

火车浏览器软件

用户使用手册

合肥乐维信息技术有限公司

前言

火车浏览器是一款可视化的自动化脚本工具。用户可以通过设置脚本的点击、判定、循环、跳转等操作，达到自动登录、识别验证码、自动抓取数据，自动下载文件等目的。

每一步流程在设定时都可以自由组合，完全自定义操作，功能无限级扩展。有了火车浏览器，可以写出强大且独一无二的脚本来辅助我们的工作，同时也可以生成单独的 exe 程序分发给他人使用。

研发背景

➤ 网页越来越复杂，需要使用可视化技术

随着网页防采集技术的不断发展，网页结构变得愈发复杂，瀑布流布局、JS 混淆加密、滑动验证码.....这些使得软件对网页操作的处理日趋繁琐。针对这种情况，更加易于操作的可视化技术变得尤为重要。

➤ 大数据时代来临，非技术人员也要掌握操作

大数据时代所引领的新纪元已然开启，这场全世界范围内的大数据浪潮层层包围着每个人的工作和生活，并将迅速渗透到各个行业，因此要抓住时代机遇，我们每个人都要能够掌握大数据相关操作。

综上所述，火车浏览器在研发设计过程中，易化了操作和流程定义，功能完全自由组合，让网页操作变得更加轻松。

核心功能

➤ 可视化提取

火车浏览器完全可视化操作，通过浏览器展现在眼前的内容都能快速精准地提取，只需通过鼠标点选，不懂代码也能轻松学会采集。

➤ 自定义流程

用户自定义的流程可以让操作完全按照用户的需求进行，不再走大众路线，做独一无二的营销推广。

➤ 批量执行

重复执行任务不仅枯燥而且浪费了大量时间，火车浏览器独具特色的项目管理器，让任务按照计划设置批量执行，大幅提升工作效率。

➤ 开发自有程序

用户自定义的脚本都可以生成一个可脱离火车浏览器而单独运行的应用程序，除了自己使用还可以在火车浏览器的官方软件管理平台中进行授权，给他人使用。

➤ 强大的二次开发

火车浏览器提供了丰富的二次开发 SDK，方便用户开发基于火车浏览器的软件。操作简单，扩展无限。

编程语言

火车浏览器的编程语言是 C#，C#综合了 VB 简单的可视化操作和 C++的高运行效率，增强开发效率的同时也致力于消除编程中可能导致严重后果的错误，以其强大的操作能力、优雅的语法风格、创新的语言特性和便捷的面向组件编程的支持成为软件开发的首选语言。

软件资质

火车浏览器软件是合肥乐维信息技术有限公司自主研发而成。火车浏览器的源代码、布局、界面设计、电子文档等均已提交国家版权局登记备案，并已获得著作权审批。

火车浏览器 V6 版软件著作权登记号：2016SR291529



目 录

第一章、火车浏览器的架构	1 -
一、脚本编辑器.....	1 -
二、项目管理器.....	1 -
三、任务运行器.....	2 -
第二章、火车浏览器的功能菜单栏	3 -
一、新建脚本.....	3 -
二、打开脚本.....	3 -
三、数据库配置.....	4 -
四、变量管理.....	5 -
五、验证码识别.....	6 -
六、拦截 http 请求.....	6 -
七、验证码识别工具.....	7 -
八、下载管理器.....	8 -
第三章、脚本编辑器	9 -
一、浏览器.....	9 -
二、逻辑运算.....	16 -
三、元素操作.....	21 -
四、变量处理.....	30 -
五、文件下载.....	37 -
六、数据库操作.....	39 -
七、文件操作.....	40 -
八、其他操作.....	44 -
九、自定义代码.....	48 -
十、互动设置.....	51 -
第四章、项目管理器	53 -
一、任务控制.....	53 -
二、新建分组.....	53 -
三、新建任务.....	53 -
四、选项.....	56 -
第五章、生成 exe	57 -
一、操作指导.....	57 -
第六章、软件适应性	65 -
一、运行环境.....	65 -
二、授权方式.....	65 -
三、软件升级.....	65 -
四、适应性服务.....	65 -
五、技术支持.....	65 -

第一章、火车浏览器的架构

火车浏览器通过脚本编辑器编写脚本，通过任务运行器来运行脚本，通过项目管理器来统筹任务。在登录时分为脚本编辑器和项目管理器两个版块，方便用户需要在需要时启用相应的功能模块。

一、脚本编辑器



(图 1.1 脚本编辑器登录界面)

用途：脚本编辑器用来生成需要执行的脚本，定义脚本的具体流程和细节设置。

在火车浏览器官网 (www.locoyposter.com) 注册一个账号后即可登录。

原理：火车浏览器不再使用分析源代码的方式，而是呈现给用户已经经过浏览器可视化渲染的内容，然后使用特定的网页元素定位和爬虫引擎模拟人打开网页、点击网页内容的思维，实现一系列的逻辑操作。

二、项目管理器

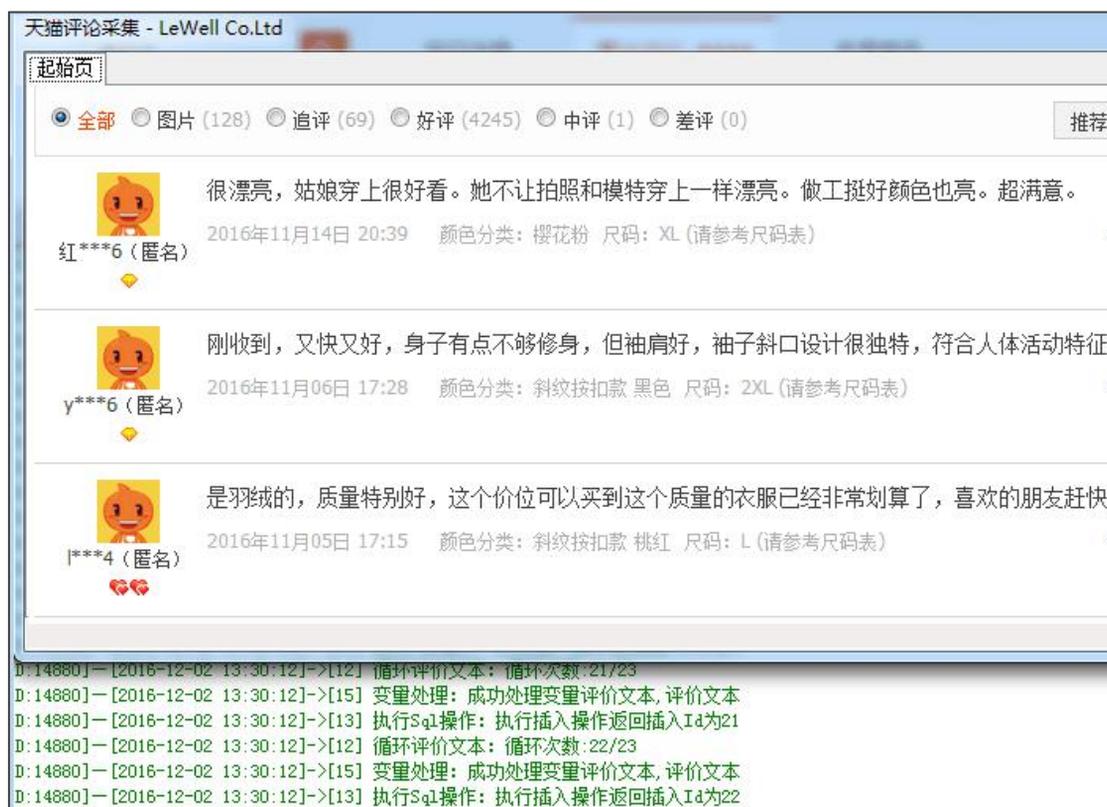


(图 1.2 项目管理器登录界面)

