**AutoRun程序使用说明书V1.1**

灰蓝

2016.11.25

**一、 文件夹结构**

此程序用于自动打开网页并填表。最小文件夹结构见下图：

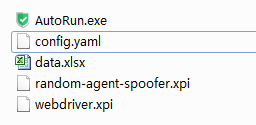


图 最小结构

详细说明：

|  |  |  |
| --- | --- | --- |
| **文件** | **说明** | **是否必须** |
| AutoRun.exe | 主程序，支持64/32位系统 | 是 |
| config.yaml | 配置文件，具体内容见配置说明 | 是 |
| data.xlsx | 数据文件，具体内容见数据说明 | 是 |
| random-agent-spoofer.xpi | Firefox agent插件 | 使用firefox必须 |
| webdriver.xpi | Firefox selenium插件 |
| chromedriver.exe | chrome驱动 | 使用chrome必须 |

**二、 配置说明**

配置文件为config.yaml，必须放在程序同一目录下。

**1. 配置节**

以‘---’来分节，第一节必须为配置节，如下图

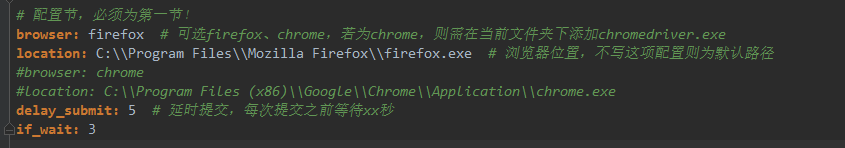


图 配置节

4项配置：

**browser：** 设置所使用的浏览器，可选chrome、firefox，如果选用chrome浏览器，则程序目录下必须添加chromedriver.exe。不配置此项默认用firefox。

**location：** 设置浏览器位置，用于浏览器不在默认路径的情况，不配置此项则使用默认路径。注意是两个反斜杠

**delay\_submit：** 设置点击提交按钮前的延时。不配置此项默认为5秒

**if\_wait:** 设置对于if标签，在判断url前所需要等待的时间，默认是3秒

**2. 任务节**

配置节之下的所有节均为任务节。任务节接受3种标签，如下图：



图 任务节

**注意：这里的标签与上一节配置不同，这里标签带有顺序，前面有‘-’，一定要严格按照配置要求填写！**

**url标签：**任务url

**sheet标签：**任务名称，也是对应data.xlsx里该任务数据对应的sheet页，如下：

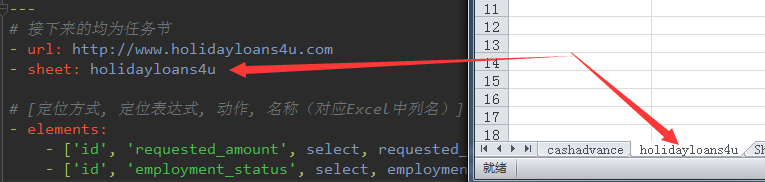


图 任务名称与sheet页对应

**elements标签：**此标签中均为要填写的元素。支持多个elements标签，每个elements标签之间有5秒的间隔，元素与元素之间有1秒间隔。所以一般翻页的时候，分开多个elements，同一个页面内不同元素建议写在同一个elements中。**注意：1）元素前一定是空格，不能是tab；2）元素中各项内容如果有空格，则一定用括号括起来；3）定位表达式中如果有引号，用双引号，想用单引号则需要在单引号前再加一个单引号作为转义**

**元素格式：**

元素格式分成4部分，[定位方式， 定位表达式， 动作， 元素名（对应Excel中列名）]

定位方式支持8种：

|  |  |
| --- | --- |
| **定位方式** | **说明** |
| id | 这些定位方式分别对应selenium的各种定位，根据定位方式填写第二项的定位表达式。一般元素用ID定位即可，其他几种只是备选。 |
| name |
| xpath |
| tag name |
| link text |
| class name |
| partial link text |
| css selector |

