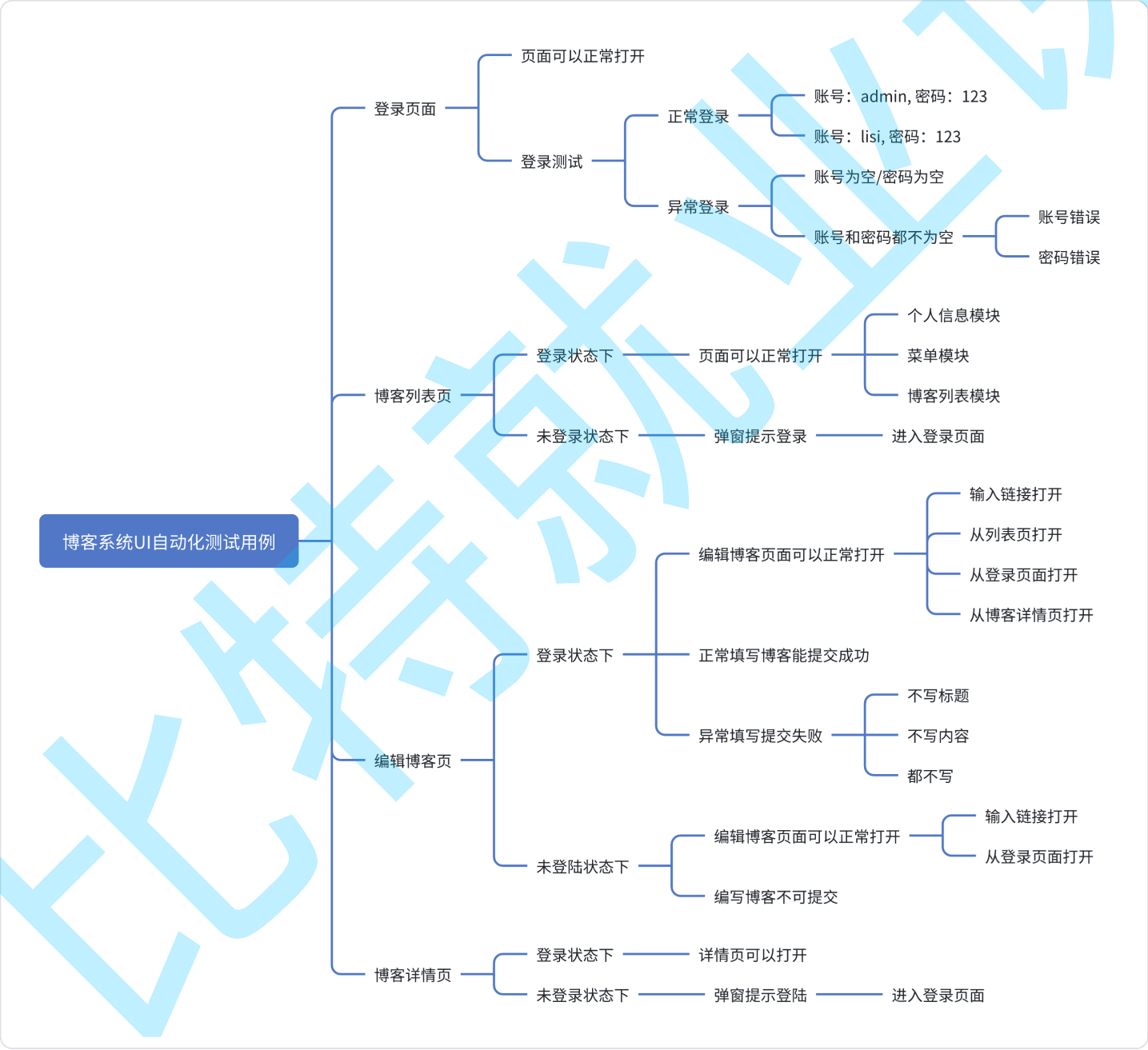


第八章节 自动化测试实战篇（C++方向）

以博客系统为例，基于web项目进行完整的自动化实战。

1. 自动化实施步骤

1.1 编写web测试用例



1.2 自动化测试脚本开发

common/Utils.py

```
1
2 #创建驱动对象
3 import datetime
4 import os.path
5 import sys
6
7 from selenium import webdriver
8 from selenium.webdriver.chrome.service import Service
9 from webdriver_manager.chrome import ChromeDriverManager
10
11
12 class Driver:
13     driver = ""
14     def __init__(self):
15         options = webdriver.ChromeOptions()
16         # options.page_load_strategy = 'eager'
17         self.driver =
18         webdriver.Chrome(service=Service(ChromeDriverManager().install()),options=options)
19
20     def getScreenShot(self):
21         dirname = datetime.datetime.now().strftime('%Y-%m-%d')
22         if not os.path.exists("../images/" + dirname):
23             os.mkdir("../images/" + dirname)
24         filename = sys._getframe().f_back.f_code.co_name + "-" +
25             datetime.datetime.now().strftime(
26                 '%Y-%m-%d-%H%M%S') + ".png"
27         self.driver.save_screenshot(f'../images/{dirname}/' + filename)
28
29 BlogDriver = Driver()
```

cases/BlogLogin.py

```
1 import time
2
3 from selenium.webdriver.common.by import By
4
5 from common.Utils import BlogDriver
6
7 class BlogLogin:
8     url = ""
9     driver = ""
10
```

```

11     def __init__(self):
12         self.url = "http://192.168.47.135:8653/blog_system/blog_login.html"
13         self.driver = BlogDriver.driver
14         self.driver.get(self.url)
15     def loginSucTest(self):
16         time.sleep(2)
17         self.driver.find_element(By.CSS_SELECTOR, "#username").clear()
18         self.driver.find_element(By.CSS_SELECTOR, "#password").clear()
19         self.driver.find_element(By.CSS_SELECTOR,
20             "#username").send_keys("admin")
21         self.driver.find_element(By.CSS_SELECTOR, "#password").send_keys("123")
22         self.driver.find_element(By.CSS_SELECTOR, "#submit").click()
23         # 对登录结果进行检测, 如果跳转到了博客列表页才算是登录成功了