用途：项目管理器可以利用项目模板和脚本整合创建特定的项目，进行长时间、有规划地运行。

原理：项目管理器可以将任务分解为许多小任务同时进行多进程的操作，比如批量搜索，原本需要查询上万个关键词，如果是单个运行会速度较慢且容易出错，启用项目管理器后，项目管理器只需要分配发送不同的关键词给运行器，就可以修改运行器中的关键词反复进行批量搜索了，还可以关闭运行出错的任务再次运行。

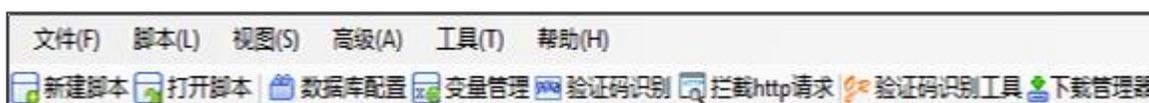
三、任务运行器



(图 1.3 TasksRunner 运行界面)

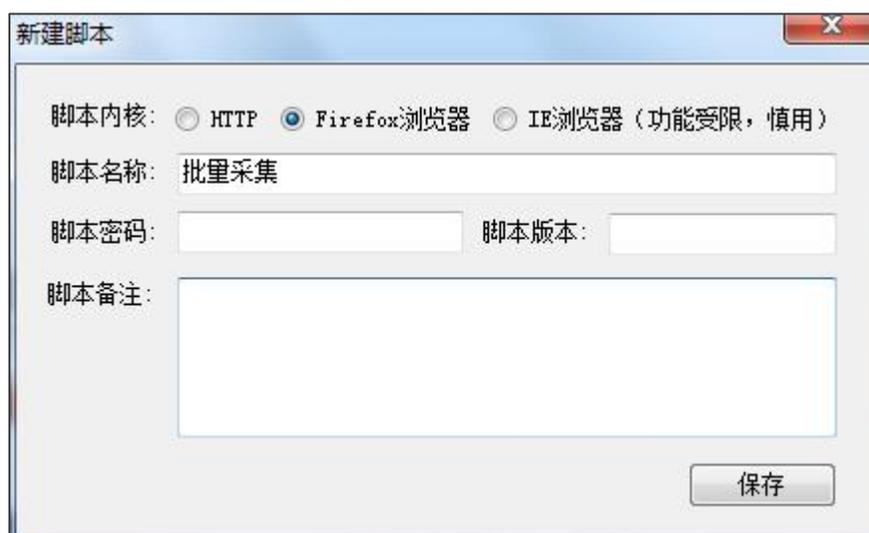
脚本完全测试和项目管理器运行任务，都是使用任务运行器 TasksRunner 来完成的，如图 1.3 为电商评论采集运行界面。

第二章、火车浏览器的功能菜单栏



(图2 火车浏览器功能菜单栏)

一、新建脚本



(图2.1 新建脚本界面)

在新建脚本时，如图 2.1 有三种脚本内核可选，区别如下：

HTTP：为无内核版本，不再是模拟网页打开，而是使用 http 请求进行操作，需要打开网页才能进行的操作都不能再使用，会显示灰色不可选状态。

Firefox 浏览器：为火狐 45 版本的内核，模拟真实网页打开操作，所有步骤都可以使用。

IE 浏览器：为 IE 的内核，也是模拟网页真实打开然后进行操作。

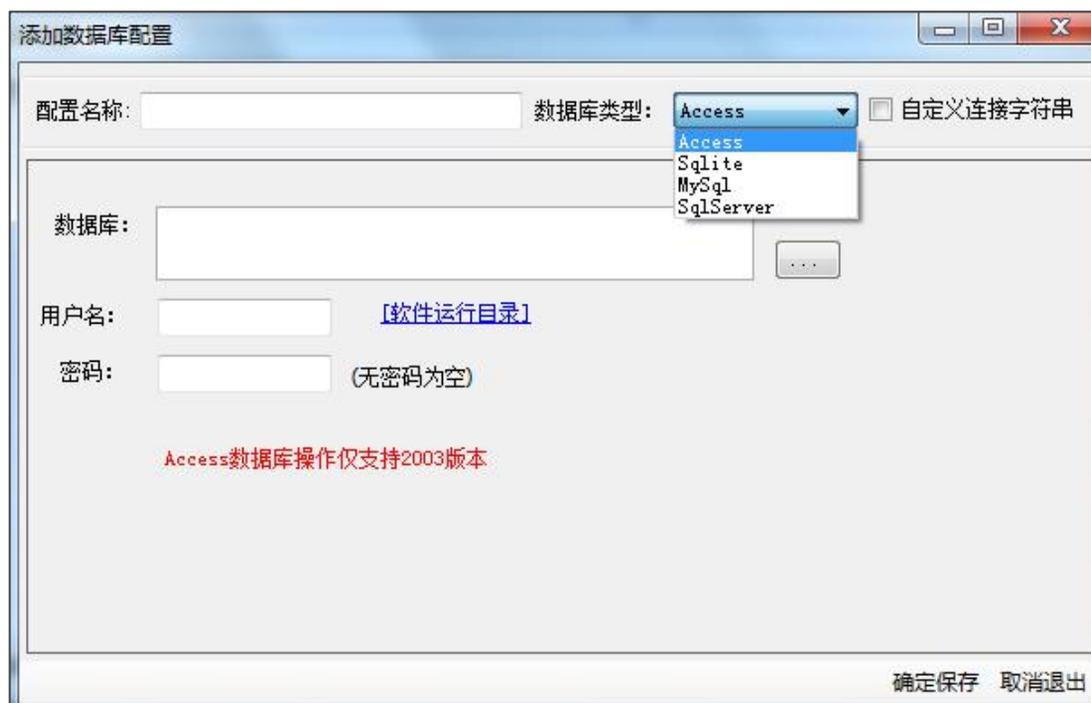
除内核选择外，新建脚本时还有脚本名称、脚本密码、脚本版本、脚本备注由用户自定义，便于区分和加强安全性。