动作支持5种：

|  |  |
| --- | --- |
| **动作** | **说明** |
| select | 对于select选项框，按元素名从data中读取数据，注意是按照select标签的value来选择选项，不要用index和visible text |
| sendkeys | 用于向input框中输入数据，按元素名从data中读取值 |
| click | 用于点击某些元素，如勾选checkbox以及radiobutton等 |
| clear | 用于清空某些input框中的默认数据 |
| submit | 用于提交表单，该元素执行前会根据配置节配置等待一定时间，默认5秒；对于提交按钮，也可以用click，不过用click不会等待 |

元素第4项为元素名，这一项是很重要的，一定要把这里定义的元素名跟data.xlsx中的列名对应，否则sendkeys和select动作是无法读取到要输入和选择的数据的！如下图：

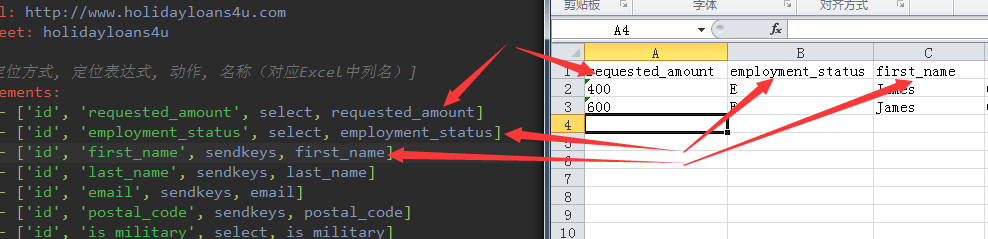


图 元素名与列名对应

在elements标签中有一项特殊的元素，为‘{}’中的‘if’，此项非必须，用于判断页面是否跳转到页面A，如果跳转到A，则任务完成关闭浏览器，下面的元素不再执行；否则，任务继续，接着执行下面的元素。如下图：

