin>
2. 点击: 密码登陆
3. 输入: 用户名
4. 输入: 密码
5. 勾选: 记住我 (可选)
6. 点击: 登陆

7. 进入：我的
8. 用户头像并截图
9. 验证园号=2910916

3-新建用例

新建：test/automation/cases/Login/LoginBlogTest.js

```
const {describe,it,before,after,afterEach}=require('mocha');
const Base=require('../../Base');

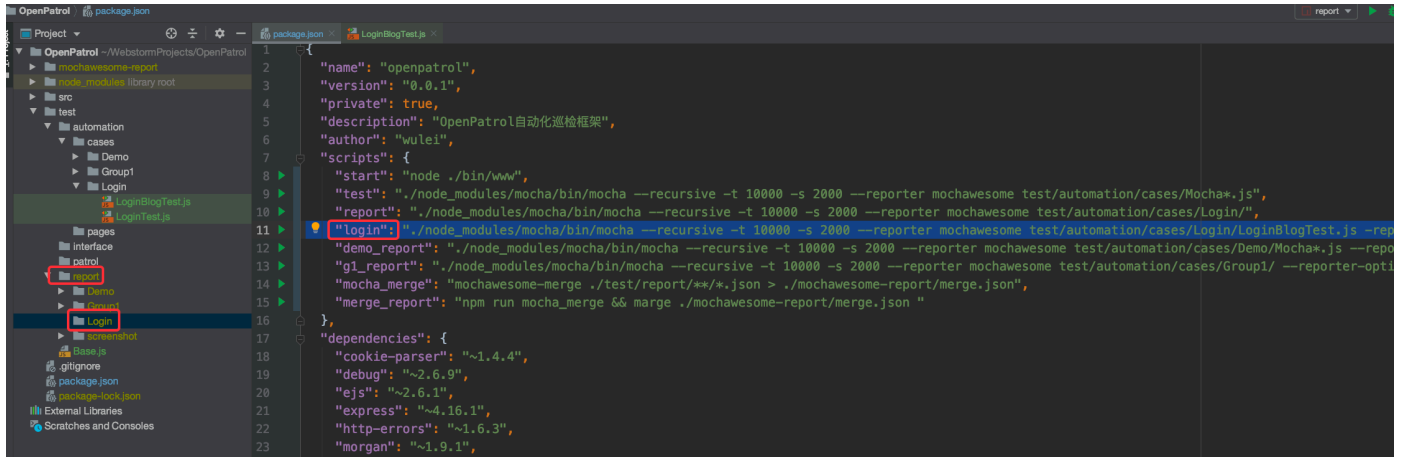
describe('博客园',function () {
  before(Base.before);
  after(Base.after);
  afterEach(Base.afterEach);
  this.timeout(20000);
  it('Case1-登陆', async function () {
    //      1. 登陆: https://account.cnblogs.com/signin
    //      2. 点击: 密码登陆
    //      3. 输入: 用户名
    //      4. 输入: 密码
    //      5. 勾选: 记住我 (可选)
    //      6. 点击: 登陆
    //      7. 进入: 我的
    //      8. 用户头像, 并截图
    //      9. 验证园号=2910916
  })
})
```