二、打开脚本

即将本地脚本选择性显示在侧边的编辑列表中，方便用户进行查看和编辑。点击打开脚本，可显示脚本编辑列表中未显示的脚本，双击该脚本名称，即可在脚本编辑框中显示。

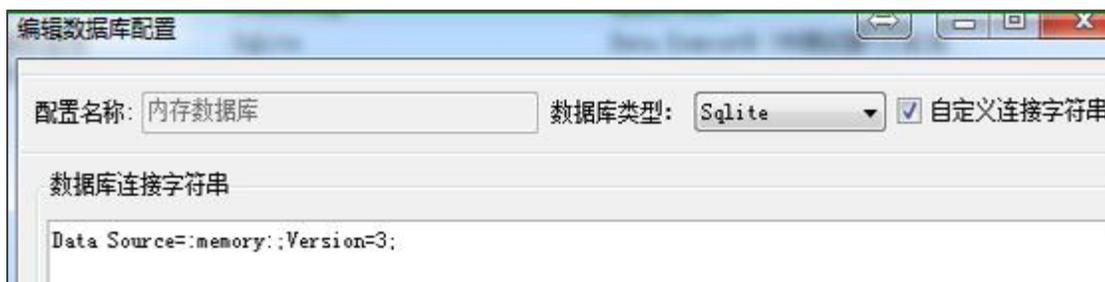
三、数据库配置

当用户在脚本中使用执行 sql 语句时，需要添加数据库配置。配置好之后执行 Sql 操作，可以实现数据的采集入库。



(图 2.3 数据库配置界面)

- 1、**数据库类型**：支持 2003 版本的 access、sqlite、mysql，sqlserver。
- 2、**自定义连接字符串**:如勾选该项则为自定义连接方式，不勾选则为直接填写用户名密码或者是路径即可连接。
- 3、**创建内存数据库**:运行过程中创建虚拟内存 sqlite 数据库，不会产生数据库文件，通常配合火车浏览器中的“导出 excel”功能使用。
- 4、**注意事项**：
 - 4.1 若是保存少量数据通常选择 sqlite 数据库，因为 sqlite 不需要安装数据库环境，并且在火车浏览器中可以自动建立数据库，较为方便。
 - 4.2 使用内存数据库，只要设置好连接字符串，就不需要再做其他改动，至于 sql 语句的操作跟其他数据库是一样的。注意：一个脚本中只能使用一个内存数据库。



(图 2.3.4.2 使用内存数据库)

4.3 若是 mysql 入库的时候出现乱码，需要首先确定 mysql 是否是 utf8 编码，然后注意数据库配置处也选择 utf8 数据库。若是显示连接失败，请先检查该 mysql 是否开启了远程连接权限。

4.4 具体操作：填写数据库配置，有图 2.3.4.4 所示两种方式。脚本中建立一个创建数据库表的语句，然后运行该语句就能自动创建一个 sqlite 数据库文件。



(图 2.3.4.4 数据库配置方式)

四、变量管理

变量管理用于添加和管理脚本中使用的变量，在单步测试运行过程中，变量管理也可以看到值的变量变化。

1、**变量属性**：变量管理里可以建立三种属性的变量。

文本型变量：存放单条数据。

list 变量：存放多条数据，类似于数组。

计数器变量：只可以存放数字。

2、**注意事项**：

在建立变量时，若是如图 2.4.2 勾选只读，则在脚本中就不能再改变其中的内容，不能对其中的内容进行二次操作。所以单次提取等选择保存的变量时就不能选择只读变量。



(图 2.4.2 新增变量)

在单步测试脚本时，变量管理中的变量值会显示随着步骤的运行而变化。

五、验证码识别

在采集或其他批量的操作中遇到验证码，可以配置验证码自动识别来节省人力，在验证码配置中需填写打码平台的账号密码等登录信息。

自动打码有若快、联众、优优云等多种打码方式可供选择，用户可以结合验证码识别的功能来使用。

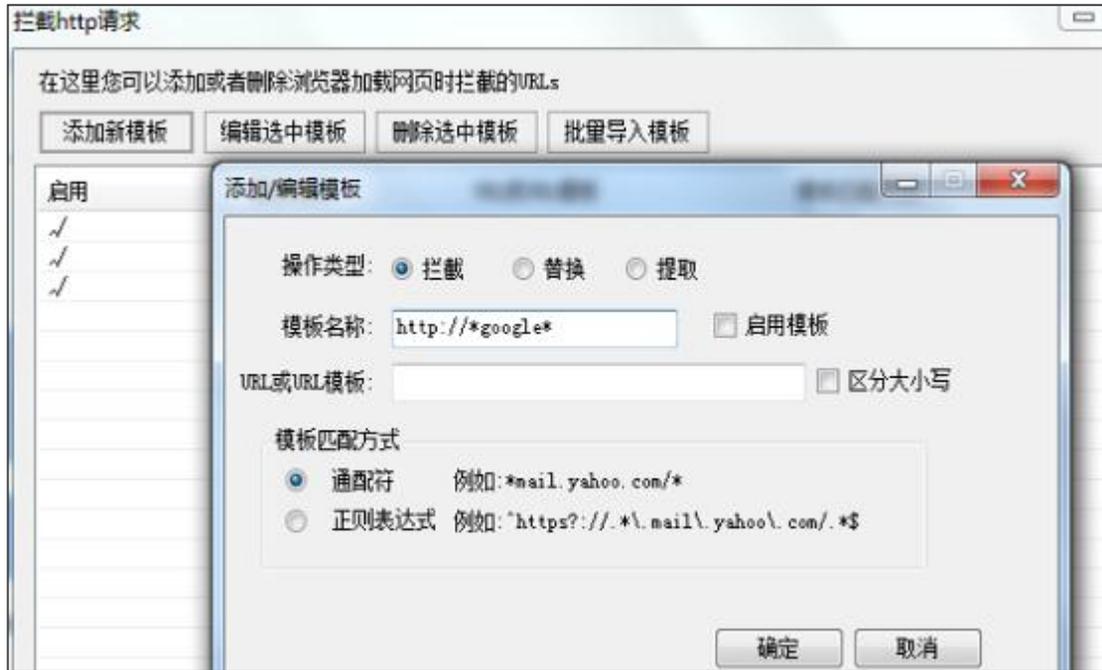
验证码识别参考地址：<http://bbs.locoyposter.com/post38911.html>

六、拦截 http 请求

在操作中打开网页时，会加载很多的网址，其中有些网址的加载比较慢会拖慢操作速度，而火车浏览器的“拦截 http 请求”功能就可以配置拦截模板，拦截掉符合模板匹配规则的网址。

1、操作指导

1.1 我们可以在拦截 http 请求中添加模板，如图 2.6.1 中，这个模板就是拦截所有包含 google 这个字符的网址，* 是通配符的意思。



(图 2.6.1 拦截 http 请求的模板编辑)

1.2 如何查看页面中加载的网址？首先用户可以在视图中打开查看 http 请求，将加载出的 url 列表中的内容清空（右键 url 列表有清空选项），然后运行打开网页，就会加载出打开网页时所有加载的列表。

1.3 这其中的网址可以直接右键添加至拦截列表，如果相似的网址很多，则可以添加通配的拦截模板，拦截同一类型的网址。

1.4 有时用户并不清楚究竟是哪一个网址拖慢了网站的加载，针对这种情况，可以正式运行一下脚本，在打开网页时，若是某一个网址加载比较慢，那么在运行窗口的左下角会显出该网址的名称，我们就可以对照加载的 URI 列表，来添加相应的模板。

2、注意事项

2.1 此功能在正式运行时才会呈现效果。

2.2 不要拦截掉操作中需要的网址内容。

七、验证码识别工具

即火车浏览器中的 ocr 简易验证码识别系统，如图 2.7 可通过网络、本地加载或提取图片，对图片进行亮度、灰度、去噪等预处理后可以自动识别图片中的验证码，达到预期效果。注意事项：该功能仅可处理普通的验证码，对于一些变形、扭曲、旋转等类型的验证码，请使用打码平台。