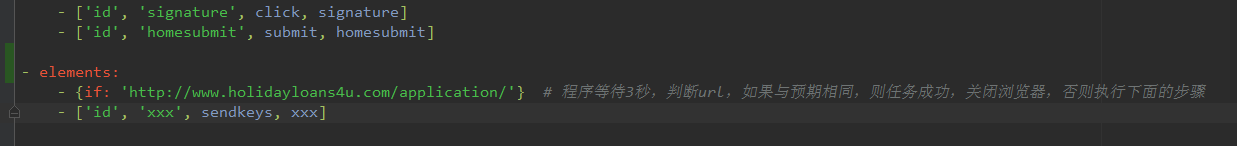


图 if标签

**注意：**

1） 每一个元素都写全四个属性。

2） 每个页面的元素放在一个elements标签中，建议把if判断也放在一个新的elements中。

3） 注意任务节中的各项标签是有顺序的，elements中的各项元素也是有顺序的。配置节中各项配置无序。

4） 如果元素没有在网页中找到，程序会在持续寻找一段时间后跳过该元素，继续执行下面的元素。

**三、数据说明**

数据文件用以配置填充表单的具体数据，主要针对select元素的选项选择和input标签的输入值。如下图所示：

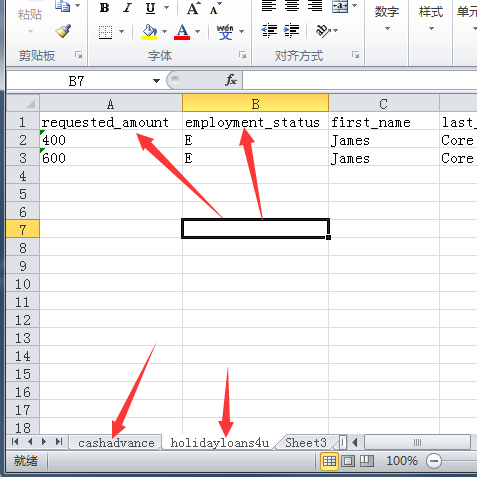


图 sheet名与列名

1. 不同的sheet页对应不同的任务节，在任务节中设置了每个任务的name，即data中的sheet名称，必须对应。

2. 不同的列对应不同的元素，每一个列名都要和配置中的元素名相同，否则找不到匹配的数据。

3. 每一行代表一次循环，程序会自动读取行数，并按行数读取data并循环执行。

4. 对于表格内容的填写，有一点需要注意并且非常重要，所有的数字都要写成文本格式，你可以设置整个表格为文本格式之后再编辑，或者在输入数字的前面加半角单引号“‘”。

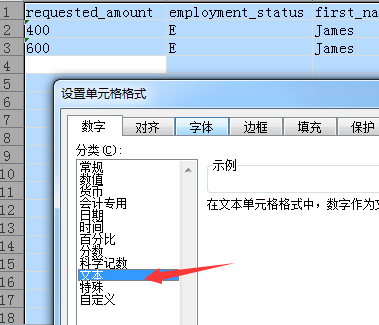
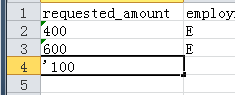
或

图 设置为文本格式

**四、其他文件的说明**

1. random-agent-spoofer.xpi（要求）和webdriver.xpi是启动Firefox必须的，名字也不可以修改，必须按照规定的名称，random-agent-spoofer插件可以根据Firefox的版本替换相应的版本。

2. 要启动chrome，必须在程序路径添加chromedriver.exe，chromedriver的版本必须与要启动的chrome版本匹配，具体的版本对应可参照我的博客【http://blog.csdn.net/huilan\_same/article/details/51896672】

