



1111111

[三] 哀云 (五十七期·下)
Cypress测试神器—小日试用笔记01
测试女巫紧跟时代脉搏之大数据分析系列(2)20
带你走进Jmeter参数化41
一文详尽Jmeter对数据库批量增删改查47
疫情之下,IT从业者都做了些啥?61
做了这么多年的功能测试,原来我与别人差在这67
全栈测试:平衡单元测试与端到端测试70
无脚本测试自动化框架:工具及示例
Appium自动化测试遇到的chromedriver/chrome坑



每次不重样,教你收获最新测试技术! 《 微信扫一扫关注我们





Cypress 测试神器—小白试用笔记

作者: 瑾沐沐

沐沐已经很久没有写文章了,原因有三个:要么"懒",要么"忙",要么"懒"。 但是接触到 Cypress 后,真的非常想给大家安利这个 UI 自动化测试神器了。以前也接触 了一些测试框架,但是 UI 自动化测试的投入产出比始终较低。我一直在思考 UI 自动化 有用吗?直到遇到 Cypress 后我明白了,有人用就有用,没人用就没用,UI 自动化脚本 往往投入的时间成本比较高,如果使用者仅仅是测试人员,那么产生的效益就会很低, 如果使用者是整个产品部门,那么价值就会显现出来。而传统的自动化框架,安装较为 繁琐,即使我们写好脚本,研发同学也不愿意去安装使用。而 Cypress 是基于 node.js 环 境的,安装则极为便利,如果我们测试人员维护好了自动化脚本,就可以直接将 js 脚本 分享给研发人员,辅助研发人员进行自测。如果能将 UI 自动化普及到整个部门都去使 用,辅助提升产品研发效率,那 UI 自动化就是非常有价值的事情。

第一章、框架介绍

1.1、Cypress 简介

官网地址: https://docs.cypress.io

官网的首页有一句标语 "The web has evolved. Finally, testing has too."

意思是 Web 已经进化了,最终,测试也是。而对于 Cypress 的介绍则是"Fast, easy and reliable testing for anything that runs in a browser."翻译为"对浏览器中运行的任何内容进行快速、简单和可靠的测试"。Cypress 是一种 E2E 测试框架,即 end to end 测试。同类型的框架还有 Testcafe 等。但是 E2E 测试似乎是比较冷门的模块,而 Cypress 使得 E2E 测试更加简单。Cypress 框架安装成功后, example 文件中还有很多的 demo,有助于我们更快的了解和学习该框架。Cypress 也可以说是 web UI 自动化的完美解决方案,



下文中将简单描述 Cypress 的优势。

1.2、Cypress 优势

(1) 安装简单

毫不夸张的说,在网络正常的情况下,10分钟就可以安装好 Cypress 框架,一天则可以入门,写一些简单的用例。

(2)运行速度快

相较于 Selenium 来说, Cypress 的运行速度真的是相当快了, 它不需要 web driver 来驱动浏览器。

(3) 脚本实时调试

在脚本编写过程中,只要编辑器中进行保存,脚本就会自动运行,可以快速高效 的知道脚本是否正确。

(4) 兼容性测试

目前 Cypress 已经支持 Chrome 和 Firefox 两款主流浏览器,在 GUI 界面可以非常便捷的切换浏览器进行兼容性测试。

(5) 失败用例自动保存截图

脚本运行失败后,会在..\cypress\screenshots 文件夹下面自动保存失败用例的截图, 方便我们追踪道失败用例的原因。

(6)运行过程自动录制视频

脚本运行完成后,会在..\cypress\videos 文件夹下自动保存脚本运行过程录制的视频。

(7)人人可用

沐沐觉得这个框架最大的好处就是"人人可用", Cypress 不但可以帮助测试人员进行版本回归测试,浏览器兼容性等测试,还可以辅助研发人员进行自测,甚至可以辅助 去造一些测试数据,只要是 web 端手工不断重复的工作,都可以用 Cypress 辅助完成。

第二章、环境搭建





备注: 以下安装步骤是基于 Windows10。

2.1、安装 node.js

- 1、官网下载地址: https://nodejs.org/en/download/;
- 2、傻瓜式安装,安装后查看是否安装成功: node-v;
- 3、最新版的 node 在安装时同时也安装了 npm, 查看 npm 版本: npm -v;

C:\Windows\system32\cmd.exe	_		×
Microsoft Windows [版本 10.0.17763.1098] (c) 2018 Microsoft Corporation。保留所有权利。			^
C:\Users\shero <mark>⊳node -v</mark> v8.16.2			
C:\Users\shero <mark>}npm -v</mark> 6.4.1			
C:\Users\shero>			
	博为峰旗下		
	51 test	Un	g
			<u>()</u>

具体安装步骤参考菜鸟教程:

https://www.runoob.com/nodejs/nodejs-install-setup.html

2.2、安装 Cypress

- 1、cmd 进入命令行模式,进入D 盘创建一个文件夹: mkdir Crypress_project;
- 3、进入新创建的文件夹下面: cd Crypress_project;
- 4、初始化项目: npm init -y;
- 5、安装 cypress: npm i cypress -S -D; 安装成功截图如下:





a npm	-		×
D:\Crypress_project>npm i cypress −S −D npm <mark>WARN</mark> deprecated request@2.88.0: request has been deprecated, see https://github.com/request/request/i	ssues/3	3142	^
<pre>> cypress@4.1.0 postinstall D:\Crypress_project\node_modules\cypress > node index.jsexec install</pre>			
Installing Cypress (version: 4.1.0)			
 ✓ Downloaded Cypress ✓ Unzipped Cypress ✓ Finished Installation C:\Users\shero\AppData\Local\Cypress\Cache\4.1.0 			
You can now open Cypress by running: node_modules\.bin\cypress open			
https://on.cypress.io/installing-cypress			
npm WARN saveError ENOENT: no such file or directory, open 'D:\Crypress_project\package.json' npm notice created a lockfile as package-lock.json. You should commit this file. npm WARN encent ENOENT: no such file or directory, open 'D:\Crypress_project\package.json' npm WARN Crypress_project No description npm WARN Crypress_project No repository field. npm WARN Crypress_project No README data npm WARN Crypress_project No license field.			
+ cypress@4.1.0 added 201 packages from 136 contributors and audited 278 packages in 242.775s found 0 vulnerabilities			~

6、打开 package.json 文件,将里面的内容全部删除,复制上去以下代码:

```
"scripts": {
"cypress": "cypress open"
}
```

2.3、运行 Cypress

{

}

- 1、运行命令: npm run cypress;
- 2、Cypress 窗口正常打开不报错,即环境安装成功。
- 3、examples 中是 Cypress 提供的 demo, 点击即可运行, 如下图:



le Edit View Window Help	
Crypress_project	🥑 Support 📾 Docs 🔒 Log In
✓> Tests ■ Runs ♥ Settings	Chrome 80 -
Q Search	► Run all specs
INTEGRATION TESTS	COLLAPSE ALL EXPAND ALL
actions.spec.js	
🗅 aliasing.spec.js	
assertions.spec.js	
C connectors.spec.js	
🗅 cookies.spec.js	
C cypress_api.spec.js	
🗅 files.spec.js	博为峰旗下 📃
Iocal_storage.spec.js	51tostuna
location.spec.js	UILCOLUIIY
misc spec is	软件测试网

第三章、实际案例

由于沐沐工作中的项目不方便作为案例,所以将以登录禅道记录缺陷作为 demo, 这样案例更具有参考性。

```
1、脚本代码如下:
   describe("禅道录入缺陷", function(){
   //登录模块
   beforeEach("登录系统",function () {
       cy.visit('url')
       //输入用户名
       cy.get('#account')
           .type("账号")
           .should('have.value', '账号');
       //输入密码
       cy.get('[name="password"]')
           .type("密码")
           .should('have.value', '密码');
       //点击登录按钮
       cy.get('#submit')
          .click();
   });
   it("提交 bug", function(){
       //点击测试
```





cy.get('[href="/zentao/qa/"]') .click(); //点击 bug cy.get('[data-id="bug"] > a') .click(); //点击【提 bug】按钮 cy.get('#createActionMenu > a.btn') .click(); //点击影响版本 cy.get('#openedBuild_chosen > .chosen-choices') .click(); //选择主干 cy.get('.active-result') .contains('主干') .click(); //输入 bug 标题 cy.get('#title') .type("bug 标题") .should('have.value', 'bug 标题'); //点击【保存】 cy.get('#submit') .click(); //断言是否新增成功 cy.get('.datatable-wrapper > .table > tbody > .text-center > .text-left > a') .contains('a', 'bug 标题') .should('be.visible'); });

});

2、脚本存放位置:在..\cypress\integration文件夹下面可以自定义文件存在脚本。如下图:







3、选中用例单击即可运行,如下图:

D:\Crypress_project	- 🗆 X
Crypress_project	🛿 Support 📾 Docs 🚨 Log In
✓/> Tests ■ Runs ✿ Settings	● 此处选择运行浏览器
Q Search 2	◆点击这里运行全部用例 ◆ Run all specs
▼ INTEGRATION TESTS COLLAPSE ALL EXPAND ALL	
 ▶ □ JG_Testcase ► Zentao ☐ ZendaoDemo.js 3 ◆选中单击运行单个用例 	
	(博为峰旗下)
	51testing
	软件测试网
Version 4.2.0 Changelog	

4、运行成功截图如下:





← → C ① 不安全 /	ests/integration\Zentao\Zendao	Demo.js						Q,	* 6	0 0
Chrome 正受到自动测试软件的控制。										,
< lesis ✓1 X - () - 19.16	e e nitp.	tentao/bug-brows	e-26.html							1366 X 768 (99)
~ 神道录入缺陷	《加速的回复》日数理系统								8	日本 日
✓ 提交bug	会现的地盘 产品 项目	制试 文化	5 9931 1999 A					1	Bug • (III) (c	tri+g) GO
* BEFORE EACH 1 VISIT http://izentao/user-login.html			ED/Alet						(C) in a second	ME AMOUNT
2 ggt Raccount	Bug Hild B	《本 著杆 旅言	用例库						(2) 数为数块	南京 公開以土3
- TYPE espected cinputRaccount.form-control> to have value cronefage	所有模块 未关闭 所有 指派	给我 由我创建 1	由我解决 未确认 未撤派 3	F解决 待关闭 久未处理 被延3	期 过期Bug 需求变动 Q.推	楝		白导出	• <u>III</u> 报表	+超Bug ・
5 GET [name="password"]	①根动监锁APP	D ID.	1591: P:	Ruolii20 :	Bun类型:	18:00 0	0021	AGE FINE:	83	彩化 創・
- TVPE - TVPE						-			0.4	Carl and
czongfang123456	维护模块	LI 3036	O teater Dogs	748	TOPHWIR	36025	-	03-30 14:29	QU/	
l gr #subelt	列表贝是否显示模块名	目选择 網	截 🔺				3	41条记录。每页	20 乐▲ 1/1	NAPN
(room sup)submitting form										
* 1657		and the second se								
1 szt [bref-"/zentao/qa/"] 🛶 በ 《 鼠标点击具	体步骤即可,右侧即可查看运行	过程								
(Mái LOND) +-page Loaded										
(MEW (ML) http:/										
2 -CLICK										
(PADF LDAD) page Louded										
(2017) (2013.)										
(NEW (BL)										
(NEW ORL) GET [data-id="bug"] > a										
<pre>(wrw unt) I get [data-id="bug"] > a 4 -CLTCK</pre>										
(NWY ML) GYT [data-dd="bug"] > a 4CLDC (MeD (040)page Loaded										
(NEW (ML)) 4 (41) 4 (41) 4 (41) (100) (10)										
(onv on) a cry [[ata-id="log"]] > a 4 -CLDC (ADD (100)page [cabd (NEV 01)] 5 ary #OrvateLcTonHeru > a.btn										
(www.wc.) 4 Get [deta-dd="bag"] > a 4CLOC (West Unit) (West Unit) 5 Get #Crowstate(ClowHenu > a.btn 4CLOC										
(2019 ML) 4 ML (data ide/hag*) > a 4CLOC (2001 ML)Oppe (2004 (2010 ML) 5 ML ROVARACEOPERU > a.0cn 4 - CLOC (2001 ML)Oppe (20040										
(even on.) [fatta bds ⁺ bag ⁺] > a (w) (box bds ⁺ bag ⁺) > ab (we black) rapp (and m ² -										
ext [dita:3de*bar] > s ext [dita:3de*bar] > s ext: [dita:3de*bar] (with int)										
even an.) left is ide"bag"] 5 a left is ide"bag"] 5 a (mail iso)popy (maint (mail iso) (mail iso)popy (maint (mail iso)popy (maint (maint (maint)) (mail iso)popy (maint (maint))										
ext [dita:3de*bar] > s ext [dita:3de*bar] > s cator [own onc) (wor onc) [own onc)										
core an;) [Eff.3d=dd=hgt] 5 a [If at dd=hgt] 5 a [I										
etc: [Esta ide"bag"] > a etc: [Esta ide"bag"] > a etc:										
core an.) statistical default[] s a statistical default[] s a core an.) statistical default[] s a core an.) statistical default[]	- Will + Jild + Walastadory - Bag								🥶 神道9 3 be	Ha 专业版 🛊 🤇
even any etc etc <td>·····································</td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td>● 70189.3.be</td> <td>ta 专业版 🛊 🔇</td>	·····································								● 70189.3.be	ta 专业版 🛊 🔇
(Person) WE (at 148-148-7mg*] > a (Person)rope Instant (Person)rope Instant (Person)rope Instant (Person)rope Instant (Person)rope Instant (Person)rope Instant (Person) (P	Will + Rich + Medicane + Bug								● 神田9.3 be	ta 专业版 🕈 🔇
even any etc.	WE - Not - Watshire - Bu								₩293b	ria 专业质 🛊 🗸
ore an.) left is dath and "ang" [5 • a eff is dath and "ang" [5 • a is dath and "ang" [5 • a (wat is als) -page (saladi	WEI、MIC、WOLLEAPP、Bag								● ₩203be	rta 带边质 ి d
att [Effild] > a att [Effild] > a att Forget failution (West interpolation)	神道,那点,那么能能能好学,Bog								● ₩203bc	rta 带业版 🛊 🗸
over an.) [dfat JdA*mg*] 5 * a ett - 0.0000 (war an.) - opper landed (war an.) - oppe	WH HIC HEALTHAP - Bug								● 神道0.3 be	eta 专业版 🕈 🤇
ter any t	MIR - SIGE - BAGEWAYP - Bag								● ₩20.3 be	ta 夺业派 🛊 4
ore an;) [dta:ds/dr/mg'] > a qt (dta:ds/dr/mg') > a (dta:ds/dr/mg') > a (dta:ds/dr/mg') > a (dta:ds/dr/mg') > (dta:ds/ds/mg') > (dta:ds/ds/dgt/ds/ds/ds/ds/ds/ds/ds/ds/ds/ds/ds/ds/ds/	Will , Bit , Watter/P , Bug								● 神道0.3 ba	nta 专业版 🛊 🤇
etc: active etc: active etc: active -representation e	Mit + Ric + Massave + Reg								● ₩2935x	tia 令说版 🏚 🗸
over an.) [dital.daftaqi"] 5 a att close (war an.) constant_interment	WH - Ric - Wattan - bu								● 神道D.3 be	eta 专业版 🛊 🤇

第四章、定位方式

Cypress 支持 id、name、class、css、text、xpath 等多种定位方式,但是需要注意的 是 xpath 定位需要额外安装 cypress-xpath 库。

4.1、xpath 定位

- 1、官网地址: https://github.com/cypress-io/cypress-xpath
- 2、安装命令: npm install -D cypress-xpath
- 3、在 cypress/support/index.js 文件中增加: require('cypress-xpath')









```
span>新增主项span> == $0
```

```
</button>
```

> <button data-v-5afd0aa1 type="button" class="el-button el-button--default el-button--small">...</button>
</div>

(1) xpath 定位举例:

cy.xpath('//span[contains(text(),"新增主项")]').click();

(2) class 属性定位举例:

cy.get('.el-icon-plus').click();

4.2、Demo 定位说明

元素定位的方式有很多种,我将以我的 demo 进行说明,demo 中的定位方式不一定 是最佳的但是都是正确有效的,在此详细说明,只是方便像我一样的测试小白进行参考。

(1) 通过标签名进行定位





*"*JT





定位方式如下:

//点击测试 cy.get('[href="/zentao/qa/"]') .click();

(2) 通过父级的 date-*属性进行定位

代码如下截图:



定位方式如下: //点击 bug cy.get('[data-id="bug"] > a')

.click();





《51 测试天地》五十七(下)

www.51testing.com

(3) 通过 i 标签的 class 属性进行定位

代码如下截图:



定位方式如下:

//点击【提 bug】按钮 cy.get('.icon-plus') .click();

(4) 通过父级的 id+子级的 class 属性进行定位





。管理系	统	↓ 退出	帮助 👻 关于	🕆 🖟 🚹 Elements Console Source	es Network 🄉 🔺 🕄
		Rug z (iii		▼	
@我的地盘 产品	项目 調试 又相 统计 组织 后台	bug -	Stoning Ooi	►	
- Pur	田岡 版本 泰州 祖生 田岡房		△湖は十万	▼	
• Bug	用例 版华 長叶 掀首 用例件		加满城主贝	div class="required"	red required-wrapper">_
				▼ <div class="input</td><td>-group"></div>	
			~	<span class="in
(/span)</td><td>out-group-addon">影明版本	
提Bug			8	▼ <span id="build</td><td>lox">	
				▶ <select id-<="" name="</td><td>openedBuild[]" td=""></select>	
新屋产品		・ 新屋橋は / ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・ ・		"openedBuild" s	ize="4" multiple="multiple"
en 847 - 18	0	sidd 100x Anim		class="chosen fr	orm-control" style="display:
所属項目		影响版本 创建发布 刷新 所有	•	v div class="ch	osen-container chosen-
1111-01-		Under Provint They		container-multi	" style="width: 100%;" title
当前指派	▼ 所有用户	載止日期		id="openedBuild	_chosen">
				··· Vul class="c	hosen-choices"> == \$0
Bug类型	代码错诊 ▼ 操作系统 ▼ 浏览器	•		► <li class="</td"><td>"search-field">_</td>	"search-field">_
				::after	
Bug标题	•	★ 严重程度 3 ▼ 优先级 ⑦ ▼			
				► <div class="</td><td>chosen-drop"></div>	
	H1+ 𝓕+ TT+ A+ A+ B I U ≡ ≡ ≡ Ξ Ξ Ξ	😑 尾 📴 🚥 📿 🍠 🍋 保存模板 🛛 应用模板 🕇			
	(北)370			► <span <="" class="int</td><td>out-group-addon fix-border" td="">	
重現步調	[少示]			id="buildBoxActio	ns" style="display: table-
	[四米]			tr trd div #buildBox #ononodBuild	chosen ul chosen-choices
	[明望]			a la construction reperiedulit	Chosen dichosen choices
				Styles Event Listeners DOM Breakpoint	s Properties Accessibility
10-2014	24150014/1000000	* * * * *		Filter :hov .cls +	position 0
12451		大幅问		element.style {	margin -
	选择文件 主法保任何文件	标题: + ×			border 1
MUC)#PER		1 P 7		<pre>#openedB bug_createD=0.html:39 wild + chosen_containen_multi</pre>	padding -
				.chosen-choices, #mailto +	0 - 1 - 64 × 28 - 1 - 0
	保存 这回			.chosen-container-multi .chosen-	
				border-top-right-radius: 0:	1
				border-bottom-right-radius: 0;	-
					0
				#module <u>bug-createD=0.html:37</u>	Filter Show all
				.chosen-single, #task + .chosen-	background-color
				container-single .chosen-single,	□rgb(255, 255, 255)
				#openedBulld + .chosen-container-	border-bottom-color