(图 2.7 验证码识别工具)

八、下载管理器

即显示火车浏览器中正在下载的文件，并可进行搜索和清除列表等操作。

第三章、脚本编辑器

火车浏览器的脚本编辑器中包含以下可选流程，下面逐个介绍其功能。



(图 3 脚本编辑器流程)

一、浏览器

这是用户通过软件可对浏览器进行的一系列操作，与人工手动操作无异，包含打开网页、设置 cookie、刷新页面、设置窗口大小，后退一步等。



(图 3.1 浏览器相关设置)

1、标签页操作

标签页操作可以用来创建、切换以及关闭指定的标签页，实现在一个脚本中同时打开多个页面。

1.1 操作指导



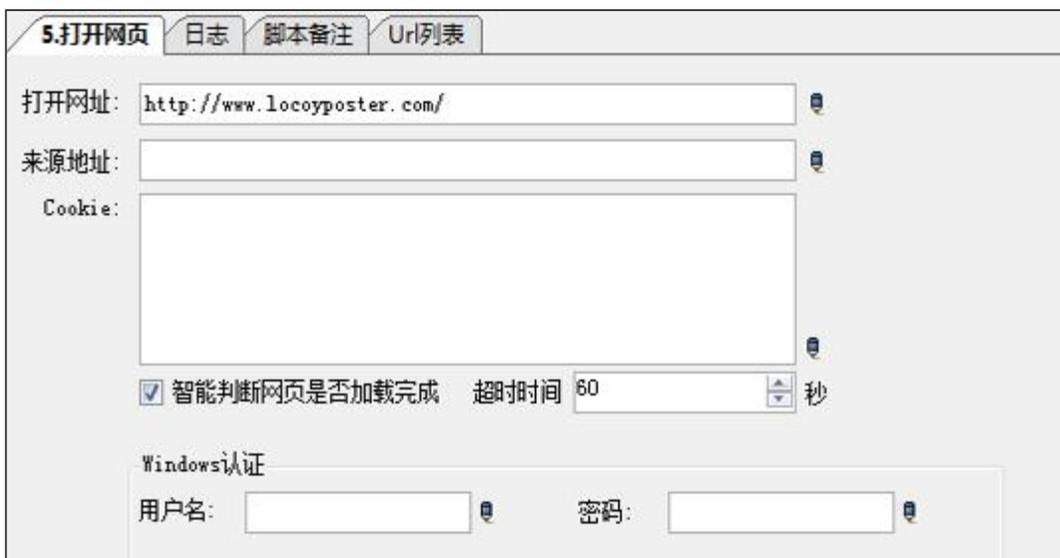
(图 3.1.1.1 创建标签页)

(1) 创建标签页：可以填写固定的标签页名称，也可以调用变量。注意：不可创建同名的标签页。

(2) 切换到指定标签页：首先需要了解，脚本中步骤操作的是当前页的元素，如果需要操作某个标签页，就要将操作焦点切换到该标签页。注意：创建一个新标签页时，会自动切换焦点到该新标签页。

2、打开网页

打开网页是用来模拟网页打开的步骤，网页打开后可以在网站中进行数据抓取以及点击模拟操作等。



(图 3.1.2 打开网页界面)

2.1 操作指导

(1) 打开网址：填写需要打开的网址。(可以调用文本型变量：将需要打开的网址保存在文本型变量中即可调用。注意：这里只能调用文本型变量，如果是 list

变量，则需要循环遍历 list 变量后保存在文本型变量中才可调用。)

(2) 来源地址：模拟 http 请求的 referer 网址，即告诉服务器我们是从哪个页面链接过来的，此处一般可以不用填写。

(3) Cookie：可以用 fiddler 抓取页面的 cookie，保存在文本型变量中，直接调用该变量。

(4) 智能判断网页是否加载完成：勾选则可启用该功能。

(5) 超时时间：设定一个时间，若是在此超时时间内，网址仍没有加载完成，就强制结束打开网页这一步，继续运行下面的步骤。

(6) Windows 认证：访问页面时有时会需要身份认证，需填入用户名密码，比如在登录路由器时，就需要使用这个功能。

3、Cookie 操作

即在火车浏览器中进行 cookie 的导出、导入或者是清除 cookie。

3.1 操作指导

(1) 导出 cookie 到文本：将火车浏览器中 cookie 导出并保存到文本中。

(2) 导入文本 cookie：将文本中的 cookie 导入到火车浏览器中。(注意：此时文本中的 cookie 必须是火车浏览器导出的 cookie 格式。)

(3) 导入字符 cookie：导入火车浏览器中接收的 cookie 格式。

(4) 清除 cookie：即清除火车浏览器中的 cookie。



(图 3.1.3.1 导出 cookie 到文本)

3.2 使用场景

(1) 网站需频繁登录：遇到此类网站，可以将登录后的 cookie 导出。然后下次运行时，可以直接导入登录后的 cookie，避免再次重复登录步骤。

(2) 直接退出登录账号：想要直接退出，可以设置清除 cookie，则浏览器中的登录信息都会清除。

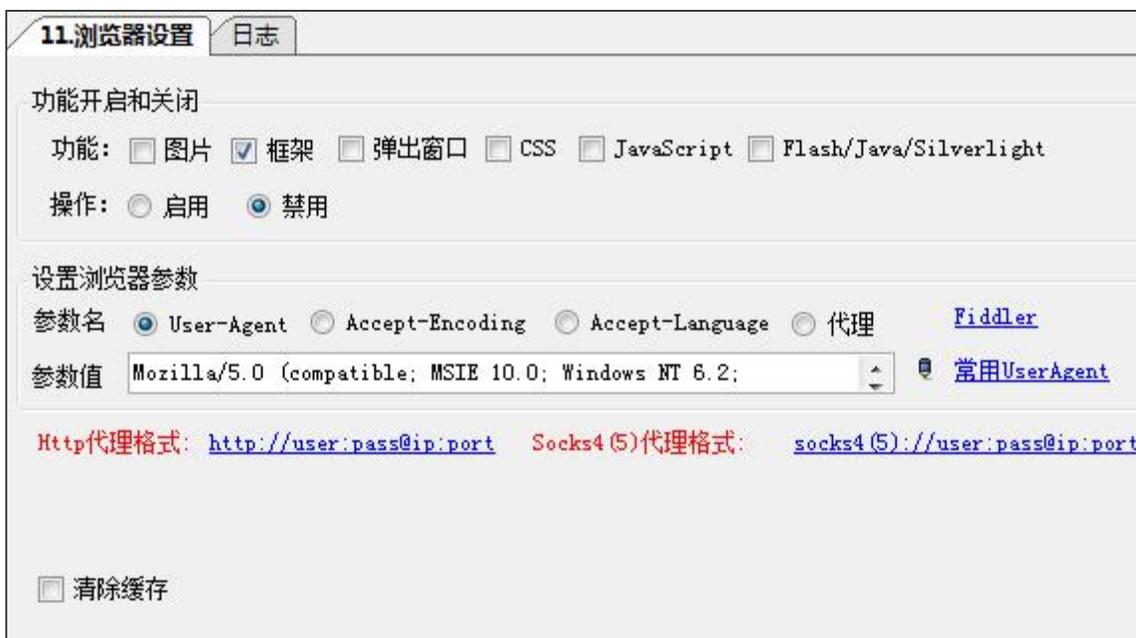
注意：如果是使用 fiddler 抓包的 cookie，则不能在本步骤使用。可以在打开网页中调用 fiddler 抓包的 cookie。