**五、程序运行**

**程序执行步骤：**

1. 根据页面元素，填写config.yaml，设置基本配置，以及页面任务

2. 根据任务中需要填写的项，在data.xlsx中添加对应的sheet页，并添加对应的列以及数据

3. 可双击AutoRun.exe运行，但这样一旦出错或程序运行结束会直接关闭窗口，看不到输出的日志信息，所以建议用cmd执行：

1）【Win+R】，打开“运行”窗口，输入cmd并确定

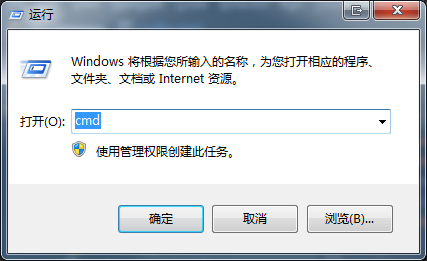


图 打开cmd

2）用cd命令切换到程序所在文件夹

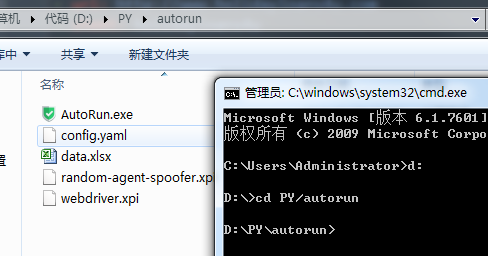


图 切到程序路径

3）输入AutoRun.exe并回车，执行程序。这样程序结束也可看到之前的日志信息

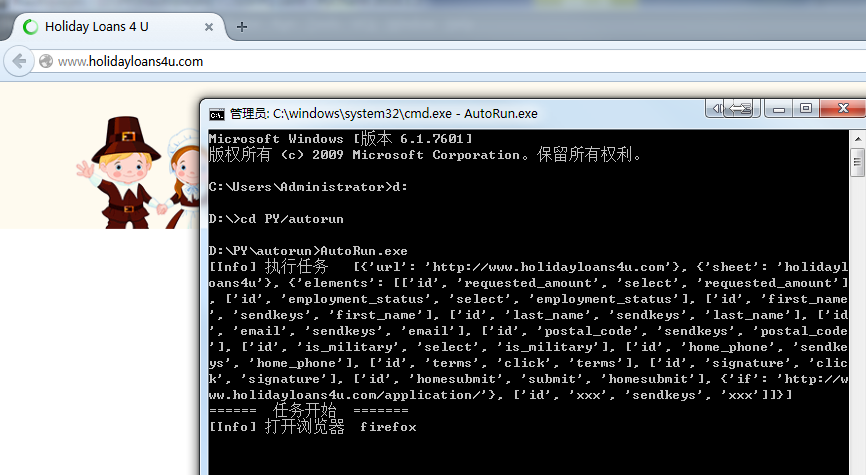


图 运行程序

4. 程序开始运行每个任务，都会创建一个任务同名log文件，每运行一条data中的数据，会往其中插入一个‘1’，当程序再次执行，会根据log中1的数量跳过相应的已执行数据，不重复使用数据。

**五、代码说明**

代码文件有三个，分别是AutoRun.py、setup.py、make\_exe.py，如下图：

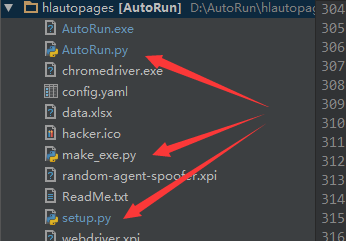


图 代码结构

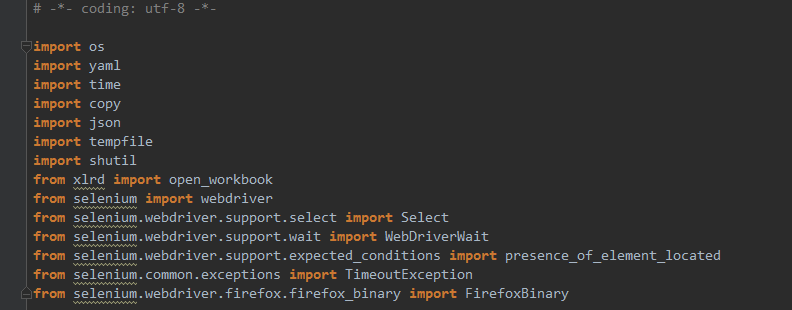
分别讲下各自的作用：

1. AutoRun.py

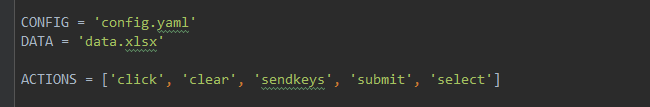
这个是主程序。编码环境是Python2，所需的一些库为：

|  |  |  |
| --- | --- | --- |
| **编码环境** | **所需第三方库** | **用途** |
| Python 2.7.10+ | PyYAML 无版本要求 | 读取yaml文件 |
| xlrd 无版本要求 | 读取excel文件 |
| selenium 2.52~2.53 | 操作浏览器 |
| py2exe 无版本要求 | 将py脚本打包成exe |