定位方式如下:

//点击影响版本

cy.get('#openedBuild_chosen > .chosen-choices')
 .click();

(5) 通过 class 属性+文本进行定位

	项目 測试 文档 统计 组织 后台 用例 版本 氧件 报告 用例库 <td< th=""><th>→ 退出 档 Bug • 第号(chi+g) 企</th><th>助 ▼ 关于 GO! 测试主页</th><th>Elements Console "program coption "program c/select V div cla container- container- container- container-</th><th>Sources ≫ ▲3 ; × n value—"trunk" data-keys- n zg">±∓ == 50 ss="chosen-container chosen- multi chosen-with-drop chosen- active" style="width: 100%;"</th></td<>	→ 退出 档 Bug • 第号(chi+g) 企	助 ▼ 关于 GO! 测试主页	Elements Console "program coption "program c/select V div cla container- container- container- container-	Sources ≫ ▲3 ; × n value—"trunk" data-keys- n zg">±∓ == 50 ss="chosen-container chosen- multi chosen-with-drop chosen- active" style="width: 100%;"
提Bug			\$	vilie id= v(u) cl. ::be: ▶ <li< th=""><th>openeodulid_cnosen > ass="chosen-choices"> fore class="search-field">_</th></li<>	openeodulid_cnosen > ass="chosen-choices"> fore class="search-field">_
所属产品	-Einelikert	所属模块 / ▼ 维护模块 刷新		::at 	ter
所属项目	-	影响版本 Q 创建发布 刷新 所有 *		v (u) v v (u) v	class="chosen-results">
当前指派	▼ 所有用户	截止日期		data	-option-array-index="0" style>
Bug类型	代码错误 ▼ 操作系统 ▼ 浏览器 •			2</th <th>'主干" Li></th>	'主干" Li>
Bug标题	-	★ 严重程度 😣 ▼ 优先级 ⑦ ▼			>
	H1- <i>F</i> - TT- A- A- B <i>I</i> <u>U</u> ≡ ≡ ≡ ∐ ⊟ ∞ ■	🥶 🚥 🖉 🄊 🍋 🔢 🐮 保存模板 应用模板 🕶		 	Tinut mus adda fin
	(步骤)			border" id="	"buildBoxActions" style=
重塊步振	[结果] [期望]			Find by string, selector, or XPath	Cancel
				Styles Event Listeners DOM Brea	akpoints Properties Accessibility
抄送给	选择要发信通知的用户 🕸 🎗	关键词		Filter :hov .cls +	margin -
附件(50M)	选择文件 未选择任何文件	标题: + ×		<pre>} *, zh-cn.defau=9.3.beta:6</pre>	border - padding -
	(477) (2011)			<pre>:after, :before { webkit-box-sizing:- border-box; -moz-box-sizing:- border- box; box-sizing: border-box; }</pre>	2 auto × auto 2 1
				<pre>select user agent stylesheet :-internal-list-box option {</pre>	Filter Show all
MMM & MILE &	A HERING	■ 100100 3 hota、大利化	85 🌲 🔥	<pre>line-height: initial limportant;</pre>	border-collapse separate
伸迫,例如,	> theoug	● 律道9.3.beta 号亚	wx 🕷 👄	/	border-box
定位方式; //选择主干 cy.get('.a	如下: ctive-result')				





《51 测试天地》五十七 (下)

www.51testing.com

.contains('主干')

.click();

(6) 过 id 进行定位

代码如下截图:

日管理系统	<u>د</u>	退出 帮助	1▼ 关于	Elemen	ts Console Sources » 🔺 3 🗄 🗙
企 我的地盘 产品	项目 测试 文档 统计 组织 后台 Bug	▼ 编号(ctrl+g)	GO!		<pre>cth>Bug标题</pre> /th>
- Bug	用例 版本 套件 报告 用例库	企課	则试主页		<pre><dv cuspan="2"> <dv cuspan="2"> <dv cuspan="2"> <dv cuspan="2"> <dv cuspan="2"> <dv cuspan="2"> <dv cuspan="2"> </dv> </dv> </dv> </dv> </dv> </dv> </dv> </pre>
提Bug		¢	•		<pre><div class="colorpicker input-group-
btn" data-<br="" data-toggle="tooltip">original-title title>_c/div> V<div class="input-group-required"></div></div></pre>
所属产品	・ 所属模块 / ・ 領	护模块 刷新			<pre>v<div class="required"></div></pre>
所属项目	▼ 影响版本 創建发布	刷新 所有 \star			<pre>id= title value class= form- control"> == \$0 ::after</pre>
当前指派	▼ 所有用户 載止日期				
Bug类型	代码错误 • 操作系统 • 浏览器 •				
Bug标题		先级 ⑦ -		<	
重現步骤		ATHIRUX *		Find by string, selector Styles Event Lister	rp_c/tr> div div div div input#titleform-control r, or XPath Prot Cancel Prot DOM Breakpoints Properties Accessibility
抄送给	选择要发信通知的用户 😢 📿 关键词			Filter :h element.style {	ov .cls + position 0
附件(50M)	选择文件 未选择任何文件 标题:	+ ×	(推)	} .inp zhicn.defa	border 1 padding 5
	<mark>6747</mark> 2600		5	<pre>digroup.innetu child, .input-grou addon:last-child, group-bentfirst-di group-vertical.mot child>.btn, .inpu btn:first-child>.l group:not(:first- .input-group-btn: child>.btn;not(:f .input-group-btn: child>.btn;niput</pre>	1 1 1 1 1 0 1
禅道 > 测试 >		1道9.3.beta 专业版	6 🕇 🖕	vertical>.btn, .ir	mput-group- Irgb(204, 204, 204)

定位方式如下:

//输入 bug 标题 cy.get('#title') .type("bug 标题") .should('have.value', 'bug 标题');

(7) 通过相对定位的并且断言是否新增成功(不推荐使用相对定位的方式)





^{₩为₩₩下} **51 Lest Ing** 软件测试网



定位+断言方式如下:

//断言是否新增成功

cy.get('.datatable-wrapper > .table > tbody > .text-center > .text-left > a') .contains('a', 'bug 标题') .should('be.visible');

第五章、登录封装

每一个用例都首先需要登录,在UI自动化脚本中一般是四个步骤,跳转到对应的 URL、输入用户名、输入密码、点击登录等,但是每个用例如果都是这样写代码就会显 得冗余,因此可将登录模块代码进行封装。

```
1、常规登录代码举例,代码如下:
```

//登录模块

```
beforeEach("登录系统",function () {
    cy.visit('url')
    //输入用户名
    cy.get('#account')
        .type("账户")
        .should('have.value', '账户');
    //输入密码
    cy.get('[name="password"]')
        .type("密码")
        .should('have.value', '密码');
        //点击登录按钮
        cy.get('#submit')
        .click();
});
```







2、在 cypress/support/commands.js 文件中, 自定义登录命令, 代码如下: ///登录禅道 Cypress.Commands.add('login_zentao', (name, password) => { cy.visit('url') //输入用户名 cy.get('#account') .type(name) .should('have.value', name); //输入密码 cy.get('[name="password"]') .type(password) .should('have.value', password); //点击登录按钮 cy.get('#submit') .click(); }); 3、每个用例中直接调用登录命令: //登录模块 beforeEach("登录系统",function () {

```
cy.login_zentao('登录账户','登录密码').wait(250)
```

});

4、用此方法可以再在 cypress/support/commands.js 中定义多个登录环境,在单个用 例中直接调用就会比较方便。

第六章、报告配置

1、mochawesome 安装说明: mocha 必须指定 7.1.1, 否则会报错。

2、安装命令:

npm install --save-dev mocha@7.1.1 mochawesome mochawesome-merge mochawesome-reportgenerator

(参考这篇文章: <u>http://www.51testing.com/html/87/n-4465287.html</u>时,文章中写到 "注意: mocha 必须指定 5.2.0, 否则会报错;",然后我安装了 mocha@5.2.0 还是报 错,踩坑两天后才醒悟,版本需要升级了)

3、配置 cypress.json 文件,代码如下: { "reporter": "mochawesome", "reporterOptions": {

"overwrite": false,





```
"html": true,
        "json": true
      }
    }
    4、编写生成报告的脚本,在 cypress 同级目录下新增 scripts 文件夹,文件夹下面新
增 cypress.js, 代码如下:
    {
        const cypress = require('cypress');
        const fse = require('fs-extra');
        const {merge} = require('mochawesome-merge');
        const generator = require('mochawesome-report-generator');
        async function runTests() {
            await fse.remove('mochawesome-report');
            const {totalFailed} = await cypress.run();
            const jsonReport = await merge();
            await generator.create(jsonReport);
            process.exit(totalFailed)
        }
        runTests()
    }
    5、package.json 文件添加启动脚本,代码如下:
    "scripts": {
                                        //正常运行 cypress 的命令
      "cypress": "cypress open",
      "report": "node scripts/cypress.js" //运行生成 mochawesome 报告的命令
    },
    6、运行命令: npm run report; 运行过程截图如下:
```



www.51testing.com

逝≄C:\Windows\system32\cmd.exe			×
nning: Zentao\ZendaoDemo.js			
道录入缺陷			
√ 提交bug(11080ms)			
hawesome] Report JSON saved to D:\Crypress_project\mochawesome-report\mochawesome_003.	json		
hawesome] Report HTML saved to D:\Crypress_project\mochawesome-report\mochawesome_003.	html 🚺	生成报告的路径	
(esults)			
Failing: 0 Pending: 0			
Skipped: 0			
Screenshots: 0 Video: true			
Spec Kan: Zentao\ZendaoDemo.js			
<u>ideo</u>)			
Started processing: <u>Compressing to 32 CRF</u>	-		
Finished processing: D:\Crypress_project\cypress\videos\Zentao\ZendaoDemo.js.mp4	2。脚本运行	,录制视频存放的路 径	준
	博 列暉湖		
		<u>256011</u>	

7、运行完成后,打开报告路径: ..\Crypress_project\mochawesome-report\

报告截图如下:

\equiv Crypress_p	roject	Ō 10.158s	E 11	Ê 13	@ 12	81
	Cypress.Commands Õ 1.785s ∎1 ✓1			*		
	Cypress.Cookies Õ1.233s ⊠3 ✓3			~		
	Cypress.Server ⓒ 390ms ■1 ✓1			*		
	Cypress.arch ♂ 293ms ■1 ✓1			~		
	Cypress.config() ⊙ 997ms			^		
	Get and set configuration options AssertionError: expected (Object (viewport/kight,)) to have property 'viewport/kight' of 660, but got 960			997ms Ō		
	Cypress.dom Õ 628ms ■ 1 ✓ 1			~		
	Cypress.env()			^		

第七章、踩坑记录

问题1: cypress 安装完成后直接运行,报错截图如下:



www.51testing.com

e Edit View Window Help bin Ø Sudd	port 🚖 Do	cs 💄	Log In
An unexpected error occurred			
EEXIST: file already exists, mkdir 'C:\Users\shero\node_modules.bin\cypress'			
▼ Stack trace			
<pre>[object Object] at fn (file:///C:/Users/shero/AppData/Local/Cypress/Cache/4.1.0/Cy at ipcBus (file:///C:/Users/shero/AppData/Local/Cypress/Cache/4.1. at Object.openProject (file:///C:/Users/shero/AppData/Local/Cypress at Project.componentDidMount (file:///C:/Users/shero/AppData/Local/Cypress at froject.componentDidMount (file:///C:/Users/shero/AppData/Local at ik (file:///C:/Users/shero/AppData/Local/Cypress/Cache/4.1.0/Cy at//node_modules/scheduler/cjs/scheduler.production.min.js.ex at fg (file:///C:/Users/shero/AppData/Local/Cypress/Cache/4.1.0/Cy at Yi (file:///C:/Users/shero/AppData/Local/Cypress/Cache/4.1.0/Cy</pre>	press/resour 0/Cypress/re AppData/Loca (constant) (constant) (constant) (press/resour (constant) (c	ce: soi l/(d)/(he, ce: le_ ce: ve:	
Copy to Clipboard	時間 「 した の した の した の の した の の した の した の した		ng
	欠 件 汃	刂试	XX
Version 4.1.0 Changelog			

问题原因: package.json 文件中没有定义运行命令;

解决方案: 打开 package.json 文件,将里面的内容全部删除,复制上去以下代码:

```
"scripts": {
"cypress": "cypress open"
}
```

}

{

问题 2: mochawesome 报告相关库安装完成后,无法运行报告,报错截图如下:

c: C:\Windows\system32\cmd.exe		_		\times
Cypress: 4.2.0 Browser: Electron 80 (headless) Specs: 4 found (JG_Testcase\Inspect\test.js, JG_Testcase\Listing\add-main-item.js, tcase\Listing\add-supervise_item.js, JG_Testcase\Listing\test.js)	JG_Tes			^
Running: IG Testcase\Inspect\test.is Error: Cannot find module 'mocha/lib/stats-collector'				ľ
<pre>Require stack:</pre>	o\reporter. o\project. o\modes\run o\modes\ind o\cypress. dex.js	js js 1. js dex. js js	16)	
at Function/lib/common/reset-search-paths.ts.Moduleresolverilename (electron/js2c/bi at Module_load (internal/modules/cjs/loader.js:691:27) at Module_load (electron/js2c/asar.js:717:26) at Function.Module_load (electron/js2c/asar.js:717:26) at Module.require (internal/modules/cjs/loader.js:853:19) at require (internal/modules/cjs/loader.js:853:19) at new Mochawesome (D:\Crypress_project\node_modules\mochawesome\src\mochawesome.js:86:3 at Reporter.setRunnables (C:\Users\shero\AppData\Local\Cypress\Cache\4.2.0\Cypress\resou	rowser_init 34) urces\app\;	t.js:7595: packages\s	16) erver\l	lib~

问题原因:是因为安装的 macha 版本与 mochawesome 版本不一致;





C:\Windows\system32\cmd.exe	- 0
D:\Crypress_project>r 'r' 不是内部或外部命令,也不是可运行的程序 或批处理文件。	
D:\Crypress_project>npm installsave-dev mocha@5.2.0 mochawesome mocha npm WARN eslint-plugin-chai-friendly@0.5.0 requires a peer of eslint@>=3 r dependencies yourself. npm WARN eslint-plugin-cypress@2.10.3 requires a peer of eslint@>= 3.2.1 pendencies yourself. npm WARN mochawesome@5.0.0 requires a peer of mocha@>=7 but none is inst	wesome-merge mochawesome-report-generator 3.0.0 but none is installed. You must install p but none is installed. You must install peer alled. You must install peer dependencies you:
17. npm <mark>WARN</mark> Crypress_project No description npm WARN Crypress_project No repository field. npm WARN Crypress_project No license field.	
+ mochawesome@5.0.0 + mochawesome-report-generator@4.1.0 + mochawesome-merge@4.0.3 + mocha@5.2.0 updated 4 packages and audited 554 packages in 25.405s found 4 low severity vulnerabilities	博为峰旗下
run `npm audit fix` to fix them, or `npm audit` for details D:\Crypress_project>npm run cy:run	51 testing
> @ cy:run D:\Crypress_project > node scripts/cypress.js	软件测试网

解决方案: mochawesome@5.0.0 对应 macha@7 以上的版本;

第八章、结束语

也许有些同学会觉得自己熟悉的语言是 python,并不熟悉 js,所以不会去尝试这个 框架,其实完全不用担心,沐沐也不会 js,但是沐沐会百度、会 copy、会一点英语,就 已经足够去写正常的 UI 自动化脚本了。适合自己产品的才是最好的。沐沐工作中使用 的前端框架是 vue,所以调研了很多 UI 自动化框架后,Cypress 无疑是最好的选择。



测试女巫紧跟时代脉搏之大数据分 析系列(2)

◆ 作者:测试女巫

一、前言

不禁叹声气,唉~上一期到这一期真的恍若隔世,真的世界乱套了,目前国内真的 比较平稳,而国外真的像地狱一样:用超大的冷冻车装载尸体,想象都不寒而栗是的, 这不是恐怖片,就是活生生的现实,人类真的如此脆弱,脆弱地不堪一击;好吧,不管 怎样,女巫还是健康的女巫,所以学习还是不能停,我还是很自豪,2020的春节假期期 间把 docker 搞定了: nginx, gunicorn, container, image, dockerfile 这些之前陌生的字 眼目前已经非常熟悉,且已经将自己的前后端数据库都已经使用 docker 正确发布,下一 步准备继续 k8s,哈哈真的有一些癫狂的状态,好吧,管它世界如何混乱,只要我和家 人还健康,学习就永不停止。

上一期我们学习了大数据分析的基础知识准备之 Jupter Numpy Pandas 的介绍;我们 这一期就开始**进攻大数据的学习思路之数据清理、数据整理与叙述统计**。

二、数据分析目标

数据挖掘分析领域最重要的能力是: 能够将数据转化为非专业人士也能够清楚理解 的有意义的见解。

利用统计学方法分析各类数据,对于结果,不只是靠感觉,靠经验,而是根据以数据为基础的科学分析方法来进行决策

(1) 来自公司内部的数据分析





經測試,軟體版本 (不論是在 Wi-Fi 2.4G 或 5G · TX 與 RX 曲 線幾乎相近. => 符合預期.

