

基于 python+Selenium+PHP+Ftp 实现的轻量级 web 自动化测试框架

by:授客 QQ: 1033553122

博客: <http://blog.sina.com.cn/ishouke>

欢迎加入软件性能测试交流 QQ 群: 7156436

目录

1、	开发环境.....	1
2、	基本配置.....	2
3、	主要功能逻辑介绍.....	3
4、	框架功能简介.....	5
5、	服务端部署.....	5
6、	框架模块详细介绍.....	5
7、	页面元素管理.....	9
8、	用例管理.....	9
9、	测试计划管理.....	10
10、	运行结果.....	10
11、	源码下载.....	11
12、	说明.....	11

1、 开发环境

win7 64

PyCharm 4.0.5

setuptools-29.0.1.zip

下载地址: <http://pan.baidu.com/s/1mhMSAkK>

官方下载地址: <https://pypi.python.org/pypi/setuptools#downloads>

python 3.3.2

mysql-connector-python-2.1.4-py3.3-win64

下载地址: <http://pan.baidu.com/s/1cDtP10>

官方下载地址: <http://dev.mysql.com/downloads/connector/python/>

selenium-3.0.2

下载地址 1: <http://pan.baidu.com/s/1bFfY4Q>

官方下载地址: <https://pypi.python.org/pypi/selenium>

火狐浏览器: 47.0.2

geckodriver-v0.11.1-win64.zip

下载地址 1: <http://pan.baidu.com/s/1jI0SHTG>

官方下载地址: <https://github.com/mozilla/geckodriver/releases>

IE8

IEDriverServer

下载地址: <http://pan.baidu.com/s/1bHhOC2>

Chrome 浏览器 54.0.2840.59 m

下载地址: <http://pan.baidu.com/s/1kUOreMr>

chromedriver_win32 2.2.5

下载地址 1: <http://chromedriver.storage.googleapis.com/index.html?path=2.25/>

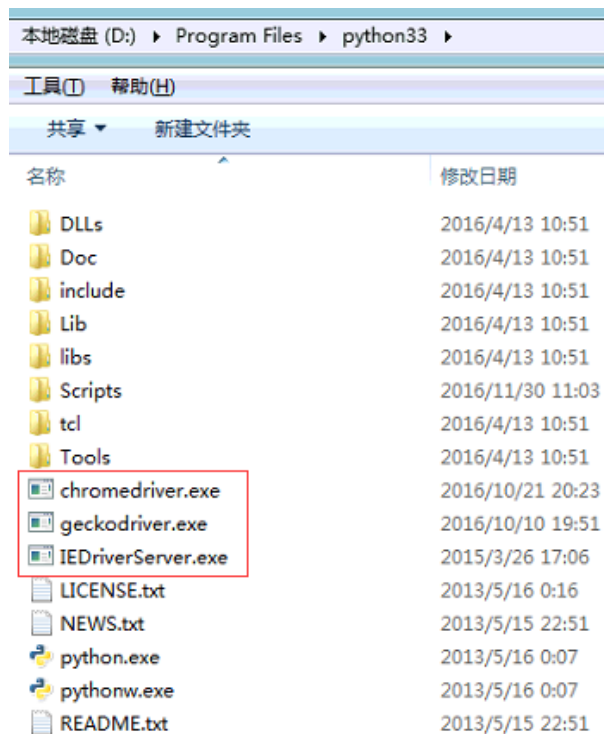
下载地址 2: <http://pan.baidu.com/s/1slGn5hV>

PHP 版本: Apache+PHP5.3

ftp 服务器

2、基本配置

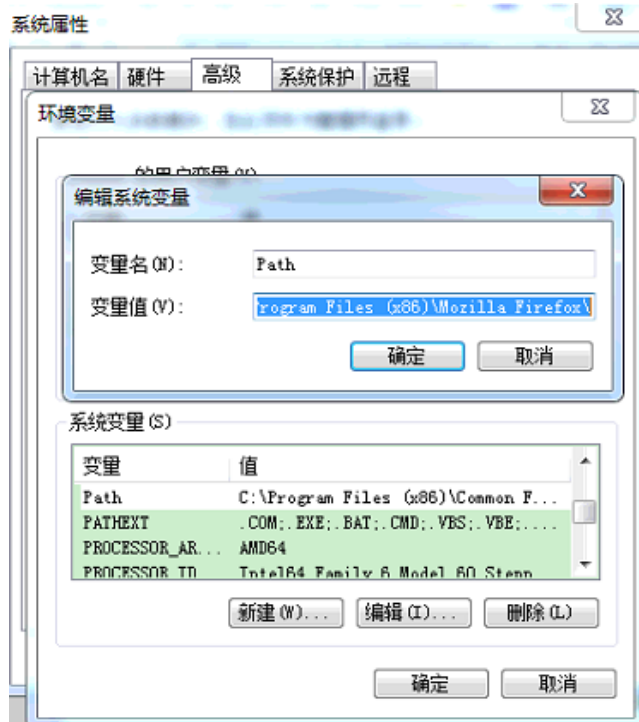
安装完 Python 后, 把 chromedriver.exe, geckodriver 和 IEDriverServer.exe 浏览器驱动, 放到 python 安装目录(例: D:\Program Files\python33)下, 如下