4-报告目录和命令

按照需要创建报告目录和命令（非必须）

命令参考：

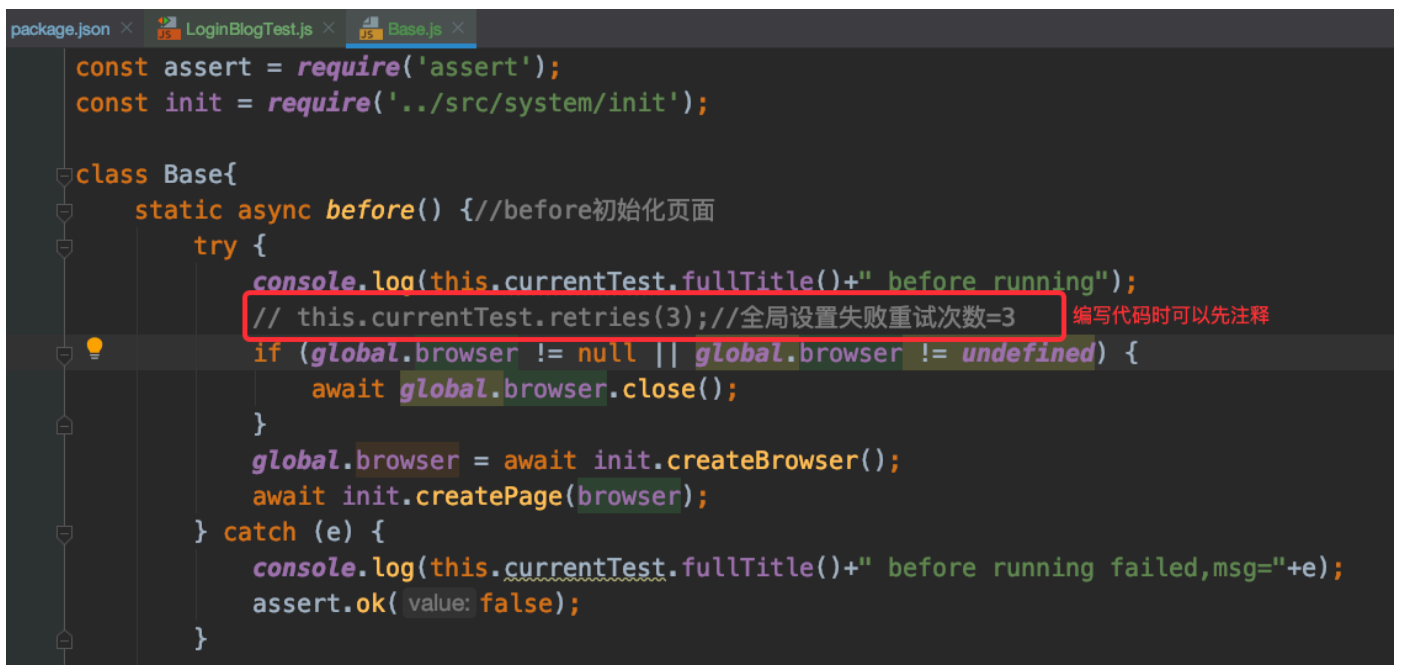
```
"login": "./node_modules/mocha/bin/mocha --recursive -t 10000 -s 2000 --reporter
mochawesome test/automation/cases/Login/LoginBlogTest.js -reporter-options
reportDir=test/report/Login",
```



5-建议

为了提升编写代码的效率:

1-注释掉before失败重试



2-注释after, 这样可以保证窗口先不关闭



6-puppeteer api

常见的方法说明：

方法	说明
page.goto()	打开指定的网址
page.waitForSelector()	等待指定的元素出现，并且可以通过then，catch完成后续操作
page.waitForXPath('xpath')	等待xpath 对应的元素出现
page.focus()	聚焦指定的元素
page.type()	输入文本内容
page.tap()	模拟手指触摸点击
page.click()	点击指定的元素
page.hover()	鼠标 hover 元素
page.waitForResponse()	等待响应
page.waitForTimeout()	等待指定时间
page.\$()	单个元素选择，返回元素信息，可以用于元素的操作或者截图保存
page.\$\$()	批量元素选择，返回元素信息，可以用于元素的操作或者截图保存
page.\$x()	Xpath选择器，返回一组元素（同page.\$\$()）
page.\$eval()	查找一个元素并执行自定义方法，多用于元素信息获取
page.\$\$eval()	查找一组元素并执行自定义方法，多用于元素信息获取
page.evaluate()	执行JS方法
page.setCookie()	设置指定cookie
page.cookies()	返回全部cookie信息
page.deleteCookie()	删除指定cookie
page.on()	事件监听，参考下一节