(2) 来自外部的数据分析要求

对于新入职的数据分析师,第一天接收的任务是:获得一份 50M 的 excel 表,来自 老板的数据分析:"这是 100 万条用户的信息,来个大数据分析:我们准备与星巴克合 作,来预测一下明年多少人喝咖啡,几千万的大项目,预测不准公司要亏很多钱"

(3) 来自网上的统计学的架构

描述统计学,把一些数据收集到一起,作图作表,求平均值,标准差或者看倾向这 些叫做描述统计学。 推论统计学,从总体取出一部分样本,通过样本的特点去推论总 体的特点,这种推论的统计叫做推论统计学。

査・论・编		統计學	[隐藏]
		集中趋势 平均数(平方・算术・几何・调和・算术-几何・几何-调和・希罗平均数不等式)・中位数・众数	
1002204002+400	连续概率	高敵程度 全距·标准差· 变异系数·百分位数·四分差·四分位数·方差·标准分数·切比雪夫不等式	
THACKENT		分布形态 偏态・峰态	
	离散概率 >>	欠数・列联表	
	推论统计学	置信区间・区间估计、显著性差异、元分析、贝氏推论	
1010.0011.00	实验设计	統计总量· 抽样 · 重抽样 (刀切法 · 自助法 · 交叉验证) · 重复 · 阻碍 · 特敏度 · 区集	•
推论统计学	样本量	统计功效,效应值,标准误差,零假设,对立假设,第一型和第二型误差,统计检定力	
假设检定	常规估计	贝叶斯推论,区间估计,最大似然估计,最小距离估计,矩量法,最大问题	
	特效检验	Z检验,学生t检验,F检验,卡方检验,Wald检验,曼-惠特尼检验, 秋和检验	
	生存分析	生存函数・乘积极限估计量・对数秩和检定・失效率・危险比例模式	
40 ¥ T.	相关性	氾浦变項・皮尔森积差相关系数・等级相关(史匹曼等级相关系数・肯德等级相关系数)・自由度 博为峰旗下	\bigcirc
相大及 同归分析	线性回归	线性模式·一般线性模式·广义线性模式·方差分析·协方差分析	F DO
	非线性回归	非多数回归模型・半多数回归模型・Logit模型 ジェレのgit模型	LUIIY
統计图形	饼園・条形園	■・双标图・箱形图・管制图・森林图・直方图・QQ图・趋势图・散点图・茎叶图・雷达图・示意地图 大方 ノノー いつ	1 2-1 100
其他	回应过程效度	· 统计课用	I II M

三、数据整理

往往我们拿到的数据,并不是我们想要的格式;或者其中包含了很多无用的信息; 例如 RF 在使用仪器厂商提供的工具时,最后得到的测试报告,有 10000 行;实际上对 于工程师而言,真正想要的数据,只有几百行;所以在分析数据前,就要先进行数据整 理;

四、数据清理

在分析数据之前必须对数据进行必要的检查以及整理,如果数据本身有问题,分析 的结果是没有任何意义的。主要检查数据的完整性,是否有空值,是否是异常值,是否





有重复;还有是否是正态分布的数据;因为数据分析的很多工具是建立在正态分布的基础上。

例如异常值分析是检验资料是否有录入错误以及含有不合常理的资料。忽视异常值的存在是十分危险的,不加剔除地把异常值包括进资料的计算分析过程中,对结果会产 生不良影响;重视异常值的出现,分析其产生的原因,常常成为发现问题进而改进决策 的契机。

五、数据分析基本概念

1、描述型:

即发生了什么,这是非常常见的一种分析方法,在业务中,它向分析师们提供重要 衡量标准的概览,就像 6 sigma 中的叙述统计一样,它可以计算出描述数据集中趋势的平 均数、中位数等,也能计算出描述数据分散程度的标准差 Q1 等,然后是均值,中位 数,标准差;

(1) 描述数据的集中趋势

• 众数 Mode

一组数据中,出现频率最高的值

• 平均数 Mean

● 中位数 Median

资料从小到大排序后,中位数即为最中间的那个数。当观测的值的个数为偶数,取 中间的两个数的平均数为中位数

(2) 描述数据的分散型态

• 全距 Range

一组资料内最大值与最小值的差异

四分位距 IQR

资料由小到大排序并分割为四个等分,每一个切割点为四分位数(quartile);四分之一位居(Inter-Quartile Range)IQR 就是第三四分位数(Q3)与第一四分位数(Q1)之差,如下图









● 变异数 Variance

从每一个资料点到平均数的差异值的平方的平均值,如下图

$$\sigma^{2} = \frac{\sum_{i=1}^{n} \left(x_{i} - \overline{x}\right)^{2}}{n-1}$$

● 标准差 Stanard Deviation

变异数的平方根

(3) 描述数据的分布

正态分布:实验工作者认为它是一个数学定理,数学研究者认为他是一个经验公 式;而我们可以把它理解为一种商业模式;因为它就像一个公理一样的存在。



Height in inches

我们看一下正态分布产生的过程: https://www.zhihu.com/question/26854682







偏斜度(Skewness):分析数据分布是不是正态分布

我们来复习一下6 sigma 有关正态分布的介绍,如下图

常態分配曲線之圖型決定於

1.平均數(µ):決定曲線的位置。

2.標準差(σ):決定曲線之形狀·即衡量曲線分散程度。

- σ愈小、曲線愈高聳、表數值集中於μ附近之數值愈多。
- σ愈大、曲線愈扁平、表數值集中於μ附近之數值愈少。



偏斜度是关于表现数据分布的对称性指标。如果其值为 0,则代表是一个对称性分 布(中位数约等于均值);如其值是正值,代表分布的峰值偏左(说明中位数小于均值);若 其值是负值,代表分布的峰值偏右(说明中位数大于均值)。

skewness 的绝对值如果大于 1,则是一个很明显的信号,说明你的数据分布有明显的不对称性。

例如我们读取以下的数据:如下图



《51 测试天地》五十七(下)

www.51testing.com



A	В	С	D	E
id	age	place	u	
0	1	11	101	
1	2	12	102	
2	3	13	103	
3	4	14	104	
4	5	15	105	
5	6	16	106	
6	7	17	107	
7	8	18	博力增於108	0
8	9	19	51 - (109)	r m
9	10	20	110	
			软件测	it

import pandas as pd

excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2')
excel_data
print("skew>>>>",excel_data.skew())

运行结果

 skew>>>> id
 0.0

 age
 0.0

 place
 0.0

 u
 0.0

 dtype: float64

```
• 使用 kstest 计算是否是正态分布
```

```
import numpy as np
import pandas as pd
from scipy import stats
excel_data = pd.read_csv('11adtoolsingle.csv')
excel_data_txpower = excel_data['Value']
u = excel_data_txpower.mean()
std = excel_data_txpower.std()
p_value = stats.kstest(excel_data_txpower, 'norm', (u, std))[1]
p_value
```

```
运行结果如下:
```

```
0.09290780539585992
```

(4) 使用 pandas 如何计算叙述统计数据

import pandas as pd df = pd.read_csv('qunar_freetrip.csv',index_col=0) df.describe()

运行结果:





有平均数,标准差,最小值,25%是第一四分位数(Q1);50%是中位数,75%是低三四分位数(Q3);最大值

具体的 Q1, 中位数, Q3 的意义请看下图



(5) 标准分数 Standard Score

• 定义

标准分数又称 z-score,中文称为 Z-分数或标准化值,它可以回答这个问题: "一个 给定分数距离平均数多少个标准差?"

它是将原始分数与团体的平均数之差除以标准差所得的商数,是以标准差为单位度 量原始分数离开其平均数的分数之上多少个标准差,或是在平均数之下多少个标准差。 它是一个抽象值,不受原始测量单位的影响,并可接受进一步的统计处理。在平均数之 上的分数会得到一个正的标准分数,在平均数之下的分数会得到一个负的标准分数。

• 公式表示为

z=(x-μ)/σ;其中z为标准分数; x为某一具体分数,μ为平均数,σ为标准差。

注意: sta 是 pandas 中的函数, 它的本质意义是:

sta = (exceldata['age']-exceldata['age'].mean())/excel_data['age'].std()

应用





在资料是正态分布的基础上,用上述的理论主要检验数据中的异常值,也就是离群 值,即数值明显偏离其余的观测值。常用的检测方法是3 σ原则;即 P(|x-μ|>3σ)≤ 0.003,即如果资料是正态分布,标准分数大于3的几率很小,小于或等于0.003

就是确认每个数值与平均数的距离,再除以标准差,用这个来衡量变异的程度,需 要进一步确认 sta 的意义

```
import pandas as pd
excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2')
sta = (excel_data['age']-excel_data['age'].mean())/excel_data['age'].std()
excel_data[sta.abs()>3
```

运行结果如下:

一定要注意:我们在 data.xlsx 中的第一个栏位是 id,而实际上我们使用 pandas 读取 的 xlsx 的文档,也会自动添加一个连续的 index,所以我们筛选出来的值会有两个 id。

```
id age place u
38 100 14 104
44 100 15 105
```

2、数据整理

(1) 透视表

它可以对数据进行动态排布并且分类汇总的表格格式,灵活性高,它可以随意定制 你的分析计算要求

在使用透视表之前要确保充分理解你的数据,并清楚地知道你想通过透视表解决什么问题,透视表拥有强大的数据整理功能,它的功能非常类似 excel 的透视表的功能。

(2) 函数以及常用 parameter

查看 pivot_table 函数,如下图





anado.pi	
as.pivot_table	e(data, values=None, index=None, columns=None, aggfunc='mean', fill_value=None, la=True_memins_name='4ll'_observed=False) [source]
Create a spreads objects (hierarch	sheet-style pivot table as a DataFrame. The levels in the pivot table will be stored in MultiIndex ical indexes) on the index and columns of the result DataFrame.
	data : DataFrame
	values : column to aggregate, optional
	index : column, Grouper, array, or list of the previous If an array is passed, it must be the same length as the data. The list can contain any of the other types (except list). Keys to group by on the pivot table index. If an array is passed, it is being used as the same manner as column values.
	columns : column, Grouper, array, or list of the previous If an array is passed, it must be the same length as the data. The list can contain any of the other types (except list). Keys to group by on the pivot table column. If an array is passed, it is being used as the same manner as column values.
Parameters:	aggfunc : function, list of functions, dict, default numpy.mean If list of functions passed, the resulting pivot table will have hierarchical columns whose top level are the function names (inferred from the function objects themselves) If dict is passed, the key is column to aggregate and value is function or list of functions
	fill_value : scalar, default None Value to replace missing values with 文 件 测 试 网

我们 study 四个重要的参数: index, value, columns, aggfunc

• index and values

每个 pivot_table 必须拥有一个 index,可以理解为参数;而且输入的次序不一样, 会有不同层次的索引

values 是需要计算的数据的筛选

注意不管 index 还是 values 都是可以输入多个参数;参数用 list 的形式传入如下

import pandas as pd import plotly.graph_objects as go excel_data = pd.read_csv('l1adtool.csv') result_test = excel_data.pivot_table(values="Value",index=["X-axis","Y-axis","Sector"]) result_test.head(20)

原始数据如下图







Α	В	С	D	E	F	G	Н
Wlan	X-axis	Y-axis	Channel	Sector	Sector Gain	Value	Test Report
wlan0	45	-10	Ch2:60.48GH	56	4	-44.3898	7220
wlan0	44	-10	Ch2:60.48GH	56	4	-44.379529	7219
wlan0	43	-10	Ch2:60.48GH	56	4	-44.468435	7218
wlan0	42	-10	Ch2:60.48GH	56	4	-44.502088	7217
wlan0	41	-10	Ch2:60.48GH	56	4	-44.205716	7216
wlan0	40	-10	Ch2:60.48GH	56	4	-44.442611	7215
wlan0	39	-10	Ch2:60.48GH	56	4	-44.579738	7214
wlan0	38	-10	Ch2:60.48GH	56	4	-45.178409	7213
wlan0	37	-10	Ch2:60.48GH	56	4	-46.268629	7212
wlan0	36	-10	Ch2:60.48GH	56	4	-47.200782	7211
wlan0	35	-10	Ch2:60.48GH	56	4	-48.664909	7210
wlan0	34	-10	Ch2:60.48GH	56	4	-49.222962	7209
wlan0	33	-10	Ch2:60.48GH	56	4	-50.654988	7208
wlan0	32	-10	Ch2:60.48GH	56	4	-50.302287	7207
wlan0	31	-10	Ch2:60.48GH	56	4	-49.262246	7206
wlan0	30	-10	Ch2:60.48GH	56	4	-47.377496	7205
wlan0	29	-10	Ch2:60.48GH	56	4	-46.035206	7204
wlan0	28	-10	Ch2:60.48GH	56	4	-45.061647	7203
wlan0	27	-10	Ch2:60.48GH	56	4	-44.543806	7202
wlan0	26	-10	Ch2:60.48GH	56	4	-44.365123	7201
wlan0	25	-10	Ch2:60.48GH	56	4	-44.264322	7200
wlan0	24	-10	Ch2:60.48GH	56	4	-44.211619	7199
wlan0	23	-10	Ch2:60.48GH	56	4	-44.387989	7198
wlan0	22	-10	Ch2:60.48GH	56	4	-44.592308	7197
wlan0	21	-10	Ch2:60.48GH	56	4	-45.063201	7196
wlan0	20	-10	Ch2:60.48GH	56	4	-46.148847	7195
wlan0	19	-10	Ch2:60.48GE	56	4	-47 535398	7194

上述代码运行完,如下图

			Value
X-axis	Y-axis	Sector	
		38	-39.202033
		39	-39.833337
		40	-44.721073
		41	-43.996115
		42	-48.498689
		43	-46.288638
		44	-46.834163
		45	-41.180928
		46	-51.850119
50	-10	47	-41.614514
-50		48	-45.019921
		49	-54.064241
		50	-42.506711
		51	-41.783221
		52	-49.600631
		53	-55.406758
		54	-45.005196





• aggfunc

当我们未设置 aggfunc 时,它默认 aggfunc='mean'计算均值; aggfunc 的函数是使用 numpy 其中的函数

我们假想它是一个 excel,这个函数计算的结果就是,它是筛选 X-axis 和 Y-axis 后 Value 值的总值以及平均值,

如下图

А	В	С	D	E	F	G	Н	- I
Wlan	X-axis	Y-axis	Channel	Sector	Sector Gain	Value	Test Report	ID
wlan0	-25	10	Ch2:60.48GH	19	4	-54.332341	3425	
wlan0	-26	10	Ch2:60.48GH	19	4	-52.476347	3424	
wlan0	-27	10	Ch2:60.48GH	19	4	-50.582677	3423	
wlan0	-28	10	Ch2:60.48GH	19	4	-48.761875	3422	
wlan0	-29	10	Ch2:60.48GH	19	4	-47.499998	3421	
wlan0	-30	10	Ch2:60.48GH	19	4	-46.33713	3420	
wlan0	-31	10	Ch2:60.48GH	19	4	-45.634368	3419	
wlan0	-32	10	Ch2:60.48GH	19	4	-45.01847	3418	
wlan0	-33	10	Ch2:60.48GH	19	4	-44.660961	3417	
wlan0	-34	10	Ch2:60.48GH	19	4	-44.426514	3416	
wlan0	-35	10	Ch2:60.48GH	19	4	-44.390966	3415	
wlan0	-36	10	Ch2:60.48GH	19	4	-44.809847	3414	
wlan0	-37	10	Ch2:60.48GH	19	4	-45.316486	3413	
wlan0	-38	10	Ch2:60.48GH	19	4	-46.230864	3412	
wlan0	-39	10	Ch2:60.48GH	19	4	-46.620346	3411	
wlan0	-40	10	Ch2:60.48GH	19	4	-47.71006	3410	
wlan1	-25	10	Ch2:60.48GH	19	4	-48.821119	3409	
wlan1	-26	10	Ch2:60.48GH	19	4	-50.681382	3408	
wlan1	-27	10	Ch2:60.48GH	19	4	-52.215193	3407	
wlan1	-28	10	Ch2:60.48GH	19	4	-52.93186	3406	
wlan1	-29	10	Ch2:60.48GH	19	4	-54.122759	3405	
wlan1	-30	10	Ch2:60.48GH	19	19.74	-53.139146	3404	
wlan1	-31	10	Ch2:60.48GH	19	- 4	-51.514594	3403	
wlan1	-32	10	Ch2:60.48GH	19	- 4	-50.921361	3402	
wlan1	-33	10	Ch2:60.48GH	19	4	-49.733256	3401	
wlan1	-34	10	Ch2:60.48GH	19		-49.12046	3400	
wlan1	-35	10	Ch2:60.48GE	19	平人 4	-48 207437	3399	NV.

import pandas as pd import numpy as np import plotly.graph_objects as go excel_data = pd.read_csv('11adtoolsingle.csv') result_test = excel_data.pivot_table(values="Value",index=["X-axis","Yaxis"],aggfunc=[np.sum,np.mean]) result_test.head(20)

运行结果如下图



《51 测试天地》五十七(下)

www.51testing.com



		sum	mean
		Value	Value
X-axis	Y-axis		
50	-10	-873.908388	-45.995178
-50	10	-883.491611	-46.499558
-49	-10	-871.895308	-45.889227
-43	10	-908.500660	-47.815824
-48	-10	-899.156014	-47.324001
-40	10	-912.727432	-48.038286
-47	-10	-928.892063	-48.889056
-47	10	-919.491880	-48.394309
-46	-10	-935.653503	-49.244921
40	10	-916.059748	-48.213671
-45	-10	-931.414938	-49.021839
-40	10	-915.934904	-48.207100
-44	-10	-913.992313	-48.104859
-44	10	-911.444911	-47.970785
-43	-10	-903.740514	-47.565290
-43	10	-899.983004	-47.367527

• columns

设定列的层次字段,即可以将列细分;即列进行展开

如下图, Wlan 有两个值(wlan0 和 wlan1), 我们可以将 Value 的按照 wlan0 和 wlan1 进一步区分,并进行计算

A	В	C	D	E	F	G	н	1
Wlan	X-axis	Y-axis	Channel	Sector	Sector Gain	Value	Test Report	ID
wlan0	-25	10	Ch2:60.48GH	19	4	-54.332341	3425	j
wlan0	-26	10	Ch2:60.48GH	19	4	-52.476347	3424	
wlan0	-27	10	Ch2:60.48GH	19	4	-50.582677	3423	J.
wlan0	-28	10	Ch2:60.48GH	19	4	-48.761875	3422	1
wlan0	-29	10	Ch2:60.48GH	19	4	-47.499998	3421	
wlan0	-30	10	Ch2:60.48GH	19	4	-46.33713	3420	ř.
wlan0	-31	10	Ch2:60.48GH	19	4	-45.634368	3419	r-
wlan0	-32	10	Ch2:60.48GH	19	4	-45.01847	3418	1
wlan0	-33	10	Ch2:60.48GH	19	4	-44.660961	3417	1
wlan0	-34	10	Ch2:60.48GH	19	4	-44.426514	3416	ji.
wlan0	-35	10	Ch2:60.48GH	19	4	-44.390966	3415	ii.
wlan0	-36	10	Ch2:60.48GH	19	4	-44.809847	3414	
wlan0	-37	10	Ch2:60.48Gł	19	4	-45.316486	3413	l.
wlan0	-38	10	Ch2:60.48GH	19	4	-46.230864	3412	1
wlan0	-39	10	Ch2:60.48GH	19	4	-46.620346	3411	
wlan0	-40	10	Ch2:60.48GH	19	4	-47.71006	3410	1
wlan1	-25	10	Ch2:60.48GH	19	4	-48.821119	3409	h-
wlan1	-26	10	Ch2:60.48GH	19	4	-50.681382	3408	J.
wian1	-27	10	Ch2:60.48GH	19	4	-52.215193	3407	1
wlan1	-28	10	Ch2:60.48GH	19	4	-52.93186	3406	j.
wlan1	-29	10	Ch2:60.48GH	19	4	-54.122759	3405	je.
wlan1	-30	10	Ch2:60.48GH	19	4	-53.139146	3404	
wlan1	-31	10	Ch2:60.48GH	19	4	-51.514594	3403	li
wlan1	-32	10	Ch2:60.48GH	19	4	-50.921361	3402	
wlan1	-33	10	Ch2:60.48GH	19	4	-49.733256	3401	
wlan1	-34	10	Ch2:60.48GH	19	4	-49.12046	3400	J
wlan1	-35	10	Ch2:60.48GH	19	4	-48 207437	3399	1





import pandas as pd import numpy as np import plotly.graph_objects as go excel_data = pd.read_excel('data.xlsx',sheet_name='Sheet2') excel_data.head(5) result_test = excel_data.pivot_table(index=["X-axis","Y-axis","Sector"],columns = ["Wlan"],values="Value",aggfunc=[np.sum,np.mean]) result_test.head(20)

运行结果如下图

			sum		mean	
		Wlan	wlan0	wlan1	wlan0	wlan1
X-axis	Y-axis	Sector				
-40	10	19	-47.710060	-53.139146	-47.710060	-53.139146
-39	10	19	-46.620346	-54.122759	-46.620346	-54.122759
-38	10	19	-46.230864	-52.931860	-46.230864	-52.931860
-37	10	19	-45.316486	-52.215193	-45.316486	-52.215193
-36	10	19	-44.809847	-47.516551	-44.809847	-47.516551
-35	10	19	-44.390966	-48.207437	-44.390966	-48.207437
-34	10	19	-44.426514	-49.120460	-44.426514	-49.120460