24         self.driver.find_element(By.CSS_SELECTOR, "body > div.container >
25             div.left > div > img")
26         BlogDriver.getScreenShot()
27         self.driver.back()
28
29     def loginFailTest(self):
30         self.driver.find_element(By.CSS_SELECTOR, "#username").clear()
31         self.driver.find_element(By.CSS_SELECTOR, "#password").clear()
32         self.driver.find_element(By.CSS_SELECTOR,
33             "#username").send_keys("admin")
34         self.driver.find_element(By.CSS_SELECTOR, "#password").send_keys("111")
35         self.driver.find_element(By.CSS_SELECTOR, "#submit").click()
36         # 等登录失败的结果进行检测, 不能仅仅通过body来校验结果, 因为登录成功的结果页也有
37         # body元素
38         expect = "用户名或密码错误!"
39         actual = self.driver.find_element(By.CSS_SELECTOR, "body").text
40         print(actual)
41         BlogDriver.getScreenShot()
42         assert expect == actual
43         self.driver.back()

```

cases/BlogList.py

```

1 from selenium.webdriver.common.by import By
2
3 from common.Utils import BlogDriver
4 class BlogList:
5     url = ""
6     driver = ""
7     def __init__(self):
8         self.driver = BlogDriver.driver
9         self.url = "http://192.168.47.135:8653/blog_system/blog_list.html"

```

```

10         self.driver.get(self.url)
11
12         # 前提：登陆状态下才能进入到列表页
13         def ListTest(self):
14             self.driver.find_element(By.CSS_SELECTOR, "body > div.container >
div.left > div > img")
15             self.driver.find_element(By.CSS_SELECTOR, "body > div.container >
div.right > div:nth-child(1) > a")
16
17             articles = self.driver.find_elements(By.CSS_SELECTOR, "body >
div.container > div.right > div")
18             assert len(articles) > 10
19
20             #点击文章
21             self.driver.find_element(By.CSS_SELECTOR, "body > div.container >
div.right > div:nth-child(1) > a").click()
22
23             title = self.driver.title
24             assert title == "博客详情页"
25             BlogDriver.getScreenShot()