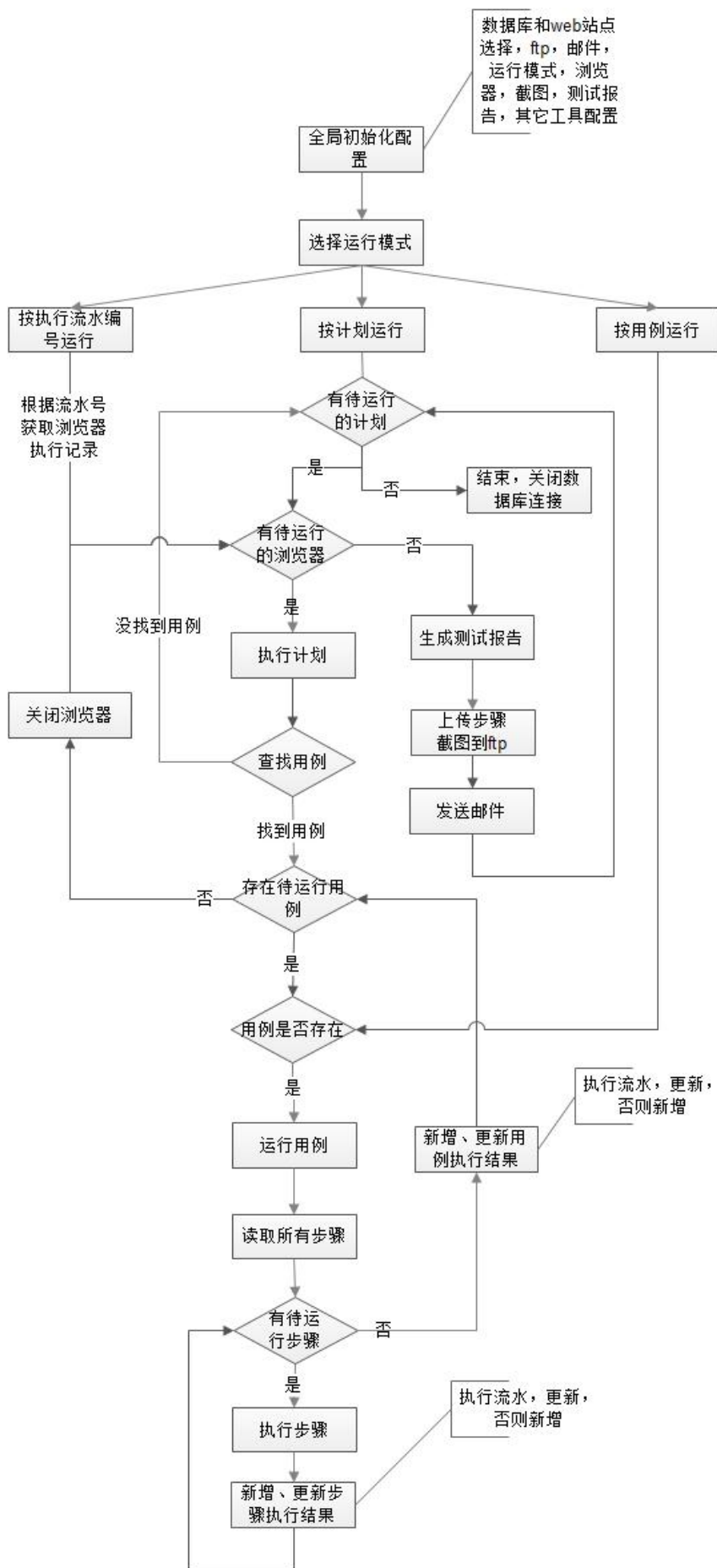
如果火狐浏览器不是安装在默认安装目录下, (例: D:\Program Files (x86)\Mozilla Firefox\), 需要在 path 中添加 firefox.exe 所在的安装目录, 否则会报错, 类似如下:

Message: Expected browser binary location, but unable to find binary in default location, no 'moz:firefoxOptions.binary' capability provided, and no binary flag set on the command line

然后重启浏览器和 pycharm



3、主要功能逻辑介绍



4、 框架功能简介

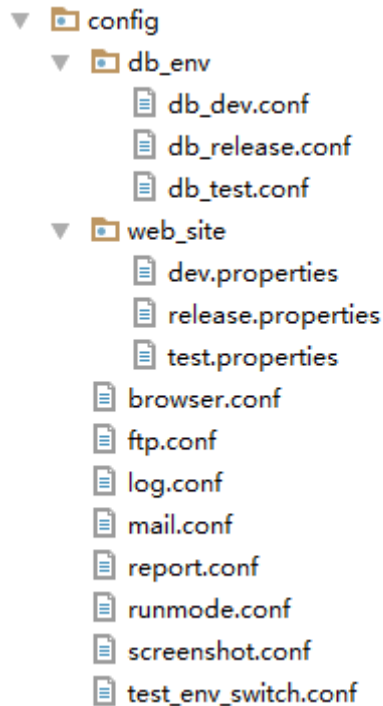
- 1、框架采用 PHP web 页面，管理页面对象，测试计划，测试用例，用例步骤，支持界面元素操作，支持函数执行
- 2、可通过配置文件灵活配置运行模式：
 - 支持按测试计划运行：一次运行单个、多个指定的测试计划；
 - 支持按测试用例运行：一次运行单条测试用例
 - 支持按照执行流水编号运行：每启动一种浏览器，每执行一次都会生成一个执行编号，按流水编号运行，会自动选择上次执行的浏览器，重复运行该计划对应的测试用例（注意：目前计划创建后，其拥有的测试用例保持不变，即无法删，减）
- 3、可通过配置文件，配置需要运行的浏览器，支持自由扩展
- 4、可通过配置文件灵活选择测试需要的 web site, 和数据库
- 5、可自动生成 HTML 可视化接口测试报告
- 6、可根据配置在测试完成后，自动发送测试报告邮件，邮件发送支持 SSL 加密发送和非 SSL 加密发送，同时支持往多个邮箱发送邮件
- 7、支持文件、控制台的日志打印，可分别控制开关
- 8、可集成 Jenkins 自动运行脚本
参考文章: [为 Jenkins 添加 Windows Slave 远程执行 python 项目脚本](#)

5、 服务端部署

参考 “Web 自动化测试-服务端测试环境部署.doc” 文件

6、 框架模块详细介绍

a) config



db_env.conf: 包含不同环境的测试数据库, 应用数据库的配置信息 (目前仅用到了测试数据库)

web_site: 包含不同环境的 web 站点信息, 包括 web url, 帐号, 密码, 验证码 (目前只用到了 url)

browser.conf: 提供浏览器类型配置, 即需要进行测试的浏览器

ftp.conf: 提供 ftp 站点配置信息, 包括 host, port, 用户, 密码

log.conf: 包含日志配置信息, 具体如下:

```
[LOGGING]
log_file = d:/myrunlog.txt ← 日志文件存储路径
backup_count = 5 ← 同一时刻, 可存在的日志数量
fmt = |(asctime)s |(filename)s[line: |(lineno)d |(levelname)s: |(message)s
logger_name = test_logger ← 日志打印器名称
log_level_in_console = 20 ← 日志级别
log_level_in_logfile = 10 ← 日志级别
console_log_on = 1 ← 日志开关
logfile_log_on = 1 ← 日志开关

#日志级别: CRITICAL = 50 ERROR = 40 WARNING = 30 INFO = 20 DEBUG = 10 NOTSET = 0
#console_log_on = 1 开启控制台日志, logfile_log_on = 1 开启文件日志
```

mail.conf: 包含邮件发送配置信息, 如下,

```
[SMTP]
login_user = shouke@yunnex.com
login_pwd = xxxx
from_addr = shouke@yunnex.com
to_addrs = ['1033553122@qq.com']
host = smtp.exmail.qq.com
port = 465
encrypt = 1

[README]
ecrypt = '1 - send via SSL, 0-sent not via SLL'
```