• groupby

是 index and values, aggfunc 实现表格重组的另一种方式

```
import pandas as pd
import plotly.graph_objects as go
excel_data = pd.read_excel('data.xlsx',sheet_name='Sheet2')
result_test =excel_data.groupby(['X-axis','Y-axis'])['Value'].mean()
result_test
```

运行结果如下

X-axis	Y-axis	
-40	10	-50.424603
-39	10	-50.371553
-38	10	-49.581362
-37	10	-48.765840
-36	10	-46.163199
-35	10	-46.299201
-34	10	-46.773487
-33	10	-47.197108
-32	10	-47.969916





-31	10	-48.574481			
-30	10	-49.738138			
-29	10	-50.811379			
-28	10	-50.846868			
-27	10	-51.398935			
-26	10	-51.578865			
-25	10	-51.576730			

Name: Value, dtype: float64

• query

在 dataframe 中通过 query 函数,可以在一个栏位中,查找符合 query 条件的所有数

据

excel_data.query('Wlan == ["wlan1"]')

运行结果如下图

	Wlan	X-axis	Y-axis	Channel	Sector	Sector Gain	Value	Test Report ID
16	wlan1	-25	10	Ch2:60.48GHz	19	4	-48.821119	3409
17	wlan1	-26	10	Ch2:60.48GHz	19	4	-50.681382	3408
18	wlan1	-27	10	Ch2:60.48GHz	19	4	-52.215193	3407
19	wlan1	-28	10	Ch2:60.48GHz	19	4	-52.931860	3406
20	wlan1	-29	10	Ch2:60.48GHz	19	4	-54.122759	3405
21	wlan1	-30	10	Ch2:60.48GHz	19	4	-53.139146	3404
22	wlan1	-31	10	Ch2:60.48GHz	19	4	-51.514594	3403
23	wlan1	-32	10	Ch2:60.48GHz	19	4	-50.921361	3402
24	wlan1	-33	10	Ch2:60.48GHz	19	4	-49.733256	3401
25	wlan1	-34	10	Ch2:60.48GHz	19	4	-49.120460	3400
26	wian1	-35	10	Ch2:60.48GHz	19	4	-48.207437	3399
27	wian1	-36	10	Ch2:60.48GHz	19	4	-47,516551	3398
28	wlan1	-37	10	Ch2:60.48GHz	19	4	-52.215193	\$407
29	wlan1	-38	10	Ch2:60.48GHz	19	4	-52.931860	25 <i>E</i> 3406
30	wian1	-39	10	Ch2:60.48GHz	19	4	-54:122759	: 测 铅

使用 groupby 在基于原表格栏位的基础上产生新的栏位

注意这个功能不管是使用被 pivot_table 整理后的表格或者直接使用读取 csv 的表格,都是可以的~

import pandas as pd
import plotly.graph_objects as go
excel_data = pd.read_csv('l1adtoolsingle.csv')
result_test = excel_data.pivot_table(values="Value",index=["X-axis","Y-axis","Sector"])
result_test_group =result_test.groupby(by="Sector").size().reset_index(name='个数')
result_test_group.sum()
#result_test_group['个数']





result_test_group['占比']=result_test_group['个数']/result_test_group.sum()[0] result_test_group

七、数据梳理

一组数据是不是可以被分析,首先需要观察这组数据是不是存在异常情况,这些异常情况是不是值得被修复,如果数据损坏严重,例如缺失值达到 50%,这组数据就没有分析的价值。

一定要注意所谓对数据的修复,并不会对读取的数据文档进行任何改变,而只是对 于读取后存在变量中的数据进行修复梳理清理。

一般的数据分析以连续性数据为主,因为很多分析方法都是针对连续性且正态分布的数据,而在使用各种工具或者 function 对数据进行分析前,需要先对数据进行分析,然后进行必要的数据清理,确认数据可以被分析后,才能进行数据分析。

1、数据的基本属性分析

(1) info

数据有多少 entries,每个栏位的 type,每个栏位占多少内存

```
import pandas as pd
excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2')
excel_data.info()
```

运行结果

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 4 columns):
id 10 non-null int64
age 10 non-null int64
place 10 non-null int64
u 10 non-null int64
dtypes: int64(4)
memory usage: 448.0 bytes
```

假如我把 place 栏位中删除一个数据,我在 age 栏位中写一个字串: "我",再次使用 info()查看数据类型,就可以看出来 place 栏位的 non-null 变成 9个; age 的 type 变成

object(可以暂时将其理解为 string)

<class 'pandas.core.frame.DataFrame'> RangeIndex: 10 entries, 0 to 9 Data columns (total 4 columns):





id 10 non-null int64 age 10 non-null object place 9 non-null float64 u 10 non-null int64 dtypes: float64(1), int64(2), object(1) memory usage: 448.0+ bytes

(2) shape

查看 excel 表格有多少行多少列,就是这组数据的形状

import pandas as pd

excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2')
excel_data

运行结果

即10行4列

(10,4)

2、数据缺失

取值为空(null)的值为缺失值。缺失值比例是确定该字段是否可用的重要指标。一般情况下,如果缺失率超过 50%,则该字段就完全不可用。

```
import pandas as pd
excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2')
excel_data
excel_data.isnull().sum()
```

一定要注意区别对待 null 和 0 的关系; null 为缺失值, 0 是有效值。举一个例子: 某客户在银行的某账户余额为 null, 一位置该客户没有这个银行的账户; 如果将 null 改 为 0, 则说明该用户有账户, 但是账户余额为 0。

运行结果

```
即 place 中有一个栏位是 null, 其它栏位没有缺失的栏位
```

```
        id
        0

        age
        0

        place
        1

        u
        0

        dtype:
        int64
```

3、数据重复

所谓重复的定义是两行数据一模一样,

import pandas as pd




^{₩≫₩₩™} **51 Lest Ing** 软件测试网

```
excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2')
   excel data
   excel_data.duplicated().sum()
   加入 data.xlsx 中有两行数据完全相同,则运行结果如下:
   1
   4、简单的数据清理
    (1) 重复数据处理
       重复的数据到底有哪些
   import pandas as pd
   excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2')
   excel_data[excel_data.duplicated()]
   运行结果
   id 45 指的就是 id 从 0 开始计算, 第 4 和 5 行是重复的值
    id age
           place
                 u
   54
        8 15 105
       删除重复的数据
   将 data.xlsx 中资料创造复制一笔重复的值, 然后使用 drop_duplicate 删除重复的
值,再使用
   exceldata[exceldata.duplicated()查看重复的值,以及查看重复值的数量,来确认是否
已经全部山吃]
   import pandas as pd
   excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2')
   excel_data.drop_duplicates(inplace = True)
   excel_data[excel_data.duplicated()]
   excel_data.duplicated().sum()
       删除重复数据后处理索引
   删除重复数据后,缓存中的数据的索引,会存在断续的现象,需要将数据的索引处
```

理为连续的,不能是断续的索引

首先查一下删除后有多少行,还是使用 shape,原来 data.xlsx 有 10 行 4 列

import pandas as pd

excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2')
excel_data.drop_duplicates(inplace = True)



excel_data.shape

运行结果:

(9, 4)

所以可以看到删除了一行

如果数据不多,可以将内容全部显示出现确认一下

import pandas as pd

excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2')
excel_data.drop_duplicates(inplace = True)
excel_data

运行结果如下

可以看到索引为6的那一行被删除了

id	age	place	e	u
09	1	11	101	
11	2	12	102	
29	3	13	103	
38	4	14	104	
44	8	15	105	
55	8	15	105	
77	8	18	108	
88	9	19	109	
99	10	20	110	

重新对于 index 进行连续化处理

import pandas as pd

excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2')
excel_data.drop_duplicates(inplace = True)
excel_data.shape[0]
excel_data.index = range(excel_data.shape[0])
excel_data.index

```
excel_data
```

运行结果,可以看出 index 已经恢复正常

RangeIndex(start=0, stop=9, step=1)

id	age	place	e	u
09	1	11	101	
11	2	12	102	
29	3	13	103	
38	4	14	104	
44	8	15	105	
55	8	15	105	



《51 测试天地》五十七(下) www.51testing.com



67 8 18 108
78 9 19 109
6 9 10 20 110
(2) 栏位的处理
查看数据的栏位,是否正常,是否有空格或者空白等异常情况
<pre>import pandas as pd excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2') excel_data.columns</pre>
运行结果
可以发现 id 栏位存在空格
Index([' id ', 'age', 'place', 'u'], dtype='object')
将无用的空格拿掉
<pre>import pandas as pd excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2') col = excel_data.columns col[0].strip()</pre>
运行结果
'id'
可以使用迭代将所有的栏位迭代一遍
excel_data.columns = [x.strip() for x in col] excel_data.columns.values
运行结果如下
array(['id', 'age', 'place', 'u'], dtype=object)
(3) 异常值即数据缺失的处理
假如 data.xlsx 中的 age 这个栏位中有两个空白的 item
<pre>import pandas as pd excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2') excel_data.isnull().sum()</pre>
excel_data[excel_data.age.isnull()]
运行结果,可以发现 index 为 3 和 4 的 item, 它的 age 栏位为空
id age place u
38 NaN14 104
4.4 INAN 15 105

对于缺失值的处理,可以根据当时的数据实际情况,灵活选择;例如 KM id #36 中





就是通过对于"路线"栏位的解析,来获得"出发地",并将出发地为缺失的值补齐。可 以参考链接 http://172.16.201.48:8088/open.knowledge/view/369 我们这个例子是获取 age 为空的栏位。填入一个默认值 import pandas as pd excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2') excel_data[excel_data.age.isnull()] excel_data.loc[excel_data.age.isnull(),'age'] = 15 excel_data 运行结果 id age place u 0 9 1.0 11 101 1 102 2.0 12 1 2 9 3.0 13 103 3 8 15.0 14 104 4 4 15.0 15 105 5 5 8.0 15 105 6 5 8.0 15 105 7 7 8.0 18 108 8 8 9.0 19 109 9 9 10.0 20 110 (4) 异常值即不合理值的处理 加入 data.xlsx 中的 age 这个栏位有两个数据为不合理: 100.假如我们的这份档案是

入职人员的 age,入职人员的 age 不可能是百岁老人,所以这样的异常值也需要提前检测以及被处理。

我们先用 skew 确认一下这组数据的分布

mport pandas as pd

excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2')
excel_data['age'].skew()

运行结果如下

1.5324638284098495

结果大于1,且为正值,说明数据的分布是靠右,峰值靠左,也就是数据偏大

我们使用统计学的知识将变异数比较大的栏位找出来如下:

```
import pandas as pd
excel_data = pd.read_excel('data.xlsx',sheet_name='worksheet2')
excel_data.describe().T
sta = (excel_data['age']-excel_data['age'].mean())/excel_data['age'].std()
```





《51 测试天地》五十七(下)

www.51testing.com

excel_data[sta.abs()>1]

运行结果如下					
id	age	place	u		
38	100	14	104		
44	100	15	105		





带你走进 Jmeter 参数化

◆ 作者:小雯子打豆豆

如果想要掌握 Jmeter 的相关操作,那么前提是你一定要会参数化,那什么是参数化 呢?我的理解,就是将脚本中的某些输入值使用参数来代替,在下次使用的时候,只需要 修改输入值既可。

还是听不懂没关系,看了下面的你就会了,常见的参数化方法有如下几种:

一、CSV Data Set Config:CSV 数据控件

使用场景:适合参数范围比较大的时候,可以方便用户随时修改参数数据。使用最 多的就是参数化用户名和密码登录系统。



准备 txt 文档参数数据





1.txt - 记事本	
文件(E) 编辑(E) 格式(Q) 查看(V) 帮助(H)	
xiaodoudou <mark>1,</mark> ,123abc xiaodoudou <mark>2,,</mark> 123abc	注意这里的逗号是英文格式的

参考接口: http://api.nnzhp.cn/api/user/login

	CSV 数据文件设置	
	名称: CSV 数据文件设置	
	注释:	
	设置 CSV 数据文件	
	文件名:	C:/Users/EDZ/Desktop/1.bt
	文件编码:	
	变重名称(西文逗号间隔):	username,passwd
	忽略首行(只在设置了变重名称后才生效):	False
	分隔符 (用'\t'代替制表符):	
	是否允许带引号?:	False
2	遇到文件结束符再次循环?:	True
ŝ	遇到文件结束符停止线程?:	False
	线程共享模式:	所有现场

通过 Debug Sampler 来查看获取的参数是否正确

参数 消息体数据 文件上传			
			同请求一起发送参数:
			值
username		\${username}	
passwd		\${passwd}	
	TestPlan.comments= JMeterVariables: JMeterThread.last_samp JMeterThread.pack=org. START.HMS=143428 START.MS=1585636468 START.YMD=20200331 TESTSTART.MS=158563 im线程组_参数化 imeter.U_T=线程组 passwd=A123abc username=xiaodoudou1	ple_ok=true apache.jmeter.threads.Sampl 3855 37748037 _idx=0 _参数化 1-1	lePackage@7c2e69fa

二、函数助手

使用场景: jmeter 自带的函数处理,比如:随机数,可用于固定字符+随机数,修改 传入参数的值,函数助手的底层都是 java 封装的方法。





函数助手			
选	择一个功能Random	▼帮助	
	函数参数		生成汤米布
名称: 范围内的最小值 范围内允许的最大值	1 10		
拷贝并粘贴函数字符 <mark>串</mark> \${Rand	pm(1,10,)}	生成	Reset Variables
result of the function i :	-	函数返回值	
1 START.MS=1585 2 DMeterThread. 3 START.MMS=233 4 START.YMD=2020 5 5	568809009 Last_sample_ok=true 229 90331	^{時入睡旗下} 51との 软件 测	Lung 」 试 网
基本 高级 Web服务器 协议: http 服务器名称或IP: HTTP请求 方法: POST ▼ 路径	api.nnzhp.cn/api/user/login		端口号: 内容编
自动重定向 ✔ 孫膝重定向 ✔ 使用 KeepAlive	対POST使用multipart / form-data ———————————————————————————————————		
参观 相忌体效据 文件工传	同请求一起发	送参数:	
名称 : username passwd	1〕 xiaodoud\${ Random(1.10 aA123456	.)}	ntent-Type
ext ▼ 取样認结果 滿 ② HITP商家 ③ Debug Sampler 1 POST http username= 6 [no cooki	求 响应数据 Request Headers ://api.nnzhp.cn/api/user/login/ :iaodoud8&passwd=aA123456 rs]	道机的用户名	Find 医分大小3
三、用户自定义变量			
使用场景:一般可以适用于	在测试计划中不经常	宫变化的参数设置, 女	□ host、端口
号、url等;而且用户自定义变量	是在配置元件中的。		
文件 編編 査携 运行 逃项 帮助 JDBCO Java明 LDAP LDAP LDAP LDAP LDAP LDAP LDAP LDAP LDAP LDAP LDAP LDAP LDAP LDAP LDAP 使用 LDAP	onnection Configuration 认请求 一次请求我认值 认请求 若程2番 MPP Connection - Lock File Config - Lock File Config - Variables From CSV File PVT 2015年 電子に伴楽 置元件 選 置元件 選 直封需要	 	





《51 测试天地》五十七(下)

www.51testing.com

名称: 用户完义的变量				
注释:				
		用户定义的变量		
名称:		值	C	escription
base_url port	api.nnzhp.c 8080	cn/		
上田人一丁に一				
吉朱如卜所不:				
Web服冬哭				
hiù. http. 眼冬哭	么称武ID. Sthase unt			一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一
				刘明山 行:
HIIP項水				
方法: GET	▼ 路径: api/user/s	stu_info		内容编码
📄 自动重定向 🗹 跟随重定向 🗹 🕅	使用 KeepAlive 对POST使用multipart,	/ form-data 🔲 与浏览蔡萧察的头		
			1.1 2 2	
参数 消息体数据 文件	上传			
参数 消息体数据 文件	牛上传	同请求一起发送参数	<u>k. 511</u> 2	sting
参数 消息体数据 文件 名称:	牛上传	同请求一起发送参数	ه: 5166 ه 47 40	
参数 <mark>消息体数据 文件</mark> 名称 : stu name	牛上传 mini123	同请求一起发送参 多 值	教: 编 text/plain C	SLUng
参数 消息体数据 文件 名称 : stu name	牛上传 mini123	同请求一起发送参数 值	数: 编 text/plain	sting
参数 消息体数据 文件 名称: stu name ≈xt ♥ 町町画案	牛上传 mini123 P#器结果 请求 响应数据	同请求一起发送参考 值 :吉式专工:) 品 (約11)	数: 编 text/plain C	sting
参数 消息体数据 文件 名称: stu name ≈xt ♀ ⅢⅢ酿家	牛上传 mini123 R样器结果 请求 响应数据 tequest Body Request Headers	同请求─起发送参数 值 1<u>青求正确的</u>ur	≴: 编 text/plain C	pontent-Type
参数 消息体数据 文件 Stu name ext ② ⅢⅢBitk	牛上传 mini123 R祥器结果 请求 町向広納招 sequest Body Request Headers 1 GET http://api.nnzhp.cn//api/	同请求一起发送参数 值 请求正确的ur ^{(user/stu_info?stu_nome=mini123}	数: 编 text/plain C	Find 区的大作
参数 消息体数据 文件 名称: stu name axt ♀ ♥■■PF##	牛上传 mini123 RF器結果 海求 吻应動揺 Request Body Request Headers 「GET http://api.nnzhp.cn//api/ SET data:	同请求一起发送参数 值 请求正确的ur user/stu_info?stu_name=mini123	数: 编 text/plain	Pind CATA

四、用户参数

使用场景:适合参数取值范围很小的时候使用 一般会将登录的用户账号和密码作 为用户参数,用户参数是在前置处理器模块中的

SSOL
e







username=xiaowenzi9&passwd=Aabc123

五、关联获取参数-正则/json 提取(个人角度)

常见的参数化是四种,个人理解通过正则和 json 提取也是一种提取返回结果作为 参数的一种功能,常见的使用场景有:用户登录时,获取返回结果 token,作为下一个 请求的参数,这里要注意取参数的顺序,否则可能会出现还没提取出参数的值,其他请 求却已经引用了参数的情况。

[no cookies]

▶♣ 获取	添加	►	取样器	►		
▼ቆ 登录	为子线程添加响应时间		逻辑控制器	►		
⊿ 察	启动		前置处理器	►	l行的动作————	
🖉 De	不停顿启动		后置处理器		CSS/JQuery提取器	
▲ 沃力	验证		断言	Þ	JSON提取器	
▼■ 45月 ▼● 45月	剪切	Ctrl-X	定时器	Þ	正则表达式提取器	
/ HT	复制	Ctrl-C	测试片段	►	边界提取器	
∋ JS	粘贴	Ctrl-V	配置元件	►	JSR223 PostProcessor	
	复写	Ctrl+Shift-C	监听器	•	Debug PostProcessor	
	删除	Delete	冷配置		JDBC PostProcessor	

参考接口: http://api.nnzhp.cn/api/user/login 获取用户的 cookie,并将 cookie 作为参数请求下一个连接。这样的原因是,能保证接下去的操作是在当前用户登录后操作的。



《51 测试天地》五十七(下)

www.51testing.com





通过 json 提取器获取对应的 cookies 值,并添加在下一个请求的 http cookie 管理器

中。

JSON提取器		
名称: JSON提取器		
注释:		
Apply to:		
● Main sample and sub-samples ● Mair	n sample only 🏾 Sub-sampl	es only 🏾 JMeter Variable Nam
Names of created variables:	cookie	
JSON Path expressions:	\$.login_info.sign	博为峰旗下
Match No. (0 for Random):		Stesting
Compute concatenation var (suffix _ALL)	:=	
Default Values:		
存储在Cookie管理器中的Cookie		
名称: 值	域	路径: 安全
cookie niuhanvand=\${cooki	e}	

通过 debug sampler 来查看取值结果,说明我们已经取到对应的 cookie 值了。



看到这里,应该知道如何进行参数化了吧,学会了参数化,我们使用 jmeter 就会越 加方便啦!

■拿下"Jmeter,轻松应对性能测试>><u>https://dwz.cn/12IkuTSj</u>





一文详尽 Jmeter 对数据库批量增 删改查 ◆ 作者:风行万里

本文主要的内容是使用 Jmeter 对数据库进行数据的操作,包括单条语句的增删改 查,多条语句的增删改查,本文主要介绍操作流程,关于流程的运作原理,对原理感兴趣的同学可自行查阅资料。

首先需要准备一个数据库驱动 Jar 包【mysql-connector-java-bin.jar】可以直接百度下载,版本不建议特别低, jar 包需要 Jmeter 引用才能使用, jar 包引用如下:

将 jar 放入 Jmeter 安装/解压文件路径: eg: D:\apache-jmeter-5.0\apache-jmeter-5.0\lib 下, 需要重启 Jmeter 才能生效

驱动包的作用:没有数据库驱动,就不能连接数据库,Jmeter 在没有驱动的情况下 链接数据库会报: no suitable jdbc driver 错误。

以下是常见的数据库驱动:

常见的数据库驱动

针对不同类型的数据库,需要的驱动也是不一样的!如下所示:一般都是放到1ib下

数据库类型	数据库Driver class	Database URL
MySQL	com.mysql.jdbc.Driver	jdbc:mysql://host[:port]/dbname
PostgreSQL	org.postgresql.Driver	jdbc:postgresql:{dbname}
Oracle	oracle.jdbc.OracleDriver	jdbc:oracle:thin:@//host:port/service OR jdbc:oracle:thin:@(description=(address=(host= {mc-name})(protocol=tcp)(port={port- no}))(connect_data=(sid={sid})))
Ingress (2006)	ingres.jdbc.IngresDriver	jdbc:ingres://host:port/db[;attr=value]
Microsoft SQL Server (MS JDBC driver)	com.microsoft.sqlserver.jdbc.S QLServerDriver	jdbc:sqlserver://host:port;DatabaseName=dbname 博为峰旗下 511051
Apache Derby	org.apache.derby.jdbc.ClientDr iver	jdbc:derby://server[:port]/databaseName[;URLAttribute s=value[;]] 软件测试网

图一

汉化 Jmeter: 打开 JMeter 解压/安装路径下的文件 JMeter.properties,在 #language=en 下面添加: language=zh_CN 保存文件,并重启 JMeter。





一、主要配置元件介绍(以下为增删改查操作的共有元件)

如图主要元件如下:



1、添加测试计划: 各个值直接默认就好, 为空的地方也不用填

2、添加线程组: 各个值默认, 为空的地方也不用填

3、添加 JDBC 配置文件 (JDBC Connection Configuration):

重要字段介绍

名称:设置的是该元件的名称,设置名称后【测试计划】树配置元件也会对应更改

Variable Name Bound to Pool:数据库连接池的名称。可以有多个 jdbc connection configuration,每个可以起不同的名称,在 Jmeter 其他元件中只要需要用到数据库的连接信息,直接引用该变量的变量名即可。可以理解为如果 Jmeter 其他的元件要获取数据 库的连接信息,可以通过这个名称进行获取。

注意:变量的命名规范:命名要清晰,建议使用英文,便于引用。

Connection Pool Configuration 、 Connection Validation by Pool 这两部分内容不需要更改,使用默认值即可

Database Connection Configuration (以 MySQL 数据库为例):

Database URL: jdbc:mysql://host[:port]/dbname ——>【数据库地址: jdbc:mysql://数据库主机名或 IP 地址:端口号/需要使用的库名】

JDBC Driver class: com.mysql.jdbc.Driver【其他数据库跟进图一展示进行选择】

Username:数据库名称,即用户名

Password: 数据库链接密码





自己的测试库 - 连接信息	
常规	
连接名:	自己的测试库
主机名或 IP 地址:	rm-m5ep01a1
端口:	3306
用户名:	hjhp.
设置位置:	D:\Documents\Navicat\MySQL\servers
编码:	65001 (UTF-8)
状态:	已连接
服务器版本:	5.6.16-log
通讯协定:	10
信息:	rm-m5ep01a1rv5jt3d01oo.my. cs.com v
SSH 主机名或 IP 地址:	N/A
HTTP 通道地址:	N/A
该图为:数据库 不早加	链接可视化工具Navicat截图频频率
复制到剪贴板	新日本

4、添加配 JDBC Request 数据库请求

重要字段解释:

名称:同 JDBC 配置文件

Variable Name Bound to Pool: 引用 JDBC Connection Configuration 元件里面的参数 值【test】写法两边保持一致即可,不需要使用\${}进行引用。错误的不存在的参数不能 被引用。

Query type: 必填,指 SQL 请求类型

Select statement: 查询语句类型 (select), 只支持一条查询语句, 多条查询语句只执 行第一条

Update statement: 更新语句类 (insert, update, delete), 只支持一条更新语句, 多条更新语句只执行第一条

Prepared Select statement: 支持多条查询(select)语句, 查询响应数据只展示第一条 SQL 的查询结果

Prepared Update statement: 支持多条更新(insert, update, delete)语句, 响应数据 展示多条更新提示

Callable Statement: 支持多条查询、更新(insert, update, delete, select)语句, 响



应数据展示展示多条数据更新结果。如果是多条 select 语句同时查询,建议使用 Callable Statement,响应数据可以展示多条查询结果值

Parameter values: 填写参数的具体的值,或者参数的名称。可以利用此字段对 SQL 语句进行参数化

Parameter types: 指 Parameter Values 参数的数据类型,例如: integer, String, double 类型

Parameter values 和 Parameter types: 必须成对出现,且 SQL 语句中有多个参数,就必须有多少个 parameter values 和 Parameter types。

Variable names: 自己设置的变量名称,用于存放 select 操作返回的查询结果。有多个字段返回时,需用逗号隔开

Result variable name:用于存放 select 操作返回的查询结果集

Query timeout: 查询超时时间

Handle result set: 定义如何处理由 callable statements 语句返回的结果

5、添加察看结果树

在响应数据——Response Body: 查看对数据库进行增删改查后的返回消息

6、测试表名 city, 表结构如下:

CREATE TABLE `city` (

`id` bigint(11) NOT NULL AUTO_INCREMENT,
`name` varchar(100) NOT NULL,
`code` varchar(255) NOT NULL,
`province_code` varchar(255) NOT NULL,
PRIMARY KEY (`id`)

) ENGINE=InnoDB AUTO_INCREMENT=405 DEFAULT CHARSET=utf8;

二、共有元件数据配置如下



1、添加测试计划:页面数据显示默认数值即可无需更改





文件 编辑 查找 运行 洗项 帮助	
📑 🏶 🤐 🔛 👗 🕞 🕒 + 🗕 🍫 🕨 🔊 (D 😡 🞯 🍘 🧆 🏷 📰 🛛 🏎
日 日 日 日 日 日 日 日 日 日 日 日 日 日	测试计划 名称: 數据库测试计划
	注4年:
	🔲 独立运行每个线程组(例如在一个组运行结束后启动下一1
	☑ 主线程结束后运行 tearDown线程组
	■ 函数测试模式 □ 左半位素面に見てくませり取る際取得的教授型(立件) □ 右半位素面に見てくませり
	六月ヨ(水帝要记末尊下隋水州服方器取得的颜谓到义扞时 选择这个选项很影响性能。
	添加目录或jar包到ClassPath 浏览 删除 某
	库

2、添加线程组:页面数据显示默认数值即可无需更改

· · · · ·	
 ● 数据库测试计划 ● 数据库测试计划 ● JDBC Connection Configuration ● JDBC Request □ JDBC Request □ 系看结果树 	线程组 名称: 线程组 注释:
	在取样器错误后要执行的动作
	۲
	《 我程属性
	线程数. 1
	Ramp-Up时间(秒): 1
	循环次数 🗆 永远 1
	□ 延迟创建线程直到需要
	□调度器
	持续时间(秒)
	启动延迟(秒) よこうとしいり
	牧 午 测 诋 网

3、添加配置元件——JDBC Connection Configuration 数据库连接配置元件

Variable Name Bound to Pool: test

Connection Pool Configuration 、 Connection Validation by Pool: 默认值

Database Connection Configuration (以 MySQL 数据库为例):

Database URL: jdbc:mysql://ip:3306/库名?characterEncoding=UTF-8&allowMultiQueries=true

JDBC Driver class: com.mysql.jdbc.Driver

Username: hjphXXX







Password: XXXXX

注意:

A. 【allowMultiQueries 设置为 true,就可执行多条 sql 语句,参数与参数之间连接需要使用 "&"符号。使用场景:如果执行多条 SQL 语句就必须要加这个字段,否则会报错;执行一条 SQL 语句可以不用加这个字段】

B.【characterEncoding=UTF-8: 链接的数据库的编码格式,没有设置该字段,将不 会转译中文字符】

- 1 新規度測法注制			
◎◎线程组	JDBC Connection Configuration		
IDBC Connection Configurat	名称: JDBC Connection Configuration		
└⊿ 察看结果树	注释,		
	Variable Name Bound to Pool		
	Variable Name for created pool: test	值test即为:JDBC Request 引用的变量	
	Connection Fool Configuration		
	Max Number of Connections: O		
	Max Wait (ms): 10000		
	Time Between Eviction Runs (ms): 60000	直接使用默认值 , 不需要更改	
	Auto Commit: True		
	Transaction Isolation: DEFAUL	1	
	Connection Validation by Pool		
	Test While Idle: True		
	Soft Min Evictable Idle Time(ms): 5000		
	Validation Query:		
	Database Connection Configuration		
	Database URL: jdbc:mysql://	/test?characterEncoding=UTF-8&allowMultiQueries=true	
	JDBC Driver class: com. mysql.jdbc.Drive:		
	Username: hjh;	链接数据库需要填写的信息,注意信息需要填写正确才能连接数据库	
	Password:		

4、添加取样器——JDBC Request

Variable Name Bound to Pool : test

Query type: 根据不同的 SQL 类型进行设置

其他字段根据实际情况进行填写

5、添加察看结果树

请求为绿色表示执行成功,响应数据显示1updates 即为一条更新数据执行成功

● 数据年週 (瓦丁 刻) ● 数据年週 JDBC Connection Configuration JDBC Request ○ 該看容無期	察看结果树 名称: 際瘤结果树 注释: 所有数据写入一个文件 文件名 重批:
---	---

请求为红色表示执行失败,根据提示修改自己配置的数据,SQL 直到成功





JDBC Connection Configuration JDBC Request	※有112.米内 名称: 黎看结果村 注释:			
红色为井石生物	所有數据写入一个文件 文件名		82. 显示日志内容:	
	亚抗: Text: ② DDC Report	Control and an and an	博为峰旗下	Find PEtros FISTER
		To have a serie is yie 38L bother: dask the small that correspond to yies \$55L series version for the right spatial to use a series and the small series are small series and the small series are small series and the small series and the small series and the small series are small series and the small series are s	into city (page, code, province_code) values 软件狈	」试网
		1		

- 三、执行增删改查操作
- 1、单个 insert 语句操作

共有元件及字段不变

变更: JDBC Request-Query Type 配置: Update Statement

SQL: insert into city (name,code,province_code) values ("北京市","500000","520000");

■ 数据库测试计划 ● 数据程组 一类 JDBC Connection Configuration	JDBC Request
TDBC Request 家看结果树] 王明:
	Variable Name Bound to Pool
	Variable Hame of Pool declared in JDBC Connection Configuration: test
	SQL Query
	Query Type: Update Statement
	Query:
	1 insert into city (name,code,province_code) values ("北京市","500000","520000");

执行结果:

 ● 数据库测试计划 ● 线程组 ● JDBC Connection Configuration ● JDBC Request ● Connection 	察看结果树 名称: 察看结果村 注释: 所有数据写入一个文件 文件名 重挑: □ ≤+5×5 □ = 55元=55 重挑 重置	前足
[East ● 取样器线集 请求 师应对据 ● 1000 Request Response Body Response headers 1 1 updates 执行成功	

2、多个 inset 语句操作

共有元件及字段不变

变更: JDBC Request-Query Type 配置: Prepared Update Statement

SQL:

insert into city (name,code,province_code) values ("城市 1","510000","512023"); insert into city (name,code,province_code) values ("城市 2","510000","512023"); insert into city (name,code,province_code) values ("城市 3","510000","512023");





■ 数据件砌取TXI ● ③ 线程组 □ ★ IDBC Connection Configuration	JDBC Request	
DBC Request 系看结果树	AMN: JUBE Kequest 注释:	-
	Variable Name Bound to Pool- Variable Name of Pool declared in JDBC Connection Configuration: test	
	SQL Query Query Type: Prepared Update Statement	
	<pre>1 insert into city (name,code,province_code) values ("城市1","510000","512023"); 2 insert into city (name,code,province_code) values ("城市2","510000","512023"); 3 insert into city (name,code,province_code) values ("城市3","510000","512023");</pre>	

查看执行结果: 三条数据更新成功

■ 数据库测试计划 300 线程组 JDBC Connection Configuration JDBC Request 派者结果树	察看结果树 名称: 察看结果树 注释: 所有数据写入一个文件 文件名
	1 updates. 1 updates. 1 updates. 1 updates. 文字 任 训 试 双

3、单个 Delete 语句操作

共有元件及字段不变

变更: JDBC Request-Query Type 配置: Update Statement

SQL1: delete from city where id=394;

- 1 数据库测试计划 - ④ 线程组 JDBC Connection Configuration TDBC Request 察看结果树	JDBC Request 名称: IDBC Request 注释:
	Variable Name Bound to Pool Variable Name of Pool declared in JDBC Connection Configuration: test
	SQL Query Query Type: Update Statement
	1 delete from city where id = 394;

执行结果:

 ● 数据库测试计划 ● 数据库测试计划 ● 3DBC Connection Config ● JDBC Connection LDBC Request 	察看结果树 名称: 察看结果树 注释: 所有数据写入一个文件 文件名



4、多个 delete 语句操作 共有元件及字段不变 变更: JDBC Request-Query Type 配置: Prepared Update Statement 方法一:利用 where 语句 SQL1: delete from city where id in (388,399,320); 可以实现批量删除 SQL2: delete from city where id > 381; 可以实现批量删除 方法二:利用 like 语句 SQL: delete from city where name like "% 五家%"; 方法三:利用 between and 语句 SQL: delete from city where id between 385 and 390; 方法四:执行多条 SQL 语句 delete from city where id =358; delete from city where id =342; delete from city where id =210; delete from city where id =350; 执行即可

5、Update 操作

Update 的操作和 Delete 的操作是一样,所有的配置以及配置的值都与 Delete 相同,只是 SQL 不同

SQL1: update city set name ="测试 city" where ID = 381;

 ■ 数据库测试计划 ● ③ 线程组 JDBC Connection Configuration IDBC Request 家看结果树 	JDBC Request 名称: JDBC Request 注释:
	Variable Name Bound to Pool Variable Name of Pool declared in JDBC Connection Configuration: test SQL Query Query Type: Update Statement 1 update city set name ="测试city" where ID = 381;
执行即可	



6、单个 select 操作

共有元件及字段不变

变更: JDBC Request-Query Type 配置: Select Statement

SQL1: select * from city where id >358;

SQL2: select * from city where id = 342;

/ 数据库测试计划.jmx (C:\Users\Administrator\Desktop\数据库测试	+13[jmx] - Apache JMeter (5.0 r1840935)
文件 编辑 查找 运行 选项 帮助	
📄 🏶 👜 🔜 👗 🗊 📵 🕂 — 🍫 🕨 🔈 🌒) 🐵 🞯 🎬 🗛 🏷 📰 🔞 💩
□ 】数据库测试计划 □ ④ 线程组	JDBC Request 名称: JDBC Request
□ 梁看结果树	注释:
	Variable Name Bound to Pool
	Variable Name of Pool declared in JDBC Connection Configuration: test
	SQL Query
	Query Type: Select Statement
	Query:
/ 数据库测试计划.jmx (C:\Users\Administrator\Desktop\数据库测试	11월jmx) - Apache JMeter (5.0 r1840935)
文件 编辑 查找 运行 选项 帮助	
🖹 🕼 🤮 🛃 👗 🗊 🛄 🕂 — 🗇 🕨 🕅) 🔘 🖓 💥 🗛 🏷 📳 📓 🚳 00:00:01 🕰 0 0/1 🚱
➡ 数据库测试计划 → ④ 线程组	察看结果树
JDBC Connection Configuration JDBC Request	名称: 察看结果树
□ □ ◎ 原着結果树	注释:
	所有数据写入一个文件
	文件名 刻葉 显示日志内容: 目 仅错误日志 目 仅成功日志 配置
	博力峰旗下
	Text ▼ 取样器结果 備求 响比数据
L L	Kesponse Body Response headers

7、多条 SQL 语句同时查询

共有元件及字段不变

变更: JDBC Request-Query Type 配置: Callable Statement

方法一: SQL:

select * from city where id = 358;

select	*	from	city	where	id	=	342	,
--------	---	------	------	-------	----	---	-----	---

- 1 数据库测试计划 - ③ ④ 线程组	JDBC Request 名称, JDBC Request											
JDBC Connection Configuration CSV 数据文件设置												
BJDBC Request □ 察看结果树	注释.											
□> <u>JDBC Request</u> □	Variable Name Bound to Pool											
Variable Name of Pool declared in JDBC Connection Configuration: test												
_	SQL Query											
	Query Type: Callable Statement Query:											
	1 select * from city where id = 358;											
	<pre>2 select * from city where id = 342;</pre>											

执行结果:在此场景下 Callable Statemen 会显示多条查询结果如下图,而 Prepared Select Statement 值显示一条查询结果



《51 测试天地》五十七(下)

www.51testing.com



 ■ 数据库测试计划 ● 数据库测试计划 	察看结果树
JDBC Connection Config	名称: 察看结果树
□□ 🛃 察看结果树	注释:
	所有数据写入一个文件
	文件名
	A T
	查找: 区分大小写 区 正则表达式 查找 重置
	▲▼ Text ▼ m光照往用 法世 顺应渤据
	AXTEGRAT HAY THERE
	Response Body Response headers
	Fin
	1d name code province_code
	410 Ces 22 33
	id name code province_code
	409 ces 22 55
	Output variables by position:

方法二: Select 查询简单参数化

JDBC Request 配置:

Query Type: Callable Statement

Parameter value: 410,411

Parameter types: integer, integer

select * from city where id = ?;

select * from city where id = ?;

第一个问号代表: 第一个参数值

第二个问号代表: 第二个参数值

□-▲	JDBC Request									
JDBC Connection Config	名称: JDBC Request									
□ ² 察看结果树	注释:									
	Variable Name Bound to Pool									
	Variable Name of Pool declared in JDBC Connection Configuration: test									
	SQL Query									
	Query Type: Callable Statement									
	Query:									
	1 select * from city where id = ?;									
	3									
	博为峰旗下 (博为峰旗下)									
Γ	51 toct pa									
	Parameter values: 410, 411									
	Parameter types: integer									
	Result variable name:									





执行结果:

www.51testing.com

□□-▲ 釵掂库侧试计划 □-④ 线程组	察看结果树
JDBC Connection Confi	· E 名称: 察看结果树
□ ▲ 察看结果树	注释:
	所有数据写入一个文件
	文件名 刻足 显示日志内容: 『
	查找: 区分大小写 正则表达式 查找 重置
	Text ♥ 取样器结果 请求 响应数据
	TDBC Request Response Body Response headers
	id name code province_code 410 ces 22 55
	id non ada muninas asta

方法三: CSV 进行多条 select 语句查询

使用场景:同一类型的数据,需要查询多条

SQL1: select * from city where id = ?;

配置文件数据展示如下:



每一个配置文件设置如下:

线程组:

📄 🏶 🚔 🔚 👗 🗊 🗊 🕂 🗕 🍫 🕨 》 🐌 🍩 🗐	🎬 🎮 🏷 📰 🛐 🏎
□ 数据库测试计划	线程组
JDBC Connection Configuration 街环均制器	名称: 线程组
	注释.
🗆 🥕 JDBC Request	本即样哭错得后更地行的动作。 ————————————————————————————————————
── 🔏 祭有结果树	在极性都在庆后安秋时间401F
	53/1王/時/1王
	线程数: 1
	Ramp-Up时间(秒): 1
	循环次数 🗌 永远 1
	🔲 延迟创建线程直到需要
	□ 调度器
	调度哭配置
	持续时间(秒)
	启动延迟(秒)





《51 测试天地》五十七(下)

www.51testing.com

■▲数据库测试计划	TBBC Connection Configuration						
一次IDBC Connection Configure	A Sec. JDBC Connection Configuration						
→ CSV 数据文件设置 ■ > IDBC Request	注释。						
▲ 察看结果树	Variable Name Bound to Pool						
	Variable Name for created pool:test						
	Connection Pool Configuration						
	Max Number of Connections: 0						
	Max Wait (ms): 10000						
	lime between Eviction Runs (ms): 60000 Auto Comit: Turue						
	Transaction Isolation: DEFAULT						
	Connection Validation by Pool						
	Test While Idle: True						
	Soft Min Evictable Idle Time(ms): 5000						
	Validation Query:						
	Database Connection Configuration Database [N::doc:wssl://rm=mben01a1rv5it3d01co.mrsol.rds.alivuncs.com:3306/test?characterEncoding=UTP=SkallowWultiQueries=true						
	parases was juoring cir. The memory of the prime of the structure of the s						
	Username: hjhp8888						
	Password:						
环控制器:	Paawood: •••••••••						
环控制器:	Pasword:••••••••••••••••••••••••••••••••••••						
环控制器: 数据库测试 数据库测试 数据库测试 	Parmord:						
环控制器: 	T计划 Connection Configuration 創器 数据文件设置						
环控制器:	T计划 Connection Configuration 制器 数据文件设置 C_Request						

CSV 数据文件设置:

● 「数据面割试计划 ● 使性性 一下 的比较易器 ● 2000 Founction Configuration ● 2000 Foundation ● 2000 Foundation ● 2000 Foundation	CSV 数据文件设置 名称, 文件 注释, 设置 CSV 数据文件 文件 + 文件++	选择你本地的cs ::[C:/Users/Administrator/Desktop/课纸.exv	v文件即可
	文件職業 支量名称(石文星中间隔 忽略首行(只在设置)支量名称后才生效 分隔符(用)\tr(代當制表符 是否允许参引导) 遇到文件结束符再次循环 通到文件结束符件止线程 线程共享集;	Id csv文件内的参数名称,参数间以 True False False False False	英文逗号隔开显示eg:id,name,coge _{来,③非国本} 。 ·

引用的 CSV 文件:

文件	・ 5・ (开始	⇒ - !	⊽ 页面布局	公式	数据 〔	审阅 视图	1							测试,csv	- Excel				
	👗 剪切	宋体		× 11	· A A	= = =	87 -	📴 自动换行	常规		*			常规	着	3	好	适中	
粘貼	■ 复制 ・	p 7	n _ L 🖙 _ L	A . A	wén		z= 3=		· · · ·		€.0 .00	条件格式	委用	检查单元格	解释性文	**	警告文本	链接单元:	格
*	🞺 格式刷		<u>∎</u> , ,	<u></u>	, x		12. 72.			0	.00 ->.0	*	表格格式 -		/	· /			-
剪则	版。		字体		Fa		对齐方:	π,	G.	数字	G.			/		样式	t		
G11	·	: ×	√ fx		ل بلد حک با		-												
一		俱与	耑安伐	用的	一参数	们省和	¥ ,	G	Н		I	J	K		M	N	0	P	
1	id																		
	1													/					
- ⁶	2																		
# 5	4	- II	ats - 1.				1 L / ±							/					
6	5	M		テナナダ	育項与	「参数	的旧												
7	6												1						
В	7																		
9	8												之件样	各式为:	CSV				
10	9								-			-	~!!!		逋 为	峰旗"	<u>к</u>	\frown	
1	10														1020				
12	12														-51	t o	CF		7
- 0	12														J	LC	DL		F
15																			-
16															47	化生	STIL	iff XX	
17															七	IT	1753	Mrc hal	

JDBC Request:





■ 素質を通知状理 ● ACTION ● ACTION	JBC Report 名称, JBC Request 注释。	
	'Variable Name Bound to Pool Variable Name of Pool declared in JUBC Connection Configuration: [test SGG Query Query Type: [Prepared Select Statement Query:	8 4
	T stlet * frei dty der i 4 - 1	
	引用CSV文件的参数,写法:\${id}	
	Prosector values/5164	130
	Variable name: parameter values: \${id};\${name Result variable name: 道写参数类型,一个参数对应一个类型,多个参数写法eg: parameter types:integer,string Qurry timeot (b): Bedle ResultSvt: Store as String	9

执行结果:



以上部分就是关于 Jmeter 对数据库增删改查操作的具体方法, Jmeter 对数据库的操 作灵活多变,方法较多,并没有统一的方法,达到最终的目的即可。





疫情之下, IT 从业者都做了些啥?

◆ 作者:咖啡猫

缘起

从古至今, 似乎每一个庚子鼠年都是那么难熬: 1840 年鸦片战争、1900 年八国联 军进京、1960 年前后的三年自然灾害, 2020 年的新冠病毒肺炎疫情......我想,这个庚子 鼠年春节会让我们每一个人终生难忘, 从 2019 年 12 月底至今, 我们经历了太多太多事 情, 有的让我们惶恐、有的让我们愤怒、有的让我们感动、有的让我们坚强、有的让我 们欣慰。终于在全体华夏儿女的共同努力下, 时至 2020 年 3 月中旬国内疫情得到了基 本控制, 很多省份逐渐恢复正常秩序, 从四面八方来援鄂的医疗队开始分批返回家 乡.....

在这场没有硝烟的战争中,有无数人在为湖北拼命,笔者作为一名湖北人,真心的 感谢每一个帮助过湖北的"抗疫战士",希望他们返程一路顺风、一生平安喜乐。他们 当中有 30 后的钟南山院士、40 后的李兰娟院士、50 后的郑杨教授、60 后的张定宇院 长、70 后的退伍军人刘森波、80 后的快递小哥汪勇、90 后的派出所民警赵闯、95 后的 护士朱海秀、00 后的护士刘家怡.......

所行至此所遇皆师! 笔者今天想从身边的 ITer (IT 从业者)的角度出发,来写写 疫情之下身边普通 ITer 都做了些啥?

人物: X 工 所在地: 湖北武汉 职业: 测试工程师

抗疫箴言: "从互联网测试变志愿者司机,谢谢老婆大人支持"

X 工是笔者的同事, 土生土长的武汉人, 是武汉一家互联网公司的测试工程师。我 印象中的 X 工是一个工作认真、性格豪爽的武汉伢, 经常因为一个 BUG 和开发、产品 争得面红耳赤, 也会在熬夜加班后, 顺路开车送送同组的小伙伴们回家......





2020年1月23日10时起,武汉全市的公交、地铁、轮渡、长途客车停运。1月26日起,武汉中心城区机动车禁行。这一系列操作,都是为了降低普通市民的感染风险,毕竟在没有特效药的情况下,隔离是最好的办法。

今天10时起,武汉全市公交地铁等停 运,机场火车站离汉通道关闭

央视新闻 1月23日

据武汉市新型冠状病毒感染的肺炎疫情防控指挥部 消息: 自2020年1月23日10时起,全市城市公 交、地铁、轮渡、长途客运暂停运营;无特殊原 因,市民不要离开武汉,机场、火车站离汉通道暂 时关闭。恢复时间另行通告。

通告全文如下:

为全力做好新型冠状病毒感染的肺炎疫情防控工作,有效切断病毒传播途径,坚决遏制疫情蔓延势 头,确保人民群众生命安全和身体健康,现将有关 事项通告如下:

自2020年1月23日10时起,全市城市公交、地铁、 轮渡、长途客运暂停运营;无特殊原因,市民不要 离开武汉,机场、火车站离汉通道暂时关闭。恢复 时间另行通告。

恳请广大市民、旅客理解支持!

武汉市新型冠状病毒感染的肺炎疫情防控指挥部 2020年1月23日 武汉: 26日起,中心城区机动车禁行 央视新闻 1月25日



武汉市新型肺炎防控指挥部通告

(第9号)

为控制人员流动引发的传染风险,现就城区交通管 制事项通告如下:

自2020年1月26日0时始,除经许可的保供运输 车、免费交通车、公务用车外,中心城区区域实行 机动车禁行管理。市民确有通行需求的,按指挥部 第8号通告执行,各社区配备足够车辆保障需求。



"公共交通暂停运营,私家车不让上路,此时很多医护人员上下班十分困难,为了 让医护人员顺利上下班,节约更多时间抢救病人,欢迎有车的司机们加入志愿者行列, 负责接送武汉的医务人员上下班,通行证统一办理,有意者请加群"1月28日这样一条 朋友圈的消息映入X工眼帘。起初,X工是有些犹豫的,毕竟自己也是到了上有老下有 小的年纪,又没有医学相关的知识背景,万一不幸感染再传给孩子怎么办?于是,他抱 着观望的心态加了志愿者群,当他看到越来越多的人入群报名加入志愿者行列,内心很 受触动,他在没有告诉家人的情况下报了名。家国情怀、艰苦奋斗、守望相助的精神也 许就是这样一传十、十传百、百传千、千传万地发扬光大。

X 工告诉笔者,他通过朋友关系自掏腰包买了成套的防护设备:防护服、手套、护 目镜、口罩、消毒液,每次出门前和回家前都会进行消毒。每天早晚接送医护,其他时 间会帮助各个小区运送蔬菜等必要的生活物资。我问 X 工:"你不怕么?"X 工说:"怕 有什么用呢?我今早送一个医生去金银潭医院,聊起疫情真的很沉重,那个医生是刚毕 业的医学生,说着说着就哭了......唉,大伙都不容易,我作为武汉人也出份力,帮帮自 己的家乡。但是接送医护的事我一直没敢和家里人说,怕他们担心,最近都是在外面吃





泡面……""X工,我要给你点赞!""不聊了啊,马上去武昌拉物资了"X工匆忙下线, 开始一天的忙碌……可见,不管是做测试还是当司机,不管是按月拿工资还是纯义务劳动,对X工而言变化的只是岗位而已,不变的是敬业精神。

就在刚刚,我看到 X 工的朋友圈晒了老婆给做的红烧肉和烧茄子,并配文:"从互 联网测试变成志愿者司机,谢谢老婆大人默默支持!"我想这大概就是普通人的家国情 怀和幸福模样吧。

人物:H老师 所在地:湖北咸宁 职业:IT教师

抗疫箴言:"谁叫我是女性平权者!我的敏感度就是高一些。"

H老师是笔者的大学同学,老家在湖北咸宁。她研究生毕业后就留在家乡一所学校 做 IT 类专业课的老师。在笔者印象里,H老师一直是个时尚精致、特立独行的姑娘。 她喜欢肖战,为了听爱豆的演唱会可以花大半个月工资穿越几个城市去听;她爱吃海底 捞,每个月都得去撮上一顿;她热衷女性平权,周末都会去参加相关的读书会和论坛, 用她自己的话说就是:希望能早日消除性别歧视;她还是组织活动的高手,大学时就是 学校社团的"风云人物",学校各类文艺晚会都少不了她的身影......



1月23日,当她看到武汉协和医院官微上寻求社会捐赠医用物资的消息时,H老师 在校友群里说:这么多医护人员援助湖北,其中有不少是女性,我们应该做点什么,除 了捐赠口罩消毒液等常规医用物资之外还可以捐赠一些女性卫生用品。一石激起千层





浪,"我可以在国外负责采购医用物资和卫生用品""我有国际物流公司的朋友,可以联 系运送物资""我们可以开始在校友圈筹款""我可以负责联系医院""我来写公众号, 做好捐赠款项物资的公示及宣传工作"……就这样,H老师连夜做了策划方案,将人员 分成了几组:采购组、筹款组、物流组、外联组、宣传组等……

筹款组开始在校友群、公众号、朋友圈筹款,大家一呼百应,你五十我一百地捐 着,每一笔捐赠都会详细记录捐赠时间、捐赠人、捐赠金额、转账截图等信息;采购组 拿到筹集的资金,开始在国外街头巷尾的药店超市采购医用物资和卫生用品,每一笔交 易都开具发票;外联组联系需要捐赠的医院,获取收货地址和联系人信息;物流组组织 发货运送物资;宣传组定期在校友会的公众号上公开捐赠信息......大家时不时在校友群 中汇报各自进展,那劲头仿佛回到学生时代大家准备迎新晚会一样。最终,医用物资和 卫生用品送到了各大医院的医护人员手中,校友会也收到了若干捐赠证明。有医护收到 女性卫生用品时,直呼真的太贴心了。

我问 H 老师: "你脑洞可真够大的,除了捐赠常规医用物资,居然能想到女性卫生用品。" H 老师答: "谁叫我是平权者! 我的敏感度就是高一些。"

人物:L叔 所在地:湖北黄冈 职业:电商个体户

抗疫箴言: 我是共产党员, 我不能发国难财!

L 叔是我一个远房亲戚,早年当过兵,退伍后就在家开了个农村淘宝店营生。印象中,L 叔家境并不富裕,L 叔的父母早已年迈,父亲更是瘫痪在床需要人照顾,L 叔的 老婆在家务农,儿子在武汉读大学,女儿在读高中。尽管家境不富裕,但是凭借着正直 勤劳的品格,他做买卖从不缺斤少两也不会坐地起价,所以L 叔在街坊中口碑极好,很 多人都愿意去他店里买东西。

随着湖北各个地市开始封城,生活物资开始变得十分紧俏。有些地方的猪肉价格甚 至涨到了 50 元一斤,还很难买到肉......笔者周围很多朋友都吐槽到: 宅家一个月,半点 荤腥不沾,反倒瘦了好几斤......



《51 测试天地》五十七(下)

www.51testing.com



湖北日報 1 有温度更有态度

2月29日下午,湖北省新冠肺炎疫情防控工作指 挥部召开第35场新闻发布会。



发布会上,武汉市人民政府副市长徐洪兰介绍称,前段时间正好是春节期间,再加之疫情, 带来人工成本、物流成本大幅上涨。现在的物 流成本和人工成本是平时的3倍左右,这样直接 推动了菜价、肉价的上涨。针对如此情况,武 汉市积极推动多渠道保障物资供应,企业积极 参与,大型商超带动,共同发出倡议降价惠 民,让老百姓购物实惠。 软件测试网

某天,社区几百人的大群里,传来L叔的消息:"乡亲们,从明天开始我们店供应 猪肉,由志愿者集中采购,28元一斤。我是共产党员,我不能发国难财!欢迎有需要的 乡亲定购!"引来社区群里点赞声一片......除此之外,L叔还报名了志愿者,负责在社区 路口执勤,测量体温,搬运物资,上报数据等工作......一个普通的小店主,尽管自己并 不富裕,但是疫情面前不忘共产党员的初心使命,不发国难财,积极参与防疫工作,那 一句"我是共产党员,我不能发国难财"深深感动了笔者......

后记: 致敬亿万普通人







其实,除了我上文列举的三位主人公,还有很多很多的 ITer 加入到抗疫行列中 去。他们有火神山医院的 IT 信息化建设者;有自己用电脑建模仿真疫情传播过程,然 后制作视频,提醒公众要宅家抗疫最安全的软件工程师;有开发口罩人脸识别系统的算 法工程师......哪怕由于专业所限你没法去医院第一线医治病患,你只是响应国家号召, 宅家自我隔离,按时上报体温,不造谣不信谣不传谣,为身边的人传播正能量,在国务 院官方平台上反映问题,在各种正规渠道捐款捐物奉献爱心,这就是在为我们的国家做 贡献。最后,借用人民日报2月25日的一篇文章标题做结尾:英雄的城市、英雄的人 民,献给疫情防控斗争中的武汉人民。致敬全国亿万普通人,是你们的爱心、信心、决 心连成一片,才让我们的国家转危为安,你们才是历史的缔造者!

■Python 全栈接口测试实战精英训练>>https://dwz.cn/txAURE7A







当你打开这篇文章,相信很多小伙伴都是做功能测试的,不知道有没有过这样相似的情景,比如编写测试用例漏点,同样的功能旁边的小伙伴能写出多余自己好几条的 case;或者在参与敏捷测试过程中,由于时间仓促某些功能点覆盖度不全,导致线上问题遗漏等等。

不知同样作为测试的你,有没有思考过究竟问题出现在哪里呢?

我相信很多同学都看了市面上和测试相关的书籍,也看过很多测试大咖的博客,培 训课程也多多少少参加过,就是发现在实际应用的过程中在大脑里提取不到,在这里我 提出2点自己的见解

一、测试知识点零散,没有形成自己的架构

1、从功能测试来讲,给你某个应用让你写 case,这里有个原则列举所有可能发生的"正确的、错误的、异常的"场景,下面我们来看看这一原则的架构:

序号	测试点	分散
1	安装	首次安装
		復盖安装(使用甲、非使用甲)
		不同的操作系统(不同的补丁)安装
2	卸载	卸载后遗留文件
3	更新	近版本更新
		老版本更新
		强制更新

功能测试架构:



《51 测试天地》五十七(下)



www.51testing.com

4	正常功能	所有流程点一遍
		所有按钮、输入框正常异常判断
5	异常	输入错误数据
		进行异常流程
		断网
		app 切换后台
		快速多次点击呆个功能
		问一个切能多次裸作
		网个于指分别问时点击返凹按钮及功能键
6	不同网络	- 北
0		nat 与非 nat
7	UI 测试	各功能 UI
8	兼容性	电脑:不同浏览器不同系统不同补丁
		手机: android 、ios 系统主流手机
9	竞品测试	不同产品之间功能比对
10	易用性	产品符合用户习惯
11	可维护性	产品更新和维护
12	文档测试	帮助文档、帮助信息、
13	性能和压力测试	不同负载情况下
14	稳定性测试	长时间运行软件是否稳定
15	负载均衡测试	A\B 服务请宕机
16	冲突测试	防病毒软件
		防火墙
		同类产品

二、不会学习,形成"一看就会,一用就废"现象

何为学习,学习是在特定情境下由于练习和反复经验而产生的行为或行为潜能的相 对持久的变化。从上面的测试架构来说,读完后感想"哦,原来是这样啊" 然后就没 有然后了,第二天回到单位继续按照原有的知识架构进行工作,头一天看的文章内容 随云彩飘走了。

我们都知道学习是会遗忘的,德国心里学家有个著名的艾宾浩斯遗忘曲线大家都应





该不陌生,大致的意思是学习的内容仅仅看过一遍1天后能记住一部分50%,一周后、一个月后也就剩20%了。

所以我在这里的建议是把核心知识架构进行背诵,这样无论什么样的功能拿到你面 前你都知道要测试什么了。





全栈测试:平衡单元测试与端到端 测试 ◆ 译者: 咖啡猫

全栈开发人员的核心竞争力是端到端地交付并发布一个特性的能力。教程和书籍通 常关注建立全栈开发和测试工作所需的管道(我的教程和书籍将 Angular、Rails、 Bootstrap 和 Postgres 结合在一起)。通常会缺少如何通过 WEB 开发栈来测试应用程序的 指导。下面,让我们在这篇文章中深入探讨研究一下这一问题。我们将学习如何让端到 端测试得到更多的产出,包括关于测试内容和如何保持测试的可靠性和易维护性的指 导。我们也会接触单元测试并认识它们在我们端到端测试策略中所扮演的角色。但是首 先,让我们来完全理解编写测试代码的目的。

测试目的的核心在于,测试能够确保应用程序正在执行你希望它执行的操作。它们 是一个自动化的脚本,用于执行代码并检查代码是否达到了预期效果。测试代码编写的 越好,你就越可以依靠它们为你的后续部署把关。如果你的测试代码不合格,你要么需 要一个 QA 团队,要么就要发布有缺陷的软件(两者都意味着你的用户获得价值的速度 比理想情况下要慢得多)。如果你的测试代码编写的很好,你可以放心地快速发布软 件,无需批准,也无需像 OA 这样缓慢的手动过程。

你一定要保证你编写的测试代码的未来可维护性。你的应用程序会更改,所以测试 代码也会随之更改。理想情况下,你的测试只需要与你在软件中所做的更改成比例地进 行更改。如果要更改错误消息,则不需要重写大量测试套件。但是,如果你正在完全改 变一个用户流,那么有理由期望重写许多测试。

实际上,这意味着你不能将所有的测试都作为端到端的完整集成测试来完成,但是 你也不能将其作为一个非常小的单元测试来完成。本文就是探讨关于如何达到两者之间 的平衡。



一、测试的类型



实际上,根据不同的划分标准,我们能得到无数种测试类型。但是根据本文的写作 目的,我们仅仅讨论两种测试类型:端到端测试和单元测试。

端到端测试是一种模拟用户行为的测试。在 web 应用程序中,他们将启动服务器、 启动浏览器、单击鼠标,并断言浏览器中发生的某些事情会让我们确信我们的功能正在 工作。这些测试给人很大的信心,但它们速度慢、脆弱,并且与用户界面紧密耦合。

单元测试根据它们的公共 API 来执行代码单元。这些测试包括创建一个类的实例, 并使用特定的输入对其调用方法。断言你调用的方法具有预期的效果(通常它们返回预 期的输出)。这些测试快速、稳定,并且没有与系统的其他部分紧密耦合。但是,它们 并不能让你确信整个系统正在工作,而只是测试代码单元正在工作。

构建特性的工作是在这两种测试之间找到适当的平衡。如果你有太多的端到端测 试,那么将来对应用程序的更改将是痛苦而缓慢的。如果你的端到端测试代码太少,即 使快速测试用例具有 100%的代码覆盖率,细微的 bug 仍然会渗透到生产环境中。

二、从用户体验出发

你的软件是为某些用户服务的,所以应该由这些用户的需求来驱动你的工作。我不 建议使用测试来设计用户体验,所以在编写测试之前,要弄清楚用户将如何使用软件(通 过实验性编码或与设计人员一起工作)。一旦你有了这些,就开始工作吧。

理想情况下,你将为部分用户体验创建端到端测试,并编写代码使其通过。在编写 代码时,你将创建单元测试,以具体化需要创建或修改的代码的细节(通常是后者)。

问题在于在没有用户接口产品(HTML)参考引用的情况下,很难编写一个失败的端到端测试用例。原因是因为大部分端到端测试的形式如下:

1、在页面上找某个元素

2、用某种方式和它交互

3、验证交互是 OK 的

4、重复操作过程直到测试结束

这就意味着你需要一些与之进行交互的用户接口元素的细节(比如一些 DOM 对象)。当你考虑到由 JavaScript 支持的交互设计时,如果没有实际构建接口(至少部分构


建),就更难做到这一点。

为了解决这一问题,你需要大致了解 UI 在浏览器中如何工作。使用固定的数据, 而不考虑其它流程---次只关注一件事。它运行起来后,你就可以开始工作了。

这样做的时候,你需要考虑两件事:这个特性应该被测试么?如果应该,又该如何 测试呢?

三、你应该测试它么?

虽然在编程中没有捷径可言,但是用户在代码中经历的路径要比可能的少得多。例 如,当用户购买产品时,我们可能会根据用户的地址、选择的送货方式或以前的购买历 史而有不同的处理方式。在这些情况下,用户体验都是相同的,所以从用户的角度来看 这是一个流程。

因此,你的目标是测试所有用户流。你需要一组测试来模拟用户做你希望做的事 情,并断言你希望用户拥有的所有体验都是正常工作的。

既然你知道要测试什么,那么应该如何着手呢?

四、如何进行端到端的测试

如果要修改用户流,请修改该用户流相关的测试。由于端到端测试模拟用户活动, 所以你不需要对你想要断言的每件事都进行一次测试。如果用户应该在结账屏幕上看到 三个重要的信息片段,那么你就不需要三个测试——个测试就可以检查所有三个信息。 因此,在修改现有用户体验时,寻找可以更改的现存测试,以避免过多的重复性劳动。

否则,你将需要一个新的测试。请记住,你的目标是模拟用户会做什么。务必要对 如何组织测试中的导航和行为开诚布公。用户真的会直接导航到某个深度链接吗?或者 他们会从一些常见的开始页面点击到他们需要去的地方吗?

这很难做到,尤其是通常要使用最少的标记实现该功能。你的测试需要定位到与之 交互的特定 DOM 元素,要找到所需的精确元素并不容易(或可能)。你需要路标。

路标是插入到 DOM 中的东西,专门用来定位感兴趣的元素。尽早决定这些路标将如何工作。不应该使用用于样式化的 CSS 类来定位 DOM 元素。这样做意味着前端开发 人员将通过更改类名来破坏测试。你也不应该使用 JavaScript 代码(如 js-前缀类)使用的



^{博为峰做下} 51**Lest Ing** 软件测试网

CSS 类或数据属性。这些都会导致同样的危害。 使用前缀为 test- 的 CSS 类或者前缀为 data-test- 的属性是两种常用的技巧: <section class="component dark test-checkout-confirmation"> <!-- ... --> </section>

<section class="component dark" data-test-checkout-confirmation>

<!-- ... -->

</section>

这可能看起来有点讨厌...是的。但是,比起不得不将你的测试代码与内容类/表示类 结合起来,这一点也不让人讨厌。你需要在这里找到一个平衡点---不要盲目地用数据测 试属性标记每个元素。通常,你只需要一点上下文就可以找到元素。例如,如果要单击 某个按钮来购买某个特定的产品,实际上只需要找到包含该产品及其"购买"按钮的某个 元素。

<article data-test-product="1234">