7-完整代码

LoginBlogTest.js

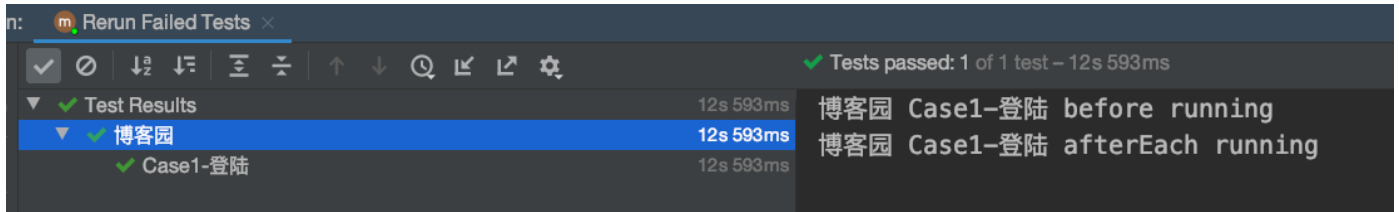
```
const {describe,it,before,after,afterEach}=require('mocha');
const Base=require('../../Base');
const {assert}=require('chai');
```

```

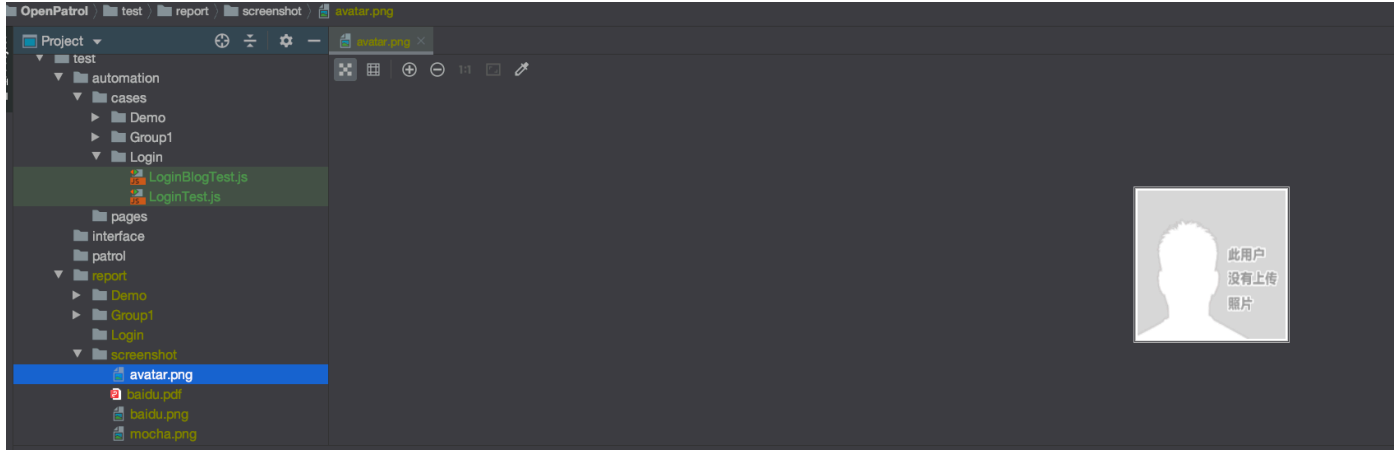
describe('博客园',function () {
  before(Base.before);
  // after(Base.after);
  afterEach(Base.afterEach);
  this.timeout(40000);
  it('Case1-登陆', async function () {
    //1. 登陆: 博客园
    await page.goto("https://account.cnblogs.com/signin");
    //2. 点击: 密码登陆
    await page.waitForSelector('#mat-tab-label-0-0 > div',
{timeout:2*1000}).then(ele=>{ele.click()}).catch(e=>{console.log(e)})
    //3. 输入: 用户名
    await page.focus('#mat-input-0');
    await page.type('#mat-input-0','UI自动化');
    //4. 输入: 密码
    await page.focus('#mat-input-1');
    await page.type('#mat-input-1','520bokeyuan');
    //5. 勾选: 记住我 (可选)
    // await page.click('#mat-checkbox-1 > label > span.mat-checkbox-inner-
container');
    //6. 点击: 登陆
    await page.waitForSelector('body > app-root > app-sign-in-layout > div > div >
app-sign-in > app-content-container > div > div > div > form > div > button',
{timeout:2*1000}).then(ele=>{ele.click()}).catch(e=>{console.log(e)})
    //等待响应返回200
    await page.waitForResponse(response => response.status() === 200);
    await page.waitForTimeout(2000);
    //7. 进入: 我的
    await page.goto("https://home.cnblogs.com/u/2910916");
    await page.waitForTimeout(2000);
    //8. 用户头像, 并截图
    const avatar=await page.$('#user_profile_block > table > tbody > tr > td:nth-
child(1) > div > img');
    await avatar.screenshot({
      path: './test/report/screenshot/avatar.png',
      type: 'png'
    })
    //9. 验证园号=2910916
    const name =await page.$eval('#user_profile > li:nth-child(2) > span:nth-
child(2)',el => el.textContent);
    assert.equal(name,'2910916');
  })
})