4、浏览器设置

对浏览器的 UserAgent、代理等一些基本参数进行设置。功能开启和关闭支持多选，如图 3.1.4.1。

4.1 操作指导

(1) UserAgent：模拟浏览器信息。可以选择常用 useragent 中的示例，也可以自己找真实的 UA 填写。这里只可以调用文本型变量。如果存储大量的 UA 在 list 变量中，需要循环遍历 list 保存至文本型变量，然后这里再调用该文本型变量。



(图 3.1.4.1 浏览器设置界面)

(2) 代理：这里支持 http 以及 Sock4、Sock5 类型的代理。

格式形如：http://user:pass@ip:port (socks4://user:pass@ip:port) 如果没有用户名密码，那么格式形如 http://:@ip:port，例如 http://:@127.0.0.1:8888。

如果 IP 代理地址存放在文本中，可以使用文本读取，读取 IP 地址保存至 list 变量，然后循环遍历 list 变量保存至文本型变量，再调用该文本型变量用于设置浏览器代理，这里需要注意代理的格式需要处理成火车浏览器默认的格式。

(3) 弹出窗口：勾选则允许自动弹出新的标签页。

5、滚动条设置

滚动当前页面，可设置滚动到顶部、滚动到底部、滚动到指定元素位置的等。

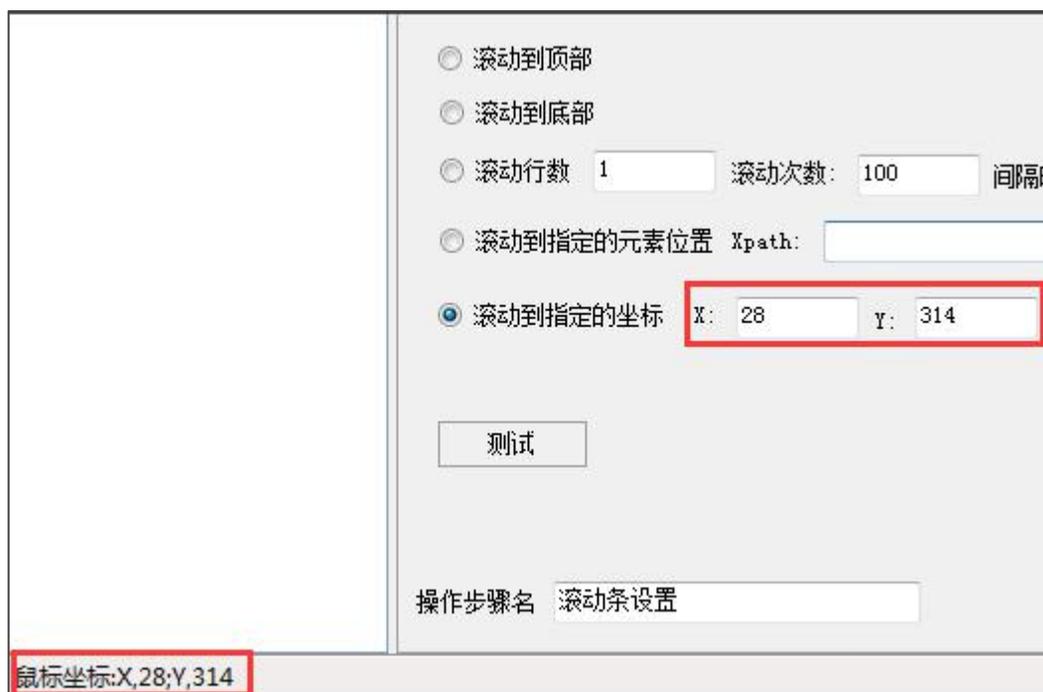


(图 3.1.5 滚动条设置界面)

5.1 操作指导

(1) 滚动到指定的元素位置：需要找到指定位置的 xpath。

(2) 滚动到指定的坐标：此功能需要结合“设置浏览器大小”功能使用，先确定界面大小，这样在运行过程中，指定元素的坐标不会发生变化。在设置浏览器大小后，将鼠标放在滚动目标元素处，火车浏览器左下角会出现该元素的 X、Y 坐标，将其写入步骤即可，如图 3.1.5.1。（注意：一定要先设置浏览器大小，再取元素坐标。）



(图 3.1.5.1 滚动到指定的坐标)

6、刷新页面

即刷新当前页面。

7、后退一步

即后退到上一个操作界面。

8、网页截图

在当前页面截图并将图片保存在本地。截取方式可以选择截取整个页面、按照区域坐标截取，按照 xpath 截三种。

8.1 操作指导

(1) 截取整个页面：指定图片保存路径、文件名保存格式，保存文件路径(是将图片保存在本地的路径再保存至变量，此处可以为空)。

注意：保存格式可以选择原文件名以及变量、日期等命名方式。格式结尾要指明图片格式后缀，如 jpg、png 等图片格式。如文件名保存格式中填写的名称呈现绿色，则需要在前面上加上转义字符“\”，如图 3.1.8.1-1。

109. 网页截图

网页截图方式： 截取整个页面 截取指定区域 截取指定Xpath

指定区域坐标：X 0 Y 0

指定Xpath：

文件保存文件夹：[软件运行目录]\截图 [软件运行目录]

文件名保存格式：{-var. 时间-}.jpg 文件保存的格式结尾需要写文件名后缀

保存文件路径至： 保存路径可以为空

(图 3.1.8.1-1 网页截图)

(2) 截取指定区域：可以借助坐标来截图某一元素。首先找到元素的任意区域 xy 坐标点，如图 3.1.8.1-2，鼠标放在该区域中，则在火车浏览器左下角，会出现相应坐标，将坐标填在步骤里，则可截取该元素的图片。

注意：在使用此方式时，依旧需要结合“改变浏览器大小”步骤，设置浏览器界面大小，以免运行过程中界面大小变化，导致设置的坐标无效。



(图 3.1.8.1-2 截取指定区域)

(3) 截取指定 xpath : 截取指定 xpath 元素内的图片。首先要分析出需要截图区域的 xpath , 然后将 xpath 填写在步骤中, 即可截取该图片。

9、页面信息

如图 3.1.9 所示。



(图 3.1.9 页面信息界面)

10、浏览器窗口大小

此功能可以更改浏览器显示的窗口大小。在前面的滚动到指定坐标和截取指定区域功能中已经介绍过, 结合此功能可以保证页面中的 x、y 坐标不变, 能够正确点击到设定的坐标。

11、弹窗信息

操作中如出现 alert 弹窗，则无法通过 xpath 获得内容，所以可以通过此步骤获取弹窗信息。

注意：只有当整个脚本在右键测试运行，或 exe 以及项目管理器中正式运行时，才能取得 alert 中的内容。单个步骤右键测试运行时，是无法获取其中内容的。

二、逻辑运算



(图 3.2 逻辑运算流程)