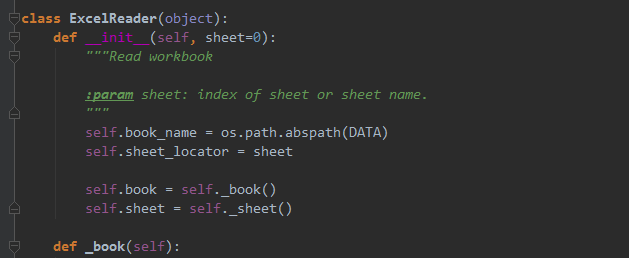
程序解读：



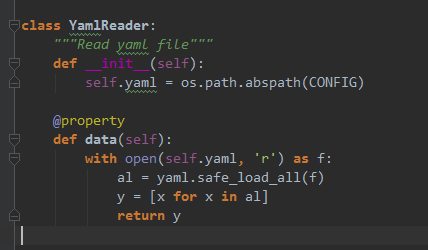
以上一段用于定义编码格式以及导入各种依赖包。



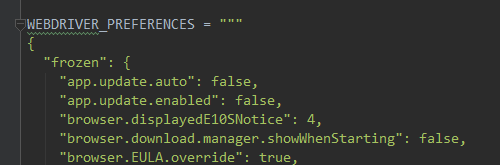
以上是配置文件、数据文件以及支持的动作的常量。

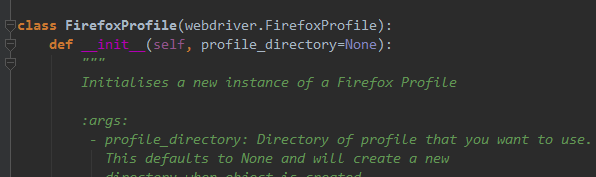


以上ExcelReader类是用于从excel中读取数据，包括上面的一些异常。

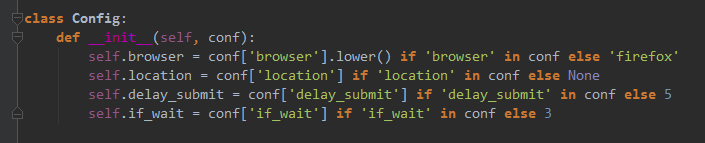


以上YamlReader是用于从yaml文件中读取数据。

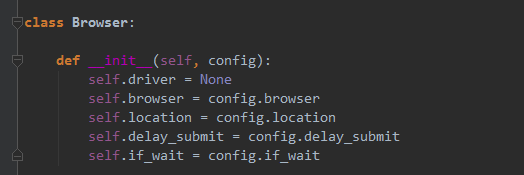




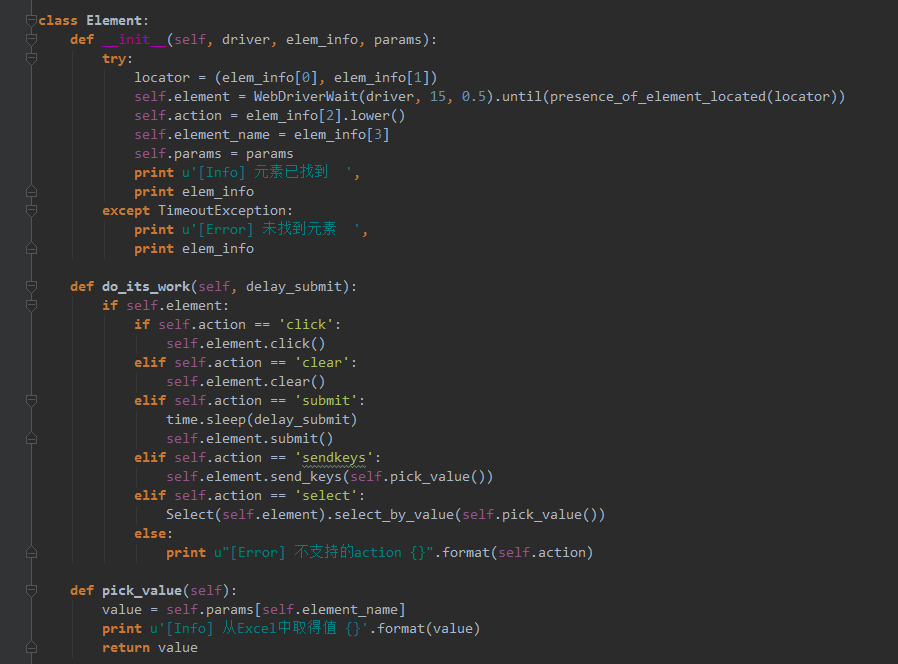
以上WEBDRIVER\_PREFERENCES和FirefoxProfile类是用于将py文件打包成一个exe时必须的解决办法，否则无法打包成一个exe，包括文件夹下的webdriver.xpi也是打开firefox必须的。