```

cases/BlogDetail.py

```

1  from selenium.webdriver.common.by import By
2
3  from common.Utills import BlogDriver
4  class BlogDetail:
5      url = " "
6      driver = ""
7
8      def __init__(self):
9          self.url = "http://192.168.47.135:8653/blog_system/blog_detail.html?
blogId=15"
10         self.driver = BlogDriver.driver
11         title = self.driver.title
12         #列表页已经跳过来了，无需再指定url跳转
13         if not title=="博客列表页":
14             self.driver.get(self.url)
15         def DetailCheck(self):
16             #body > div.container > div.right > div:nth-child(1) > a
17             BlogDriver.getScreenShot()
18             self.driver.find_element(By.CSS_SELECTOR, "body > div.container >
div.right > div > h3")
19             self.driver.find_element(By.CSS_SELECTOR, "body > div.container >
div.right > div > div.date")

```

cases/RunCases.py

```
1 from common.Utils import BlogDriver
2 from tests import BlogLogin
3 from tests import BlogList
4 from tests import BlogDetail
5 if __name__ == "__main__":
6     BlogLogin.BlogLogin().loginSucTest()
7     BlogList.BlogList().ListTest()
8     BlogDetail.BlogDetail().DetailCheck()
9     BlogDriver.driver.quit()
```

1.3 将自动化测试补充至测试报告

测试报告模板：

【测试报告】博客系统

| | | | |
|--------|-------------------|-------|-----|
| 项目名称 | 博客系统 | 版本号 | / |
| 发布类型 | 分级发布 | 测试负责人 | 马绵 |
| 测试完成日期 | 3.1 | 联系方式 | |
| 评审人 | 王一博、肖战、马绵、李产品、王交互 | 批准人 | 李产品 |
| 评审日期 | | 批准日期 | |

- 1、项目背景
- 1.1 测试目标及测试任务概括
- 1.2 被测的系统、代码包及其文档等信息
- diff链接: [xxxxx](#)
- 接口: [xxxx](#)
- 《测试计划》
- 1.3 产品需求和设计文档等
- 《需求文档》、《技术文档》、《设计图》
- 2.测试安排

| 模块 | 子模块 | 前端 | 开发 | 提测时间 | 测试 | 工时 | 排期 | 进度 | 备注 |
|----|---------------|----|-----|------|----|------|------|------|----|
| 登陆 | 登陆功能 | 肖战 | 王一博 | 2.30 | 马绵 | 0.5d | 2.31 | 测试完成 | |
| 主页 | 博主信息、 博客列表 | 肖战 | 王一博 | 2.30 | 马绵 | 0.5d | 2.31 | 测试完成 | |

| | | | | | | | | | |
|-------|---------|----|-----|------|----|------|-----|------|--|
| | 火、守阳仁 | | | | | | | | |
| 写博客 | 博客编辑与发布 | 肖战 | 王一博 | 2.31 | 马绵 | 1d | 3.1 | 测试完成 | |
| 博客详情页 | 博客详情页 | 肖战 | 王一博 | 2.31 | 马绵 | 0.5d | 3.2 | 测试完成 | |

3、测试分类

1) 功能测试

- 测试用例

功能测试结果：测试用例100%通过

2) 自动化测试

- 自动化测试覆盖模块：登陆页面、首页、博客编辑页面、博客详情页面等等
- 自动化代码示例（可以是代码链接，也可以直接放代码）

<https://gitee.com/mamianer/classcode/blob/master/allcode/java104&105/BlogAutoTest>

- 自动化测试用例数量：20
- 自动化测试结果：pass: 19/20、fail: 1/20
- 自动化测试问题是否修复：是

4.遗留风险

测试时间紧张，先保证博客系统主功能没有问题，可能存在细节测试不到位等风险。

5.测试结果评估

- 1) 主功能测试通过，项目可以上线。
- 2) 项目上线后高优观察线上数据，查看线上用户操作日志，及时跟进用户反馈。