1、条件判断

是对当前页面源码、xpath、以及简单的 js 代码进行一系列判断。

1.1 操作指导

- (1) 源代码包含：判断当前页面源代码是否包含某元素。
- (2) 源代码等于：判断当前页面源代码是否完全等于某元素。
- (3) 存在 Xpath：判断当前页面是否包含该 xpath。
- (4) Js 代码判断：进行一些简单的 js 代码片段判断，如图 3.2.1。



(图 3.2.1 条件判断操作)

(5) 逻辑或({|}): 多个条件有一个满足即为 true, 相当于 js 中的 “||”, 例如, 在存在 xpath 元素中写作: //a[@id="user"]{|} //a[@id="name"]。

2、变量比较

对字符串变量, List 变量, 计数器变量进行比较。

2.1 操作指导



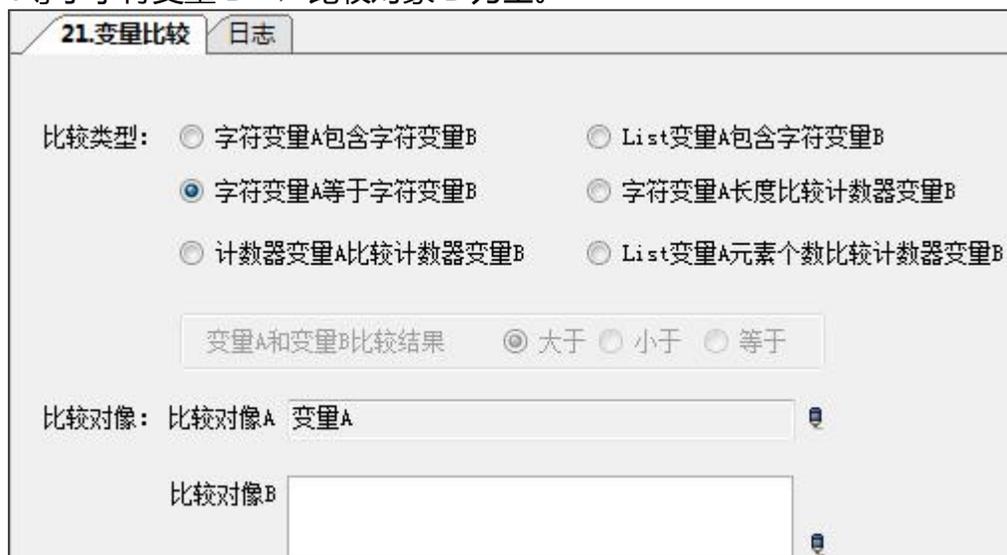
(图 3.2.2.1 比较类型界面)

(1) 比较类型: 从图 3.2.2.1 可以看出, 在比较类型的六项中, 有三项是包含“比较”一词的, 因此变量 A 和变量 B 的比较结果中的大于、小于、等于是针对这三项的, 其余三项类型不受此影响。

(2) 比较对象: 比较对象 B 可以直接写入字符串, 也可以选择变量。

2.2 使用场景

如果判断变量是否为空, 可以按照图 3.2.2.2 的设置, 比较类型选择“字符串 A 等于字符串 B”, 比较对象 B 为空。



(图 3.2.2.2 判断变量是否为空)

3、循环

使用循环后，循环操作中的步骤可以重复循环运行。

3.1 操作指导

(1) 无限循环：即操作可以无限循环，如果使用无限循环，那么循环步骤中一定要有条件判断可以结束循环，否则会造成程序的死循环。

(2) 循环指定次数：指定循环的运行次数，也可以使用计数器变量。

(3) 循环遍历 list：脚本步骤有很多只能调用文本型变量，比如打开网页、写值等。这时如果有多个值是存在 list 变量中的，就可以使用循环遍历 list 保存至文本型变量，然后循环中的步骤再调用该文本型变量，list 变量的值就可以循环调用了，如图 3.2.3.1。



(图 3.2.3.1 循环指定 list)

(4) 从数据库查询数据：数据库配置中配置数据库路径，然后在“查询语句”框中写上 select 语句(必须要写完整的列名，不可以使用*代替)，循环过程中，就会将列中的值自动赋值给与列同名的变量。比如：select title from lianxi，就会将 title 列中的值自动赋值给文本变量 title。

4、跳出本次循环

只能在循环内添加此步骤，作用是跳出当前正在运行的循环，不再运行本次循环内剩余步骤，直接运行下一次循环。(注意与后面即将讲到的“结束当前循环”进行区分)

举例介绍：比如定义了循环次数为 4 次，在第 2 次时，如使用此步骤，则跳过循环内该步骤下面的步骤，直接进行第 3 次循环，直到结束。

5、结束当前循环

只能在循环内添加此步骤，作用是结束整个循环操作。

举例介绍：比如定义了循环次数为 4 次，你在第 2 次时，使用此步骤，则直接结束整个循环，不再进行第 3 次、第 4 次，而开始运行循环外步骤。

6、暂停等待

加载一个新页面是需要一定时间的，为了保证页面的正常打开，需要在加载新页面的步骤下加上暂停等待步骤。

6.1 操作指导



(图 3.2.6.1 暂停等待步骤)

(1) 等待指定时间：指定时间的单位是毫秒（1 秒=1000 毫秒）。使用计数器变量的值是将等待时间放在计数器变量中，单位为毫秒。随机时间，即等待一个随机时间。

(2) 等待指定的元素出现：以指定元素作为页面加载完全的标志，该元素出现，即代表暂停等待就结束了。

(3) 超时时间：如果在超时时间内，等待的元素 xpath 还未出现，就结束暂停等待步骤，继续运行下面的步骤。

6.2 使用技巧

很多时候页面加载的时间都是不确定的，使用等待指定的时间有时候会很耗时，这时要提高脚本效率，可以使用循环加条件判断的方式来做暂停等待。

具体操作介绍：循环中加上以指定元素 xpath 作为条件的条件判断步骤，如果该指定元素出现，为 true，就结束循环；未出现，为 false，则暂停等待然后循环继续条件判断。这样操作不用固定设置时间，能够提高操作的灵活性。

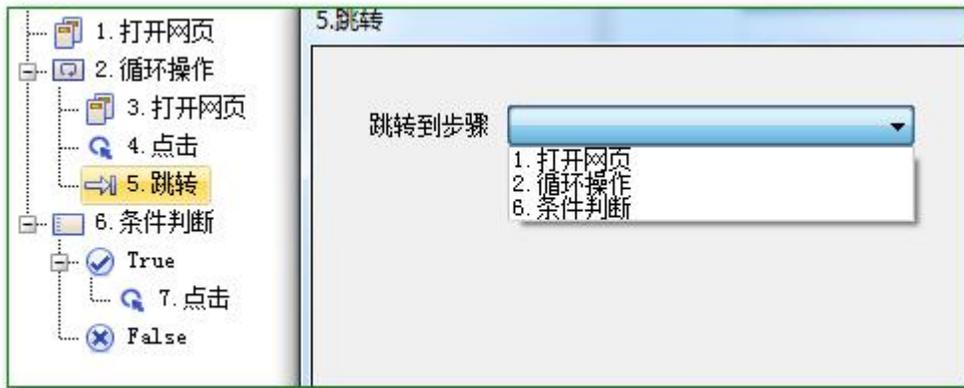


(图 3.2.6.2 循环加条件判断)