```

运行后:



元素截图保存成功



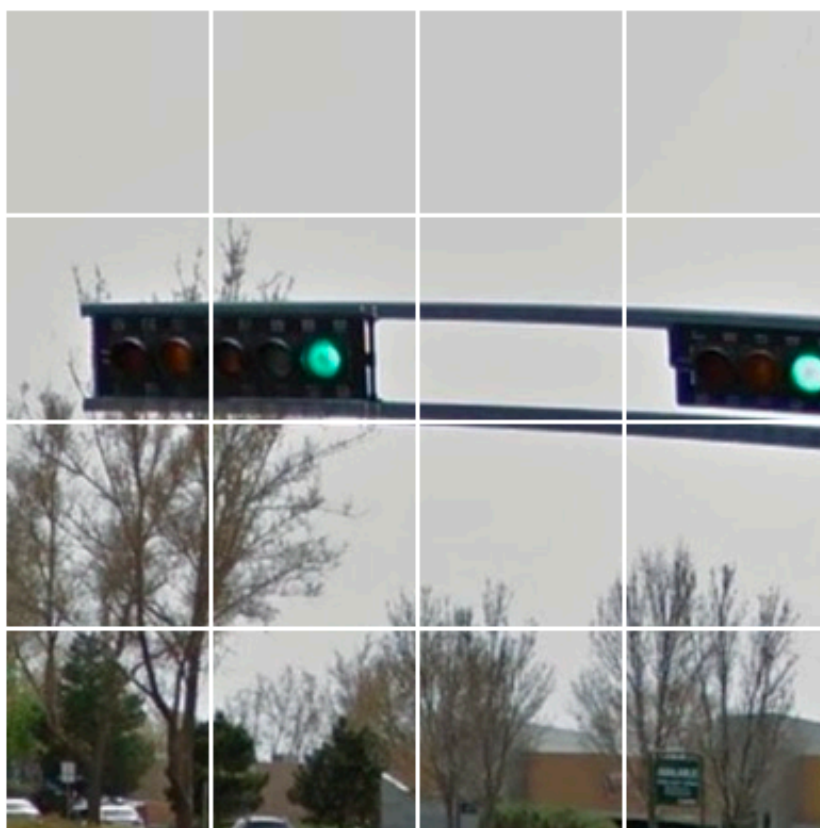
说明：由于博客园的一些安全登录校验，有时候会弹出图片选择框，这个不在我们的讨论范围，所以碰到的时候可以手动选择进行跳过。对于常见登陆校验跳过的处理方式：

- 1-万能验证码（推荐）
- 2-白名单用户（推荐）
- 3-图片识别（难度大，对于UI测试不是必须）

请选择包含

红绿灯

的所有图块



跳过