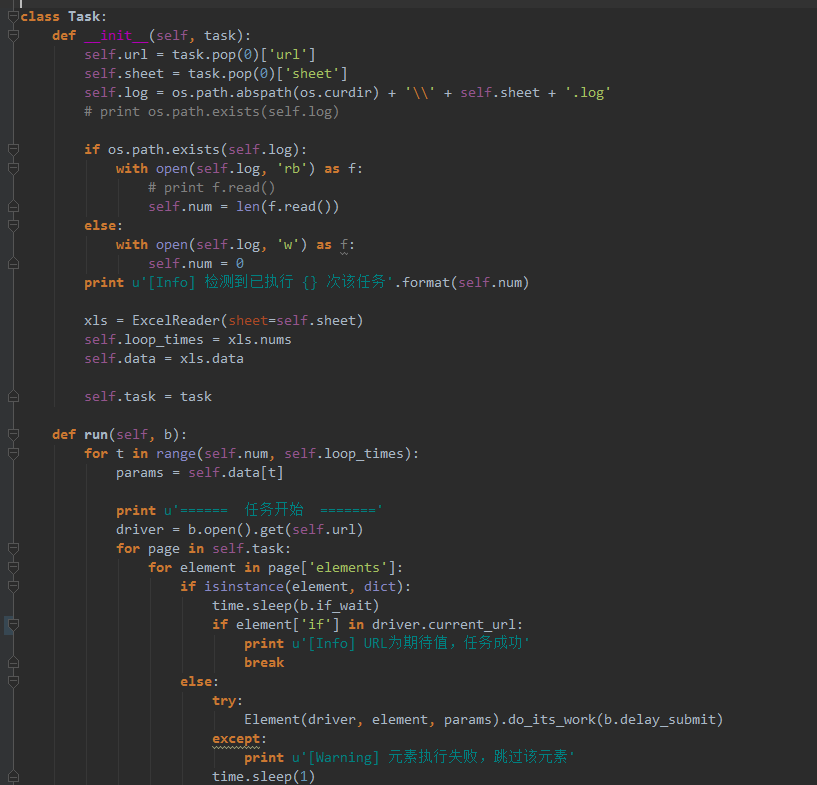
以上Config类是用于定义配置。



Browser类用于打开浏览器，打开网址以及关闭浏览器。



Element类定义元素，以及元素操作对应的处理方法，do\_its\_work()方法中的各个条件对应开头ACTIONS常量中的各个支持的动作。pick\_value()用于从一行数据中根据元素名取到一个值。



Task类表示一个任务，run()方法执行一个任务。会通过log文件判断任务执行过的次数，从而读取excel文件相应数据，从下一项开始执行。



main()函数是入口函数。读取配置，并控制任务执行。

2. setup.py

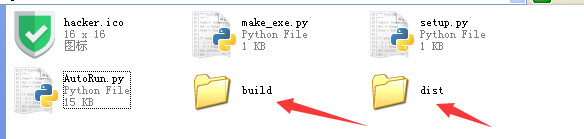
程序用于将AutoRun.py打包成exe程序。此文件并非直接运行。而是通过命令行或者执行make\_exe.py文件运行。

程序解读：



程序很简单，只有这么一段，主要是一些py2exe的设置，执行时，将AutoRun.py打包成一个exe文件，并用hacker.ico图标文件。

出来后的文件夹是这样的：



会产生两个文件夹，build没用，dist中有两个文件：

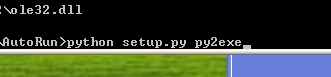


其中AutoRun.exe就是我们的程序，另一个程序是用于在win9x系统里执行的，也是没用的。

3. make\_exe.py

这个文件更简单，就是用来执行cmd命令，调用setup打包的。你也可以写个bat脚本来干这事，或者在命令行手动输入命令执行就行，这个文件可有可无。

命令如下：



执行命令后，等待打包结束，在dist文件夹中就能得到我们的程序了。