```
<!-- a ton of markup -->
<input type="submit" name="Purchase" value="Purchase">
</article>
<article data-test-product="5678">
<!-- a ton of markup -->
<input type="submit" name="Purchase" value="Purchase">
</article>
```

使用了数据测试产品的属性后,定位产品 1234 的购买按钮,只需要使用一个类似 [data-test-product='1234'] input[type='submit']的 CSS 选择器就可以啦。

这意味着你将不得不修改那些用来支持测试的标记,意味着用户为了获得你提供的体验他需要下载一些不必要的字节。这是一种折衷,但它比测试覆盖率低(这对用户的伤害远远大于 HTML 中额外的几个字节)要好。这种做法是明智的。

当页面上的交互不需要重新加载(即使用 JavaScript)就可以更改内容时,这种技术就 更加重要了。

五、处理交互

当每次单击都重新加载页面时,端到端测试更加可靠,因为底层工具知道要等待页





面重新加载。当用户交互只是简单地改变 DOM 时,就比较困难了,因为没有明显的方法来"等待事情完成"——工具不知道"事情"正在发生什么。

当你的测试需要与不需要重新加载用户行为信息的页面相交互时,你需要一种方法 来等待 DOM 操作完成,然后才开始断言发生了什么。如果不等待,在测试开始断言时 DOM 依然没有更新,这必将导致测试的失败。

就像我们在标记中使用路标来定位要操作的 DOM 元素一样,我们也可以在这里使用它们。任何新的或更改的标记都应该具有某种标记,如果交互失败或没有发生,这些标记就不会出现。换句话说,你不应该在测试中调用休眠来等待 DOM 事件——你的 DOM 应该有路标,测试可以等待的路标。

例如,假设我们想要测试一个动作是否会向用户生成一条成功消息。假设它的实现 方式是发出一个 AJAX 请求,并在调用完成时向 DOM 中插入一条消息。一个基本的实 现可能是这样的:

function purchase(productId) {

\$.post(