发件人帐号, 密码, 邮箱

收件人地址, 英文逗号分隔

邮件服务器hos和端口

1 - SSL加密发送, 0不加密发送

注: 不同类型的邮箱(发件人邮箱), 需要修改配置文件为对应邮件服务器的 **host** 和端口
smtp.163.com:25
smtp.qq.com:465

report.conf: 包含测试报告文件配置信息, 如下

```
[REPORT]
dir_of_report = E:\Projects\report\
report_name = test_report.html

[README]
dir_of_path = '不加引号', D:\tset\tkise\
report_name = '不加引号'
```

报告所在目录

报告名称

runmodeconfig.conf: 包含运行模式配置信息

```
[RUNMODE]
runmode = 运行流水

[TESTPLAN]
testplans = [2]

[GLOBALCASE]
globalcases= [23]

#说明: run_mode: 运行用例, 运行计划, 运行流水
#注意: 运行计划, 需要配置TESTPLAN节点下的testplans
#注意: 运行用例, 根据实际需要, 配置GLOBALCASE节点下的优先运行的全局初始化用例
```

screenshot.conf: 提供截图配置信息

```
[SCREENSHOT]
dir_path = E:\screenshot\
remote_screenshot_baselink = http://192.168.9.98/webautotest/screenshot/
#说明: 格式 Windows:D:\tset\tkise\ Linux: /tset/tkise/
```

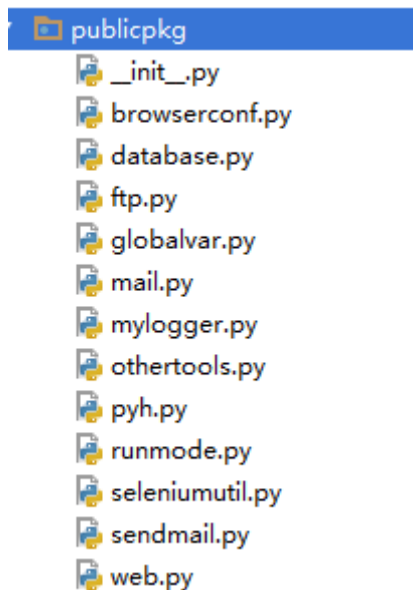
截图远程访问基路径地址, 供测试报告中, 查看截图使用,
图片地址格式: remote_screenshot_baselink+snapshot.png

test_env_swith: 提供环境切换

```
[TEST_ENV]
web_site = 1
db_env = 1

#说明:
#web_env: 1-测试环境 2-开发环境 3-release环境
#db_env: 1-测试环境 2-开发环境 3-release环境
```

b) publicpkg



broserconf.py: 浏览器配置类

database.py: 提供数据库初始化配置, 封装数据库操作, 增删改查等

ftp.py: 实现 ftp 上传下载, 封装 ftp 操作

mail.py: 实现邮件发送

runmode.py: 运行模式配置类

web.py: 浏览器配置类

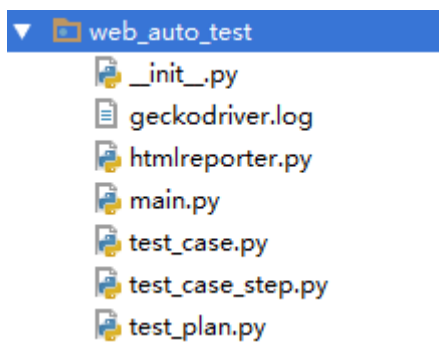
othertools.py: 工具类, 实现其它功能, 比如批量创建目录, 转换 php 页面存储的数据, 递归遍历指定目录下的特征文件

globalvar.py: 主要提供全局变量, 全局实例等

注意: 这里使用了__all__控制 import * 能导入的名称 (全局变量, 全局实例等)

seleniumutil.py: 主要是封装了 selenium 的一些基础操作, 及极少部分的 py 函数, 支持自由扩展

c) web_auto_test



说明:

- 1、提供程序入口, 测试计划, 测试用例, 测试步骤, 测试报告类
- 2、因时间仓促及其他原因等, 对象定义、设计上或多或少存在一些不合理的地方

7、 页面元素管理

登录页面											
序号	元素名	selector01(第1选择)	selector02(第2选择)[可选]	编辑	操作						
1	用户名	xpath=//input[@id="username"]									
2	密码	xpath=//input[@id="password"]									
3	登录按钮	xpath=//input[@id="loginBtn"]									

8、 用例管理

[登录] [ID:10]											
步序	页面	元素/组件/py函数	操作	输入参数	预期结果	编辑	其他操作				
1	登录页面	用户名	输入	15812232222	output				+组件		
2	登录页面	密码	输入	123456						+组件	
3	登录页面	登录按钮	点击							+组件	
4			等待	2						+组件	
5	首页	退出按钮	查找						+组件		