7、跳转

从当前跳转到其他步骤。

注意：只能跳转到第一层步骤，不能跳转到嵌套的步骤中，比如循环内、条件判断内的步骤等。如图 3.2.7，是无法跳转到“4.点击”步骤的。



(图 3.2.7 跳转到步骤)

8、结束流程

功能是结束脚本的运行。

13.结束流程

运行结果

<input checked="" type="radio"/> 成功，完成预期的整个流程	<input type="radio"/> 失败，用户没有登录
<input type="radio"/> 失败，数据提取和预期不符	<input type="radio"/> 失败，用户名或密码错误
<input type="radio"/> 失败，被目标服务器封IP	<input type="radio"/> 失败，登录帐号被禁用
<input type="radio"/> 失败，目标服务器内部异常	<input type="radio"/> 失败，当前数据已被删除
<input type="radio"/> 失败，Http请求返回异常	<input type="radio"/> 失败，未知异常

提示信息

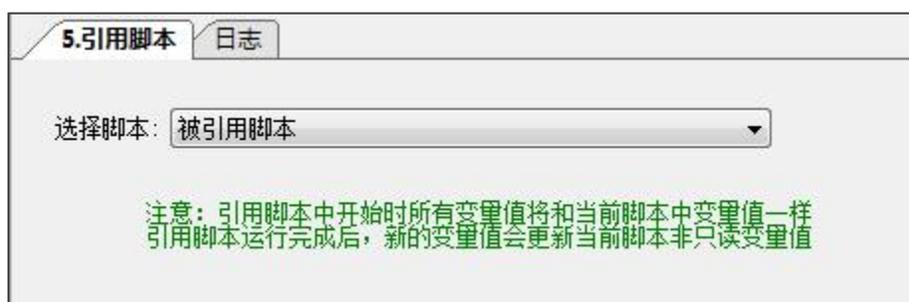
操作步骤名

(图 3.2.8 结束流程操作)

结束流程选择不同的运行结果，最后项目管理器会根据返回结果来进行统计脚本运行的情况，方便监测脚本运行。（火车浏览器 SDK 也可以调用结束流程返回值进行操作。）

9、引用脚本

在当前脚本中引用其他脚本运行。比如如果脚本较长时，我们可以将脚本按功能拆分成子脚本，以后方便管理，然后分开引用脚本运行，有利于释放内存。



(图 3.2.9 引用脚本)

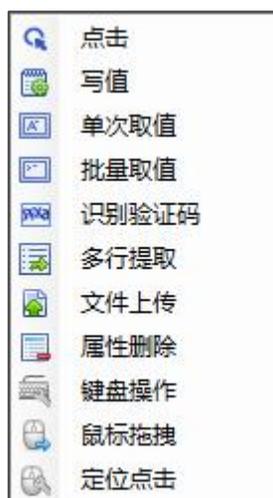
9.1 操作指导

(1) 当前脚本中的变量值会传递给被引用脚本中的同名变量，在被引用脚本中如果变量值被改变了，则在引用脚本运行结束后，将值再传递回当前脚本中的同名变量。

(2) 生成 exe 的时候，将被引用的脚本放在 exe 目录下的 projects 文件夹下，则能正常调用。

三、元素操作

元素操作是对元素进行点击、写值、取值等进行一系列的操作。



(图 3.3 元素操作)

1、点击

执行点击事件，模拟真实的点击步骤。

1.1 操作指导



(图 3.3.1.1 点击)

(1) xpath 提取规则 :填写输入框的 xpath, 可以使用火车浏览器的取 xpath 工具, 也可以在火狐中分析 xpath 然后填写 (仅可写一行)。

(2) 备选提取规则 :在"xpath 提取规则"中的 xpath 在当前界面找不到时, 则会使用备选规则。备选规则中 xpath 如果有多条, 则在输入时一行一条。

(3) 测试查找元素 :测试查找 "xpath 提取规则" 中 xpath 是否存在。

(4) 索引 :当前页面符合 "xpath 提取规则" 中 xpath 元素的数量。

(5) 执行事件 :不同的点击事件是有区别的, 可选 Click、Focus、Onchange 等, 相关知识点可以在 W3school 搜索了解。

(6) 智能判断网页是否加载完成 :判断点击之后, 当前网页是否加载完成。

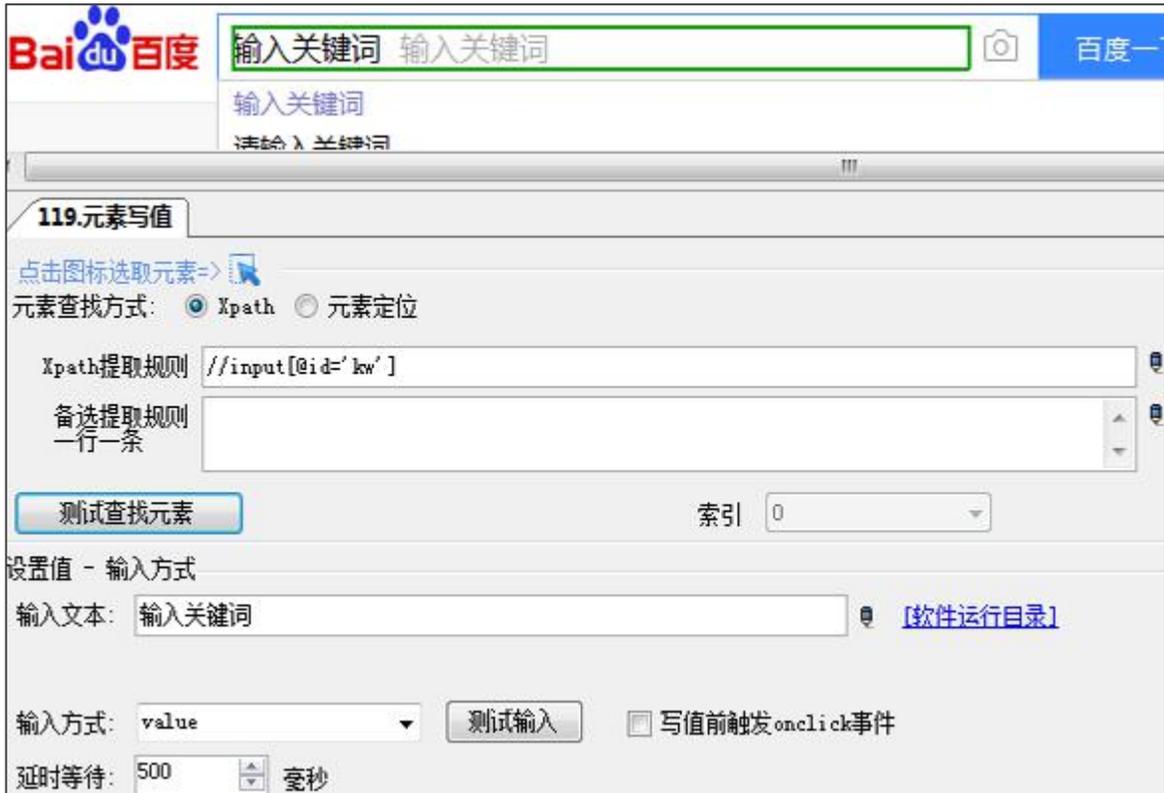
(7) 操作后延时等待 :即完成点击事件后等待的时间, 以确保点击事件完成, 最少设置为 500 毫秒。

