

实验 6 – WEB 应用程序测试

学号: 2017141463145 姓名: 龙行超

分数:

1 引言 (3')

自动化测试是在手工测试之后进行的,为了解决回归测试、压力测试和兼容性测试的问题,所属为黑盒测试(功能测试)、灰盒测试(接口测试)、白盒测试(单元测试),本实验的 web 自动化测试属于黑盒测试(功能测试),其大致可以分为四类:Web- (UI) 自动化测试、接口自动化测试、移动 (app) 自动化测试、单元测试自动化测试。

Web 项目自动化测试是指程序代替人去自动验证 Web 项目功能的过程,由于 Web 项目需求变动不频繁、周期长,需要回归测试的原因,所以非常适合自动化测试。本实验使用 selenium IDE 的浏览器插件,进行自动化的 web 测试。

selenium 插件采用录制-回放的方式来实现自动化测试,使用过程中,其记录 Web 页面上的每一个操作,产生自动化测试的脚本,执行该脚本,便能对 Web 页面实现自动化的测试。

2 测试脚本设计(35', 每个脚本 7')

测试网站为: <https://zh.numberempire.com/>, 这是一个在线的数学计算网站,其中能计算表达式、函数求导、阶乘计算、矩阵计算、因式分解等等,现就其中五个功能,写五个测试脚本。

2.1 测试脚本 1:

脚本名: ExpressionEvaluation

测试需求: 测试表达式的计算能否得出正确的结果

测试脚本描述:

1. 打开网站
2. 选择表达式计算
3. 选择输出结果
4. 输入表达式
5. 计算结果

Selenium IDE - WebCaculate*

Project: WebCaculate*

Tests + [Run] [Repeat] [Clock] [Pencil] [Pause] [REC]

Search tests... Run current test Ctrl+R .com/

ExpressionEvaluation*
FactorialCalculation
FunctionDerivation*

	Command	Target	Value
7	click	name=result_type	
8	click	name=result_type	
9	click	css=tr:nth-child(5) input:nth-child(3)	
10	verify text	css=.result	表达式 5*3+5 = 20
11	verify text	id=MathJax-Span-5	20

Command

Target

Value

Description

Log Reference

7. click on name=result_type OK 17:16:11

8. click on name=result_type OK 17:16:11

9. click on css=tr:nth-child(5) input:nth-child(3) OK 17:16:11

10. verifyText on css=.result with value 表达式 5*3+5 = 20 OK 17:16:12

11. verifyText on id=MathJax-Span-5 with value 20 OK 17:16:15

'ExpressionEvaluation' completed successfully 17:16:15

测试脚本执行结果截图：

测试（回放）结果：通过

2.2 测试脚本 2：

脚本名：FactorialCalculation

测试需求：测试阶乘的计算能否得出正确的结果

测试脚本描述：

1. 打开网站
2. 选择阶乘计算
3. 输入非负数
4. 计算结果

测试脚本执行结果截图：

Selenium IDE - WebCaculate*

Project: WebCaculate*

Executing

FactorialCalculation*

https://zh.numberempire.com/

	Command	Target	Value
7	click	css=tr:nth-child(5) form > input:nth-child(2)	
8	verify text	css=tr:nth-child(7) div:nth-child(1)	0的阶乘为:
9	verify text	linkText=1	1
10	click	linkText=1	
11	verify title	该数性质 1	

Command

Target

Value

Description

Runs: 1 Failures: 0

Log

Reference

7. click on css=tr:nth-child(5) form > input:nth-child(2) OK 17:22:00

8. verifyText on css=tr:nth-child(7) div:nth-child(1) with value 0的阶乘为: OK 17:22:00

9. verifyText on linkText=1 with value 1 OK 17:22:01

10. click on linkText=1 OK 17:22:02

11. verifyTitle on 该数性质 1 OK 17:22:02

'FactorialCalculation' completed successfully 17:22:03

测试（回放）结果：通过

2.3 测试脚本 3:

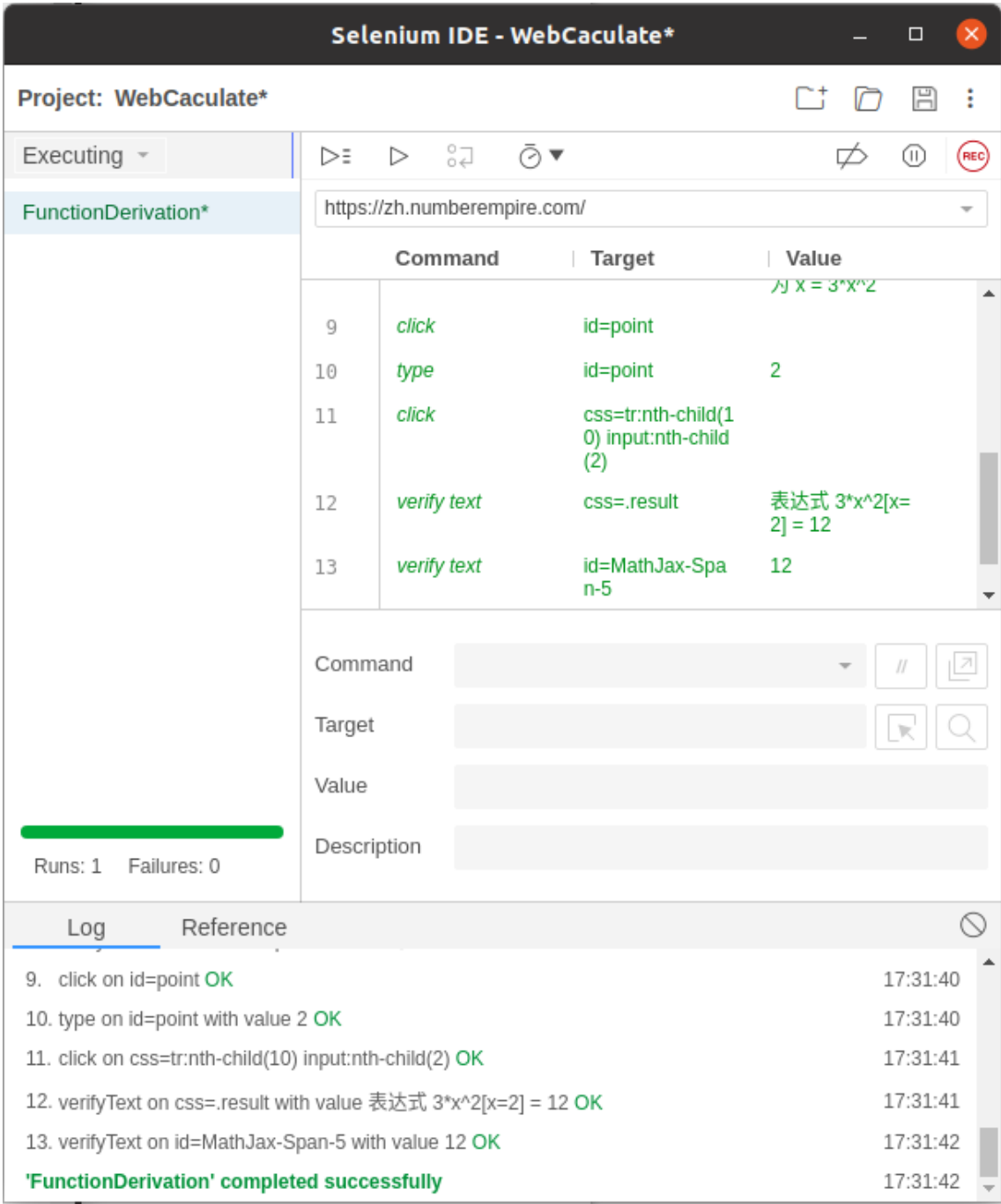
脚本名：FunctionDerivation

测试需求：测试函数求导能否得出正确的结果

测试脚本描述：

1. 打开网站
2. 选择函数求导
3. 输入函数
4. 计算
5. 输入x的具体值
6. 计算

测试脚本执行结果截图：



测试（回放）结果：通过

2.4 测试脚本 4:

脚本名: DefiniteIntegral

测试需求: 测试定积分计算能否得出正确的结果

测试脚本描述:

1. 打开网站
2. 选择定积分计算

3. 输入函数
4. 输入下限
5. 输入上限
6. 计算

测试脚本执行结果截图：

Selenium IDE - WebCaculate*

Project: WebCaculate*

Executing ▾

DefiniteIntegral*

https://zh.numberempire.com/

	Command	Target	Value
12	type	name=b	3
13	click	css=input:nth-child(6)	
14	mouse over	css=input:nth-child(6)	
15	mouse out	css=input:nth-child(6)	
16	verify text	css=.result	对此函数求积分 x^2 自变量为 x 区间[2,3] = 6.333...

Command

Target

Value

Description

Runs: 1 Failures: 0

Log

Reference

12. type on name=b with value 3 OK 17:37:16

13. click on css=input:nth-child(6) OK 17:37:17

14. mouseOver on css=input:nth-child(6) OK 17:37:17

15. mouseOut on css=input:nth-child(6) OK 17:37:18

16. verifyText on css=.result with value 对此函数求积分 x^2 自变量为 x 区间[2,3] = 6.333333333333334 OK 17:37:18

'DefiniteIntegral' completed successfully 17:37:19

测试（回放）结果：通过

2.5 测试脚本 5:

脚本名：NumberOfCombinations

测试需求：测试组合数计算能否得出正确的结果

测试脚本描述：

1. 打开网站
2. 选择组合数计算
3. 选择条件
4. 输入N
5. 输入M
6. 计算

测试脚本执行结果截图：

Selenium IDE - WebCaculate

Project: WebCaculate

Executing

NumberOfCombinations

https://zh.numberempire.com/

	Command	Target	Value
8	click	id=N	
9	type	id=N	7
10	click	id=M	
11	type	id=M	3
12	click	css=#form1 > input:nth-child(2)	
13	verify text	id=result	$n! / (n - m)! = 210$

Command

Target

Value

Description

Runs: 1

Failures: 0

Log

Reference

9. type on id=N with value 7 OK

10. click on id=M OK

11. type on id=M with value 3 OK

12. click on css=#form1 > input:nth-child(2) OK

13. verifyText on id=result with value $n! / (n - m)! = 210$ OK

'NumberOfCombinations' completed successfully

测试（回放）结果：通过

3 WEB 自动化测试的优缺点(7')

优点:

- 1 相同的时间能运行更多的测试用例。
- 2 自动化的脚本可以多次反复使用。

- 3 减少认为的错误。
- 4 能处理人工不能测试的东西。

缺点：

- 1 需要一定的技术要求。
- 2 不能完全代替人工。
- 3 发现缺陷的能力比手工测试弱。
- 4 测试工具无想象力，需要人使用。
- 5 严重依赖工具。
- 6 有可能制约项目的开发，自动化测试更难维护，从而限制开发。

4 测试时遇到的困难和挑战(3')

- 1 找一个比较好测试的网站来进行实验。
- 2 录制过程中总会进行无关操作，使得反复录制。
- 3 常用的验证与断言命令的理解。

5 意见和建议(2')

首次接触并体验了自动花测试脚本，但是有个建议，此次实验完全抛除了测试用例这个东西，只是需要完成五个测试脚本，那么测试用例到底如何与这个自动化测试工具相关联的其实并没有涉及到，希望增加这部分的内容，使得整个自动化测试的体验更加完整。