(注:参数以||分隔。)

注:

- 1、目前“操作”仅支持元素操作，执行函数
- 2、时间关系，仅实现了小部分操作
- 3、关于步骤断言的想法：这块可把断言看成是步骤，步骤失败则断言失败，用例也失败，否则成功
- 4、用例依赖：用例之间最好互不依赖，如果是按计划执行，可考虑调整计划中的用例执行顺序，如果是按照单个用例的执行，可在 `runmode.conf` 配置文件中按顺序添加全局初始化用例
- 5、我的感觉：ui 自动化这块用例之间或多或少会存在依赖，因为元素的查找依赖页面，不同操作会打开不同的页面，如果某个操作在某个页面失败了，没返回到另一个用例查找元素需要的页面，这会执行另一个页面的操作时是否也就意味着失败呢，除非每个用例之间做到了互不依赖？时间关系，没来得及拿本框架去做更多实验，这块待研究

9、测试计划管理



执行顺序	tcid	用例名称	操作
1	10	登录	↓下移
2	13	展开派券中心	↑上移

2.保存执行顺序 (注: 勿忘记保存!)

10、运行结果

test_report20161209031511.html

web_autotest_report

【 Chrome浏览器:计划执行摘要 】 #####
####

测试总耗时: 0:00:44.168609

计划名称: 测试计划2

用例执行统计: 执行用例总数: 2 成功用例数(Pass): 1 失败用例数(Fail): 1 未执行用例数(Block): 0

【 Ie浏览器:计划执行摘要 】 #####
#

测试总耗时: 0:00:44.839193

计划名称: 测试计划2

用例执行统计: 执行用例总数: 2 成功用例数(Pass): 1 失败用例数(Fail): 1 未执行用例数(Block): 0

【 Chrome浏览器:用例执行摘要 】 #####
####

ID	执行编号	用例ID	用例名称	所属计划	计划ID	运行浏览器	运行结果	运行时间	操作
221	1481224389	10	登录	测试计划2	2	Chrome	Pass	2016-12-09 03:13:09	点击查看详情
222	1481224389	13	展开派券中心	测试计划2	2	Chrome	Fail	2016-12-09 03:13:13	点击查看详情

【 Ie浏览器:用例执行摘要 】 #####
#

ID	执行编号	用例ID	用例名称	所属计划	计划ID	运行浏览器	运行结果	运行时间	操作
223	1481224432	10	登录	测试计划2	2	Ie	Pass	2016-12-09 03:13:52	点击查看详情
224	1481224432	13	展开派券中心	测试计划2	2	Ie	Fail	2016-12-09 03:13:57	点击查看详情

【 Chrome浏览器:用例执行明细 】 #####
####

>>>测试用例【testcaseID : 10 名称 : 登录】 [点击返回](#)

ID	步序	执行编号	所属用例	元素名称	操作	输入参数	运行结果	预期结果	结果描述	运行浏览器	运行时间	操作
362	1	1481224389	登录	用户名	输入	15812232222	Pass	output	输入数据成功	Chrome	2016-12-09 03:13:09	
363	2	1481224389	登录	密码	输入	123456 123456	Pass		输入数据成功	Chrome	2016-12-09 03:13:09	
364	3	1481224389	登录	登录按钮	点击		Pass		点击成功	Chrome	2016-12-09 03:13:09	
365	4	1481224389	登录		等待	2	Pass		run success	Chrome	2016-12-09 03:13:11	
366	5	1481224389	登录	退出按钮	查找		Pass		找到元素	Chrome	2016-12-09 03:13:13	

>>>测试用例【testcaseID : 13 名称 : 展开派券中心】 [点击返回](#)

ID	步序	执行编号	所属用例	元素名称	操作	输入参数	运行结果	预期结果	结果描述	运行浏览器	运行时间	操作
367	1	1481224389	展开派券中心	派券中心	点击		Fail		点击操作失败	Chrome	2016-12-09 03:13:13	查看截图

【 Ie浏览器:用例执行明细 】

11、 源码下载

下载地址: 暂不打算开源, 如果开源, 仅在 QQ 群 (群号: 71564360) 内分享

下载后解压, 用 pycharm 导入项目即可

12、 说明

- 1、时间有限, 精力有限, 暂且就到这吧, 有需要的可以自己扩展、修改框架。
- 2、PHP 页面非本人编写, 部分代码实现设计受其约束