接下来就是关于这个项目的介绍：

UItestframework项目目前具有以下功能：  
1、对webdriver进行了第二次的简单封装，使用更加方便 public/common/pyselenium.py  
(ps：这个是在虫师的pyse基础加了打印日志功能,参考：https://github.com/defnngj/pyse)  
2、可以对excel表进行数据读取，完成数据驱动:public/common/datainfo.py  
3、具有打印日志的功能，打印在控制台和文件中：public/common/log.py,日志保存在report/log/目录下  
4、读取配置文件(.ini文件):public/common/readconfig.py  
5、具有发邮件的功能:public/common/sendmail.py  
6、生成测试报告：html测试报告的路径：report/testreport/目录下  
7、使用了PageObject模式来编写测试脚本

整个项目的目录结构:  
├─config 配置文件的目录  
│ │ config.ini 存放配置文件  
│ │ globalparam.py 重要的全局参数，如log、report的路径配置等  
│ │ \_\_init\_\_.py  
│ │  
│  
├─data 测试数据  
│ ├─formaldata # 正式环境测试数据  
│ └─testdata # 测试环境的数据  
│ searKey.xlsx  
│  
├─public 公共的文件库  
│ │ \_\_init\_\_.py  
│ │  
│ ├─common 封装的公共方法  
│ │ │ basepage.py  
│ │ │ datainfo.py  
│ │ │ log.py  
│ │ │ mytest.py  
│ │ │ publicfunction.py  
│ │ │ pyselenium.py  
│ │ │ pyselenium20161107.py  
│ │ │ readconfig.py  
│ │ │ sendmail.py  
│ │ │ \_\_init\_\_.py  
│ │ │  
│ │  
│ ├─pages 使用pageobject模式编写测试脚本，存放page的目录  
│ │ │ baiduIndexPage.py  
│ │ │ \_\_init\_\_.py  
│  
├─report 测试报告  
│ ├─image 截图目录  
│ ├─log 日志目录  
│ │ 2016-11-07.log  
│ │  
│ └─testreport html测试报告目录  
│ TestResult2016-11-07\_16\_15\_51.html  
│  
└─testcase 存放测试用例  
│ test\_baidu.py

使用说明:  
安装响应的库: pip install xlrd,selenium,configparser  
1、在config.ini中配置项目路径：project\_path  
2、测试数据放在data目录下面  
3、使用pageobject，写page页面，在测试用例里面调用放在public/pages目录下  
4、在testcase目录下面，编写测试用例，可以分模块编写，建相应的目录  
5、执行run.py,就可以执行所有的测试用例  
6、在report/log里面查看日志  
7、在report/testreport里面查看html测试报告

关于pyselenium的使用:  
该py文件是根据虫师的pyse改的，加了一个日志，根据自己的需要加了几个函数  
可以参考虫师的pyse,github地址:https://github.com/defnngj/pyse  
虫师的博客园地址：https://github.com/defnngj/pyse

导入PySlenium文件  
import PySelenium  
1、启动浏览器：  
启动谷歌浏览器  
dr = PySelenium.PySelenium('chrom')  
启动远程浏览器比如使用grid施行分布式执行  
dr = PySelenium.PySelenium(RChrome','127.0.0.1:8080')  
2、在地址栏输入网址：  
dr.open('http://www.baidu.com')  
3、窗口最大化  
dr.max\_window()  
4、设置浏览器的窗口的大小  
dr.set\_window(800,500)  
5、不清除文本框的内容直接输入值(比如说：进行文件上传时，上传文件的路径，如果清除就会报错)：  
dr.type('id->su','小石头tester')  
6、先清除文本框的内容，然后再输入值（用得很多）：  
dr.clear\_type('name->su','虫师')  
7、直接点击元素  
dr.click('css->#kw')  
8、右键点击元素：  
dr.right\_click('id->kw')  
9、将鼠标移动到一个元素上  
dr.move\_to\_element('clas->btn1.btn-green.btn-search')  
10、双击元素  
dr.double\_click("id->kw")  
11、将一个元素拖拽到另外一个元素上  
dr.drag\_and\_drop('id->kw1','id->kw2')  
12、根据连接的text来点击(<a href="http://www.baidu.com">百度</a>)  
dr.click\_text('百度')  
13、关闭窗口，driver  
dr.quit()  
14、执行js脚本  
dr.js('script')  
15、获取元素的属性  
dr.get\_attribute("id->su","href")  
16、获取元素的文本信息text  
dr.get\_text('id->su')  
17、返回当前页面的title  
dr.get\_title()  
18、返回当前页面的url  
dr.get\_url()  
20、进入frame  
dr.switch\_to\_frame('id->kw')  
21、退出frame  
dr.switch\_to\_frame\_out()  
22、判断元素是否存在  
dr.element\_exist('id->kw')  
23、截图  
dr.take\_screenshot('file\_path')  
24、进入最新的table  
dr.into\_new\_window()  
25、输入内容并且回车  
dr.type\_and\_enter('id->kw')  
26、使用js来点击某个元素  
dr.js\_click('id->kw')  
27、返回原生的webdriver，进行个性化需求  
dr.origin\_driver()

     接到就到这里吧，其他功能以后会逐步添加，希望对项目的测试能带来更高的效率；也希望给各位看官带来帮助，也希望能得到大神指点，让框架更加完善，健壮。。。