```
"/products/",
{ "id": productId }
).done(function() {
    $(".header").html(
        "<div class='alert-success'>Your order was placed</div>");
}).fail(function() {
    $(".header").html(
    "<div class='alert-failure'>There was a problem</div>");
```

});

你可以将测试配置为等待一个 CSS 类 alert-success 的元素出现,然后对其内容进行断言。这意味着,如果页面其他元素使用了那个类,那么测试就会不可靠或被破坏。虽 然你可以将其限制在 HTML 头里,但这只是缓兵之计。





}).fail(function() {

\$(".header").html(

"<div data-test-purchase-failed class='alert-failure'>There was a problem</div>");

});

尽管这会给你的标记增加更多的字节,但它允许你编写一个可靠的测试,能够经受 住一些可视化的改变。只要页面的流程是显示购买成功的消息,可视化实现就可以在不 破坏测试的情况下进行更改。这就是你想要的,这是一种交换。你可以通过创建尽可能 小的、最小的标记来牺牲这种信心,但是这样做的话,你就会浪费时间来修改测试,比 如在视觉效果发生变化时,必须使用手动 QA,或者你只是发布了没有彻底测试过的软 件。

像 Capybara 这样的现代端到端测试工具包含了你需要的所有功能。有一些方法可以 等待 DOM 元素出现后再继续,断言页面特定部分的内容,并与表单元素进行交互。大 多数其他 web 应用程序堆栈都提供了类似的工具。在任何情况下,你都可以将你的测试 库与一个像 PhantomJS 这样的无头浏览器相结合,并且你的端到端测试将会令人惊讶地 快速和可靠。

值得一提的是,如何在分布式环境中实现这一点。

六、当不止一个 APP 时

当你在一个单独的、单片的系统上工作时,上述技术就是你所需要的全部。但是, 如果你在一个分布式的系统中工作,事情就会变得比较棘手。假设你正在处理一个面向 客户的应用程序,但是它必须从另一个系统获取库存数据。你将如何编写此测试呢?

首先,记住要测试的内容。你的端到端测试是测试用户交互。这意味着你的端到端 测试不负责断言远程服务的功能,也不负责断言你的应用程序正在正确地使用该远程服 务。

测试服务的使用(以及这些服务所做的宣传)的最佳方法是使用消费者驱动的契约, 这是单元测试的一种形式(至少在我这篇文章中使用的广义定义中是这样)。

这仍然遗留了如何在端到端测试期间模拟远程服务的问题。你可以建立一个实际版本的服务,但这是不可扩展的。你最终必须管理该服务的内部数据存储以及它所依赖的服务。这是一个难以管理的复杂的问题。





一个流行的选择是在 HTTP 层使用模拟系统。在 Ruby 中, VCR 就是这样一个工具。你将记录为了建立 HTTP 协议与服务端之间的来回报文交互,为了子序列测试能正常工作,模拟系统将在没有网络的情况下回放这些记录。假设你的单元测试中有适当使用服务的测试覆盖率,那么这对于端到端测试来说是很有效的。

另一种选择是建立一个简单的只会返回固定数据的模拟服务。你的 APP 将像往常一样发起 HTTP 请求,但是一个固定的服务会返回一组静态的、已知的数据给你的 APP。 这就需要一些前端配置,但是在与服务端进行简单交互的情况下仍然可以工作。如果你 的应用程序需要在服务中存储状态,并且需要长时间的来回"对话",那么这种技术实 现起来就比较困难了。

我推荐首先尝试 HTTP 层模拟服务,因为它是最简单最快速的。

既然我们已经知道在一个端到端测试中应该测试什么以及如何进行测试,那单元测 试呢?

七、单元测试

回想一下,我们应该测试的端到端标准是用户流。其思想是,虽然系统中有许多可能的逻辑流,但对用户体验产生影响的逻辑流要少得多。单元测试是我们测试其余逻辑流的地方。

这就让我们能够快速可靠地断言系统的大部分正确行为。换句话说,当我们在端到 端测试时断言系统的每一个可能逻辑流时,那是不必要的,也会非常慢和脆弱。

例如,假定一个结账的特性有两个用户流:一个成功的购买和一个失败的购买,此时用户需要再试一次。这就对应着两个端到端的测试。进一步假设在这些情况下,有如下可能性:

• 客户的卡被正确地扣款了。

• 有一个关系到客户银行的问题,但我们想假装它是成功的,然后收费。

• 客户的信用卡被拒绝了。

• 客户的卡过期了。

这是四个流,所以我们需要四个单元测试来断言每个情况都得到了正确的处理。是



的,会有重复的覆盖。我们的端到端测试可能会测试一个成功地扣款,并拒绝处理其它 的用户流,因此,当我们编写单元测试时,我们的覆盖率将超过我们的技术需求。

这是一种折中考虑,但是我们的类被单元测试很好的覆盖是非常必要的。这将容许 他们被移动、重新购买并且更加容易地付款。

关于如何编写单元测试有很多很多的理论,远远超出了我们在这里讨论的范围。我的建议是你采用一种对你来说有意义,容易向别人解释的方法,并始终如一地使用它。

单元测试最困难的部分是决定你的代码中有多少应该被用来作单元测试。这类似于 我们为了测试而向 HTML 添加属性和其他路标——这些构件的存在只是因为我们必须进 行测试。编写单元测试时,你将面临相同的问题。

例如,假设我们的信用卡支付代码被集成在一个叫做 Purchaser 的类中。假设它使用了一个第三方提供的接口 AwesomePayments 来做实际地支付。

class Purchaser

```
def charge(purchase)
   AwesomePayments.charge(purchase.customer.id,purchase.amount)
rescue => ex
   try_again_later(purchase.id)
end
```

...

end

这很清楚,也很有意义,在一个没有单元测试的世界中,这可能是最理想的设计。 然而,为了更容易地测试它,我们可能想要控制 AwesomePayments 的实例:

```
class Purchaser
def initialize(awesome_payments = AwesomePayments)
    @awesome_payments = awesome_payments
end
def charge(purchase)
    @awesome_payments.charge(purchase.customer.id,purchase.amount)
rescue => ex
    try_again_later(purchase.id)
end
end
```





现在,就可以在测试时传入 AwesomePayments 的模拟实现,从而更好地控制测试。 测试已经影响了我们的设计(虽然这里的影响比较小)。你甚至可以说,这个类就是更 好的代码。但情况并非总是如此。

我会使用同你处理端到端测试一样的标准:做让生活更轻松的事,但不要做过头, 务必要恰到好处。

八、小结

从头到尾实现一个特性的能力完全取决于你测试它的能力。QA团队或客户测试你的代码时的反馈循环是比较危险的。即使有一个QA团队,他们也不应该发现任何 bug,如果你想快速发布软件,你就不要嫌编写端到端的用户行为测试麻烦。



无脚本测试自动化框架:工具及示例

◆译者:桃子

本文将以简单的术语为您提供无脚本测试自动化框架的介绍,为了便于容易理解, 无脚本自动化的设计目标,实现要求,优势都给予示例进行了说明。

在本文总结处,将介绍该框架相关的"为什么","什么","何时"和"如何"问题的所有答案

在深入探讨无代码测试自动化框架之前,让我们快速看一下自动化的级别

1、脚本/编码

顾名思义,这需要创建一个程序/软件来实现自动化测试或过程。

示例:如果必须执行特定功能或类的单元测试,则需要创建测试程序来测试功能或 类。没有编码就无法自动化单元测试。

通常,如果自动化工具或框架要求测试人员创建一些代码或脚本以提供自动化测试 的测试输入,执行和报告,则将此类工具归类于此级别。

2、无脚本

在此级别,您不需要任何编码或脚本知识即可创建自动化测试。但是,作为框架, 提供定制功能是一个好习惯。

对于高级用户,可以通过编写自己的代码来覆盖某些独特的测试场景来进行自定 义。

3、使用人工智能(AI)的高级自动化

这是可以实现的最高自动化水平:

• 从测试用例文档创建测试脚本 (可能使用自然语言处理技术)。





● 根据应用程序的更改自动更正测试脚本,以进行有效的回归测试。

- 根据被测应用程序上的用户操作/用例,自动生成测试脚本。这是传统记录和播放功能的高级版本。
- 基于机器学习的测试结果自动验证,无需用户输入需要验证的内容和测试分析。

在本文中,我们将重点介绍无脚本的自动化。在此之前,让我们看一些顶级的无脚 本自动化工具列表

一、最好的无脚本测试自动化工具

下面列出了此类别下的顶级工具,以供您参考

1. TestCraft



TestCraft 是一个无脚本的 Selenium 测试自动化平台。其革命性的 AI 技术和独特的 可视化建模可以更快地创建和执行测试,同时消除测试维护开销。

测试人员无需编码即可创建全自动测试场景。客户可以快速发现错误,更频繁地发 布错误,与 CI/CD 集成,批准并提高其数字产品的整体质量。

主要特征:

- 无需编程技能。利用手动测试人员的业务知识,并允许他们创建无代码的自动 化测试方案。
- 大大降低维护成本。由于我们的 AI 算法,脚本会自动调整以进行更改。单击几下,即可进行主要更改。
- 基于 Selenium 快速集成和利用社区开发的模块。
- 它提供了立即实现价值的时间。无需安装(SaaS)。容易掌握。

2、ACCELQ

VCCELO





简单易用,功能强大的 AI 驱动的测试自动化平台。

ACCELQ 是唯一基于云的连续测试平台,可以无缝地自动执行 API 和 Web 测试, 而无需编写任何代码。各种规模的 IT 团队都使用 accelQ 通过自动化生命周期的关键方 面(例如测试设计,计划,测试生成和执行)来加速测试。

accelQ客户通常可以节省测试变更和维护工作所涉及成本的70%以上,从而解决了 行业的主要难题之一。accelQ通过AI驱动的内核使这一点成为可能,从而在其他独特 功能中带来了自愈自动化。

专注于设计和用户体验是 accelQ 持续创新方法的核心,它不懈地努力加快测试速度并提高其客户的交付质量。

关键能力:

- 用于设计和生命周期自动化的可视化应用程序建模。
- 无代码 API 和 UI 自动化在同一流程中。
- 自动化的测试用例生成和数据计划。
- 具有多个抽象层的 insprint 自动化。
- 内置的测试管理和计划功能。
- 无缝的 CI / CD 和 Jira / ALM 集成,具有自然可追溯性。
- 具有运行时分析引擎的自我修复功能,可实现强大的自动化。
- 没有供应商锁,可扩展框架与开源保持一致

3. Ranorex Studio



Ranorex Studio 包括一整套用于桌面,网站和移动应用程序测试的无脚本测试自动化工具。自动化测试的时间是其他方法的一半。

主要功能包括:

• Ranorex Recorder: 捕获与您的应用程序的每次交互。在捕获窗口中, 暂停并继





续记录,编辑步骤以及添加验证操作。

- Ranorex Spy: 只需单击即可识别 UI 元素,然后将它们拖放到测试用例中。
 Ranorex Studio 自动处理任何动态 Web 元素。
- 动作编辑器:编辑记录的步骤,添加验证,指定执行条件(IF语句)以及使用
 Excel 文件或数据库中的数据-所有这些都无需编码。
- 内置 Selenium WebDriver。在 WebDriver 端点,本地 Selenium Grid 或基于云的 提供程序上执行跨浏览器测试。
- 内置的测试运行报告和测试执行的视频报告。如果测试步骤失败,只需重播视频以查看发生了什么。单击测试报告中失败的步骤,以直接跳至测试用例并进行调试。
- 与包括 Jira 和 Bugzilla 在内的领先缺陷跟踪工具,包括 Jenkins 和 Bamboo 在内的 CI / CD 工具,用于测试用例管理的 TestRail 集成在一起。

Ranorex Studio 还具有完整的 IDE,因此您可以根据需要使用代码扩展测试用例。

二、设计目标

下面简要介绍了一些应考虑的重要设计目标。

1、输入驱动

运行自动化测试所需的任何输入都应超出自动化框架的"编码"部分。输入内容可以是您的测试步骤,测试条件,测试数据,预期结果等。

对于每种类型的输入,框架都应以易于使用的关键字形式提供支持,从而也可以理 解上下文。

2、允许自定义

显然,这意味着您应该提供一些选项来为高级用户自定义测试脚本的处理,以便它 们可以包含独特的方案。高级用户可以将他们的测试脚本(以程序的形式)插入到框架 的测试输入中。

3、简单性

这是一个非常重要的设计目标,因为了解如何设置,配置和使用自动化框架应该比



编码变得困难。

一个这样的示例将是提供一个模板, 该模板具有所有必需的测试输入以及默认值 (但与上下文相关)。

这样,用户将迅速了解如何根据他们的应用程序需求提供测试输入。另一种方法是 在测试输入中提供高水平的抽象。

示例: 让我们模拟单击按钮。在这里, 作为用户, 我只对单击按钮的简单操作感兴趣。

该框架应该能够通过内部处理标识按钮控件的不同方法来执行该操作,然后再执行 该操作。实际上,这与输入驱动的目标有些相关,在该目标中,您必须定义易于理解的 关键字,如本例中的 ClickButton。

4、与平台无关

必须能够在不同的 PC 和 Mobile 平台上运行。针对 Web 应用程序开发的无脚本自动化情况下的不同浏览器。作为此目标的一部分,您还应该考虑支持跨不同平台的通用 测试输入。

5、技术不可知

在当今的情况下,我们都知道技术是广泛的,开发应用程序有很多选择。

无论使用什么技术来开发被测应用程序(在给定的应用程序类型(例如 Web 或移动 应用程序)下),自动化框架都应该能够测试这些应用程序,而对测试输入和框架引擎 本身所做的更改很少或没有任何更改。

6、测试环境不可知

它应该能够在云和本地虚拟环境上运行。

7、可扩展

我们应该提供选项来插入其他软件组件,以涵盖复杂的测试方案。

让我们以包含前端应用程序和 Web 服务的电子商务网站为例,您想测试应用程序的 功能以及性能。这项要求肯定需要使用可以测试 Web 服务和网站性能的工具的插件。

三、洞察实施要求







在研究实现细节之前,让我们首先了解无脚本自动化框架的高级体系结构组件。



我们将看到3个主要组件(当然,可以将这些组件进一步划分为较低的级别,但这 不是现在的目的!),我们将看到该组件以大致了解框架:

1)框架核心引擎:这是框架的核心组件,它完成了自动化逻辑,验证,与体系结构其余组件的接口以及第三方应用程序/脚本的所有繁重工作,用于自定义测试配置和处决。

将所有测试输入操作转发到操作处理程序,以处理 AUT(被测应用程序)上的操作。对于开发此引擎,重要的是要考虑上述设计目标,选择合适的编程知识和技术。

根据明智的技术选择,您将能够满足大多数设计目标,从而为自动化框架的消费者 提供最大的收益。

2)测试输入接口:从用户的角度来看,这是前端组件,您将在其中提供所有测试输入,例如测试用例步骤,测试条件,测试数据,预期结果等。

记住简单性的设计目标,应该以一种使消费者可以快速适应并开始使用您的框架的 方式设计此组件。未能达到此目标将导致框架本身的使用减少。

核心引擎的接口是解释器,通常遵循某种协议来交换测试输入上的信息。

3)测试输出接口: 该组件负责输出测试结果,测试日志,缺陷日志,报告和仪表板。框架核心引擎会将结果转发到 Output logger,然后由 Output 接口呈现。

在上述组件中,使用者通常将与 Test 输入接口和 Test 输出接口连接。因此,我们将 仅探索这两个组件。

在上述组件中,使用者通常将与 Test 输入接口和 Test 输出接口连接。因此,我们将 仅探索这两个组件。

测试输入接口

可以以不同方式提供测试输入。但是,这完全取决于我们针对此框架所针对的消费 者类型。





- XML, Excel 等文件; 诸如测试数据字段, 测试步骤, 数据类型之类的架构将 由框架引擎定义,并且在提供测试输入时必须遵循该架构。
- 基于 GUI 的输入,您将在提供相关的测试字段,以便使用者可以提供所有测试 详细信息以执行。
- 基于可视化编辑器的输入,消费者可以以流程图的形式提供这些输入(支持基本逻辑,例如条件和循环)。实际上,这是提供输入接口的最流行的方式,尽管 开发这种接口比较耗时。

需要提供的典型功能集是:

- 能够根据测试类型,测试优先级等对测试用例进行分组。
- 必须能够创建自定义测试套件以运行冒烟和回归测试,选择性测试等。
- 测试与平台无关的输入。
- 能够多次执行单个测试用例。
- 我们应该能够针对不同的测试值执行单个测试用例。
- 能够针对特定的测试步骤或测试用例启动另一个可执行文件。例如,与 Jenkins 的接口以自动启动测试执行。
- 测试执行和输出日志记录的配置。示例:超时。
- 设置"测试脚本库"以确保在必要时可在团队和项目之间重复使用。
- 能够从过去创建的模板中导入测试数据输入。
- 准备导入测试用例文档并将其映射到测试数据输入。记住"可扩展性"设计目标。是的,这是迈向更高水平自动化的第一步!

测试输出接口

从无脚本自动化的角度来看,您还必须确保能够以无缝方式查看测试执行结果,而 无需编写特殊代码。

需要提供的典型功能集是:

● 能够查看测试结果,测试日志,框架日志等。





- 生成各种测试报告和仪表板。
- 能够根据特定要求创建自定义报告。
- 必须能够在运行时通过 GUI 传输测试结果。
- 能够与标准测试用例管理工具对接。
- 应该能够以不同格式(例如 Excel, doc, pdf等)导出测试结果,报告和仪表板。
- 能够针对特定要求(例如性能测试工程等)过滤和解析测试日志。
- 提供将所有测试数据(结果,日志,选择性数据点)推送到高级模块(例如分析)的功能。记住"可扩展性"设计目标。是的,这是迈向更高水平自动化又迈出的第一步!

四、关于无脚本自动化的神话

以下是有关此框架的一些误解。

1) SAF是 100% 无脚本的:实际上,这不是 100% 正确的。尽管目标是使其 100% 无编码,但实际上不可能实现这一壮举。

这样做的主要原因是被测应用程序(AUT),技术和用于开发软件的编程语言中存 在各种复杂性。因此,我们提到的设计目标之一是为定制提供灵活性。

2)使用 SAF,测试人员不需要任何编程或编码知识:再次,这不是 100%正确。实际上,它们确实需要一些编程和逻辑技能的基本概念。

对于基于 Web 或移动的应用程序,测试人员还需要了解对象和类的基础知识,这些 对象和类需要作为自定义脚本的一部分包含在内。

实际上,有些工具(例如 TestCraft)不需要 QA 专业人员来拥有编码知识。它们的 拖放功能以及即时运行功能使任何人都可以了解如何运行测试。

3) SAF 完全取决于记录和回放,可以依靠:否。记录和回放只会为用户提供一个工作脚本,该脚本具有输入和流的硬编码值,必须稍后对其进行自定义。

无论如何,它都不涵盖用户输入和测试步骤的所有可能情况。



五、无脚本示例

为了便于理解,下面给出了一些无脚本自动化的示例。

1) 这是一个测试执行示例

- 开启网页
- 在用户名字段中输入用户
- 在密码字段中输入密码
- 点击登录按钮

以上每个都是无脚本框架支持的关键字。

作为测试人员,您只需传递如下所示的值:打开网页"www.linkedin.com"

Enteruser dummyuser

Enterpassword dummypwd, mask = true

单击"登录"

斜体字是接受参数以执行操作的关键字。

注意: EnterPassword 接受 2 个参数。第二个是标志,用于指示在密码字段中键入字符时的掩码。

2) 假设您要自动执行付款交易流程:

- 选择账户类型
- 输入购买商品的金额
- 输入密码
- 打印收据

SelectAccount 信用

EnterAmount INR1000

EnterPIN 1234, mask = true

PrintReceipt Customercopy = false, Merchantcopy = true





斜体字是接受参数以执行操作的关键字。

注意: EnterPIN 接受 2 个参数。第二个是标志,指示在 PIN 输入字段中键入时 PIN 数字的掩码。PrintReceipt 也有 2 个参数。

如果您注意到上面的流程,这就是典型的交易流程。您可以将整个流程用作各种测 试用例的"事务测试模板"。这是使测试人员易于生成测试步骤和测试输入的方式。

关键字,关键字组和基于流的关键字的可重用性可以在无脚本自动化框架(SAF) 中完成

六、结论

通过本文,您将发现开发无代码或无脚本的自动化框架是一项具有挑战性且同样有 趣的任务。

您可以为任何类型的应用程序测试开发此类框架,即基于 Web 的,移动应用程序, Web 服务,GUI,数据库,嵌入式设备应用程序等。

但是,应格外小心地开发稳定,可靠和高性能的工具。这只是警告,选择正确的技 术和脚本/编程语言来开发一个奇妙而有用的框架。



◆ 作者:枫 叶



Appium 自动化测试遇到的 chromedriver/chrome 坑

一、前言

app 的 webview 自动化是依赖于 chromedriver 的,并且每个 app 的 webview 版本号都不太一样,这就导致了每次都需要重新去下载对应的 chromedriver 版本。如何根据当前的 webdriver 版本去匹配对应 chromedriver 版本,这是一个难题。

根据官方文档翻译过来看,版本不匹配的话一般会报错: An unknown server-side error occurred while processing the command. Original error: unknown error: Chrome version must be >= 55.0.2883.0

二、Chrome driver 启动

用 chrome 浏览器运行自动化测试用例时,如果报这样的错误

selenium.common.exceptions.WebDriverException: Message: unknown error: call function result missing value

可以这样来解决: 指定 chromedriver.exe 驱动绝对路径 driver = webdriver.Chrome(r'e:\xxx\chromedriver.exe')

三、常遇错误

NATIVE_APP

我们在使用 native 和 h5 混合的应用程序测试时,可能会遇到报错

E:\ProgramFiles(x86)\Python\Python37\lib\sitepackages\selenium\webdriver\remote\webdriver.py:1031: UserWarning: name used for saved screenshot does not match file type. It should end with a `.png` extension。 "type. It should end with a `.png` extension", UserWarning) ..['NATIVE_APP', 'WEBVIEW_chrome', 'WEBVIEW_com.android.browser']





www.51testing.com



Doctor my center Test Over. Е ERROR: test_e_AboutContact (___main__.center) Traceback (most recent call last): File "E:/ATS/TCyDoctorNew/test_case/test_dir/test_4doctormycenter.py", line 371, in test_e_AboutChengyiContact driver.switch_to.context('WEBVIEW_com.android.browser') File"E:\ProgramFiles(x86)\Python\Python37\lib\site-packages\appium\webdriver\switch_to.py", line 31, in context self._driver.execute(MobileCommand.SWITCH_TO_CONTEXT, {'name': context_name}) File"E:\ProgramFiles(x86)\Python\Python37\lib\sitepackages\selenium\webdriver\remote\webdriver.py", line 321, in execute self.error_handler.check_response(response) $File"E:\ProgramFiles(x86)\Python\Python37\lib\site-packages\appium\webdriver\errorhandler.py", line and the set of the$ 29, in check_response raise wde File"E:\ProgramFiles(x86)\Python\Python37\lib\site-packages\appium\webdriver\errorhandler.py", line 24, in check_response super(MobileErrorHandler, self).check_response(response) File "E:\Program Files (x86)\Python\Python37\lib\sitepackages\selenium\webdriver\remote\errorhandler.py", line 242, in check_response raise exception_class(message, screen, stacktrace) selenium.common.exceptions.WebDriverException: Message: An unknown server-side error occurred while processing the command. Original error: No Chromedriver found that can automate Chrome '55.0.2883'. See https://github.com/appium/appium/blob/master/docs/en/writing-runningappium/web/chromedriver.md for more details. Ran 3 tests in 66.001s FAILED (errors=1) Process finished with exit code 0

我们来看报错信息,第一个是 warning,是指截图的格式最好是 png,这个与本文无 关我们先忽略,关键看第二个错误,它主要缘于 "No Chromedriver found that can automate Chrome '55.0.2883",在 appium 日志里也能看到详情



《51 测试天地》五十七(下)

www.51testing.com





四、了解 chromedriver

通过管理 Chromedriver, Appium 支持安卓网页和支持谷歌的混合 app 的自动化。 通过 npm package 安装的总是绑定最新的 chromedriver。

但是,每一版 chromedriver 的更新会支持最小的新增谷歌的版本,所以早期的版本 不能再跟绑定的版本兼容。这时在 appium server 日志里会有类似的错误:

An unknown server-side error occurred while processing the command.

Original error: unknown error: Chrome version must be >= 55.0.2883.0

为解决这个问题, appium 可以通过加上一chromedriver_version 属性配置使用特定的 chromedriver 版本, 比如

npm install appium - chromedriver_version="2.16"

或者在 CHROMEDRIVER_VERSION 环境变量指定版本,如

CHROMEDRIVER_VERSION=2.20 npm install appium

这也能得到最新的版本。最后,还可以被指定在运行时,通过一chromedriverexecutable 服务器标识与 chromedriver 执行路径手动下载,比如

appium --chromedriver-executable /path/to/my/chromedriver

五、chromedriver/chrome 兼容

下面是 chromedriver 与最小的 chrome 版本对应表:





www.51testing.com

版本	最小谷歌版本	Chromedriver 链接
2.44	69.0.3497.0	v2.44
2.43	69.0.3497.0	v2.43
2.42	68.0.3440.0	v2.42
2.41	67.0.3360.0	v2.41
2.40	66.0.3359.0	v2.40
2.39	66.0.3359.0	v2.39
2.38	65.0.3325.0	v2.38
2.37	64.0.3282.0	v2.37
2.36	63.0.3239.0	v2.36
2.35	62.0.3202.0	v2.35
2.34	61.0.3163.0	v2.34
2.33	60.0.3112.0	v2.33
2.32	59.0.3071.0	v2.32
2.31	58.0.3029.0	v2.31
2.30	58.0.3029.0	v2.30
2.29	57.0.2987.0	v2.29
2.28	55.0.2883.0	v2.28
2.27	54.0.2840.0	v2.27
2.26	53.0.2785.0	v2.26
2.25	53.0.2785.0	v2.25
2.24	52.0.2743.0	v2.24
2.23	51.0.2704.0	v2.23
2.22	49.0.2623.0	v2.22
2.21	46.0.2490.0	v2.21
2.20	43.0.2357.0	v2.20
2.19	43.0.2357.0	v2.19
2.18	43.0.2357.0	v2.18
2.17	42.0.2311.0	v2.17
2.16	42.0.2311.0	v2.16



^{₩为峰城下} 51 Lest Ing 软件测试网

www.51testing.com

版本	最小谷歌版本	Chromedriver 链接
2.15	40.0.2214.0	v2.15
2.14	39.0.2171.0	v2.14
2.13	38.0.2125.0	v2.13
2.12	36.0.1985.0	v2.12
2.11	36.0.1985.0	v2.11
2.10	33.0.1751.0	v2.10
2.9	31.0.1650.59	v2.9
2.8	30.0.1573.2	v2.8
2.7	30.0.1573.2	v2.7
2.6	29.0.1545.0	v2.6
2.5	29.0.1545.0	v2.5
2.4	29.0.1545.0	v2.4
2.3	28.0.1500.0	v2.3
2.2	27.0.1453.0	v2.2
2.1	27.0.1453.0	v2.1
2.0	27.0.1453.0	v2.0

六、自动查找匹配的 chromedriver

从 Appium1.8.0 开始, appium 可以得到这个正确的基于 chrome 的 chromedriver 版本。当 appium 发布时, appium 与 chromedriver 绑定,更多 chromedriver 版本可供下载 到 appium 的安装文件里(不建议这样做,因为更新 appium 时将会删除它们)。

当一个 appium 版本发布,最新的 chromedriver 版本可以被获取,可以通过 chromedriver 与最小 chrome 版本映射关系,得到映射的绝对路径文件。这个文件内容必 须做成 json 对象,例如

```
{
	"2.42": "63.0.3239",
	"2.41": "62.0.3202"
}
```

七、安装网络问题



当 appium 安装时需要下载 chromedriver,所以会遇到安装失败的网络问题。

默认下 chromedriver 从 https://chromedriver.storage.googleapis.com/ 获取。使用

chromedrivre 的 npm 镜像配置 chromedriver_cdnurl。

npm install appium-chromedriver --chromedriver_cdnurl=http://npm.taobao.org/mirrors/chromedriver

或者在你的.npmrc 文件增加属性

chromedriver_cdnurl=http://npm.taobao.org/mirrors/chromedriver

另一种选择是使用 Path 环境变量 CHROMEDRIVER_CDNURL

CHROMEDRIVER_CDNURL=http://npm.taobao.org/mirrors/chromedriver npm install appium-chromedriver

在国内,可以用 cnpm~

首先,安装 cnpm

npm install -g cnpm --registry=https://registry.npm.taobao.org



输入 cnpm -v,显示 cnpm 不是内部或外部命令,也不是可运行命令或批处理文件。

cnpm 默认会安装在 C:\Users\admin\AppData\Roaming\npm 个人用户路径下。所以追溯到这个文件夹,在这里可以执行 cnpm v,因此推测没有把 cnpm 路径加入到 path 系统环境变量中。

添加到 path 后,再关闭打开 cmd,执行 cnpm - v 命令,



解决这个之后执行安装 chromedriver

cnpm install -g appium --chromedriver_version="2.28" -chromedriver_cdnurl=http://npm.taobao.org/mirrors/chromedriver



《51 测试天地》五十七(下)

www.51testing.com





《51 测试天地》(五十七)上篇 精彩预览

- Java+TestNG+Jenkins 实现接口(API)自动化测试全教程
- Django 环境搭建及 Restful 接口测试实践
- pytest 自动化测试探索
- Python 接口测试实践 (Requests) 用例封装及测试报告生成
- 人与机器人如何协同来测试软件
- 互联网江湖中的技术债
- 做测试的你, 职业倦怠了么?
- 数据处理类系统功能测试探索
- Django 数据管理及接口测试



马上阅读