2、写值

执行写值事件，模拟真实的写值步骤。

2.1 操作指导

以百度为例介绍写值的用法。



(图 3.3.2.1 写值操作)

(1) xpath 提取规则 :填写输入框的 xpath, 可以使用火车浏览器的取 xpath 工具, 也可以在火狐中分析 xpath 然后填写 (仅可写一行)。

(2) 备选提取规则 : 在"xpath 提取规则"中的 xpath 在当前界面找不到时, 则会使用备选规则。备选规则中 xpath 如果有多条, 则在输入时一行一条。

(3) 测试查找元素 : 测试查找 "xpath 提取规则" 中 xpath 是否存在。

(4) 索引 : 查找当前页面符合 "xpath 提取规则" 中 xpath 元素的数量。

(5) 输入文本 : 输入文本框中的值。

(6) 输入方式 : 三种可选。

①value 是 form 控件当前的值, 遇到的输入框节点是 input 则可选 value。
如图 4.3.2.1。

②innerHTML 是指标签内的 html, 遇到的输入框节点是/html/body, 选用 innerhtml。

③textcontent 是对输入内容进行处理后再赋值给 innerhtml。

关于三种输入方式的解释可以查阅相关资料参考一下。

(7) 写值前触发 onclick 事件 : 如果有的输入框需要触发点击 onclick 事件, 则勾选该项。

3、单次取值

执行单次提取，提取数据。

3.1 操作指导

(1) xpath 查找



(图 3.3.3.1-1 单次取值)

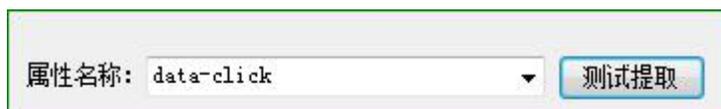
①xpath 提取规则：填写输入框的 xpath，可以使用火车浏览器的取 xpath 工具，也可以在火狐中分析 xpath 然后填写（仅可写一行）。

②备选提取规则：在“xpath 提取规则”中的 xpath 在当前界面找不到时，则会使用备选规则。备选规则中 xpath 如果有多条，则在输入时一行一条。

③测试查找元素：测试查找“xpath 提取规则”中 xpath 是否存在。

④索引：查找当前页面符合“xpath 提取规则”中 xpath 元素的数量。当索引值为多个，可以勾选随机选取，则可随机提取一个索引中值。

⑤属性名称：提取不同属性中的值，也可以自行手动填写属性名称，如图 3.3.3.1-2。



(图 3.3.3.1-2 自定义属性名称)

⑥保存至变量：可以保存至文本型变量（新建立的文本变量不可勾选只读）。如勾选了“获取所有数据保存至 List 变量”，就可以保存至 list 变量。当用户提取的数据有多个时，例如一个页面有多个网址等，便可建立一个未勾选只读的 list 变量，即可保存多条数据。

(2) 元素定位

用标记元素标记需要获取的内容，然后将需要的条件导出到右边框中，选择需要提取的属性名称。



(图 3.3.3.1-3 元素定位查找元素)

(3) 从源代码中提取

使用正则匹配从源代码中直接提取内容。



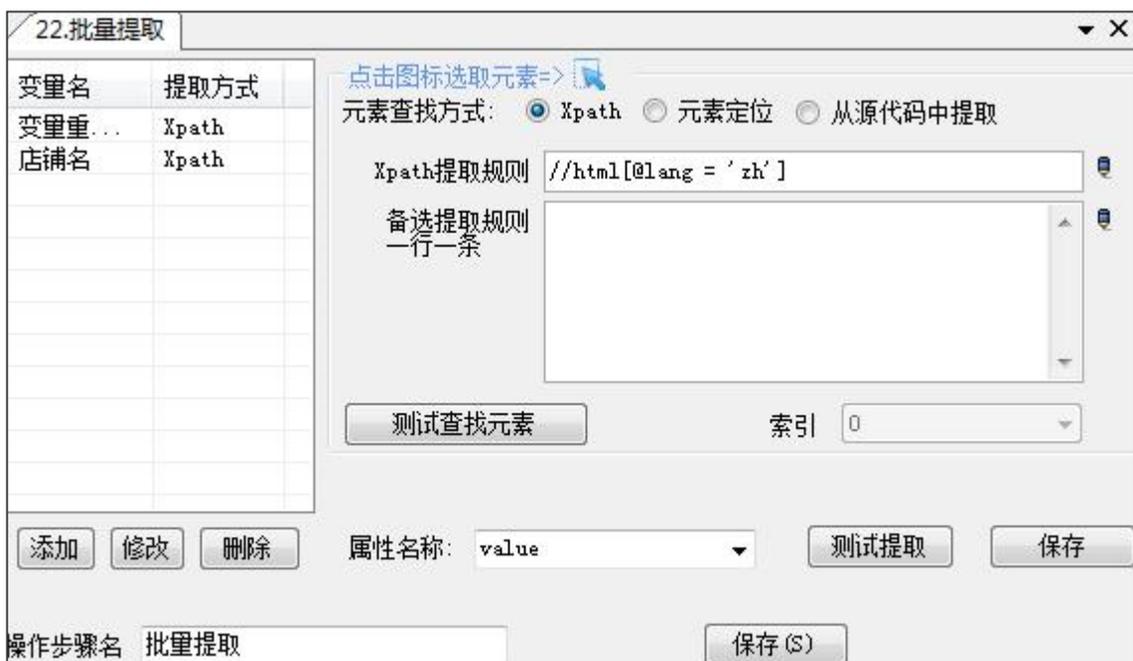
(图 3.3.3.1-4 从源代码提取元素)

4、批量取值

相当于多个单次取值的集合，批量执行单次取值后提取数据。

4.1 操作指导

操作方法参照单次取值，但需注意：批量取值只能保存至文本型变量，不能保存至 list 变量。在操作中“保存”步骤有两步（如图 3.3.4.1），需要逐一进行保存，以免编辑丢失。



(图 3.3.4.1 批量取值界面)

5、识别验证码

用于识别提取元素中的验证码信息，使用前应添加所需的验证码模板。

5.1 操作指导

(1) 验证码文件格式：为 jpg 和 png 两种。

(2) 从缓存提取验证码图片：该项是否要勾选取决于用户的打码方式，要求速度，如对接打码平台可以勾选，如是自行制作的打码插件则需要质量比较好的图片，此项可不勾选。



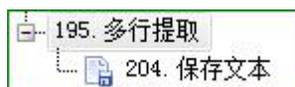
(图 3.3.5.1 验证码识别)

6、多行提取

多行提取是对表格型数据的简便提取方式。

6.1 操作指导

(1) 多行提取为一次提取一行数据，将保存数据步骤放在多行提取内(如图 3.3.6.1-1)，则可以实现循环保存每行数据。



(图 3.3.6.1-1 多行提取)

(2) 多行提取的 xpath 原理是将提取规则的一条完整的 xpath 拆分开。

“xpath 提取规则”所填写的是 xpath 中定位到行的部分，添加变量中填写的是剩余 xpath。

如图 3.3.6.1-2 所示，比如提取标题的 xpath 为：

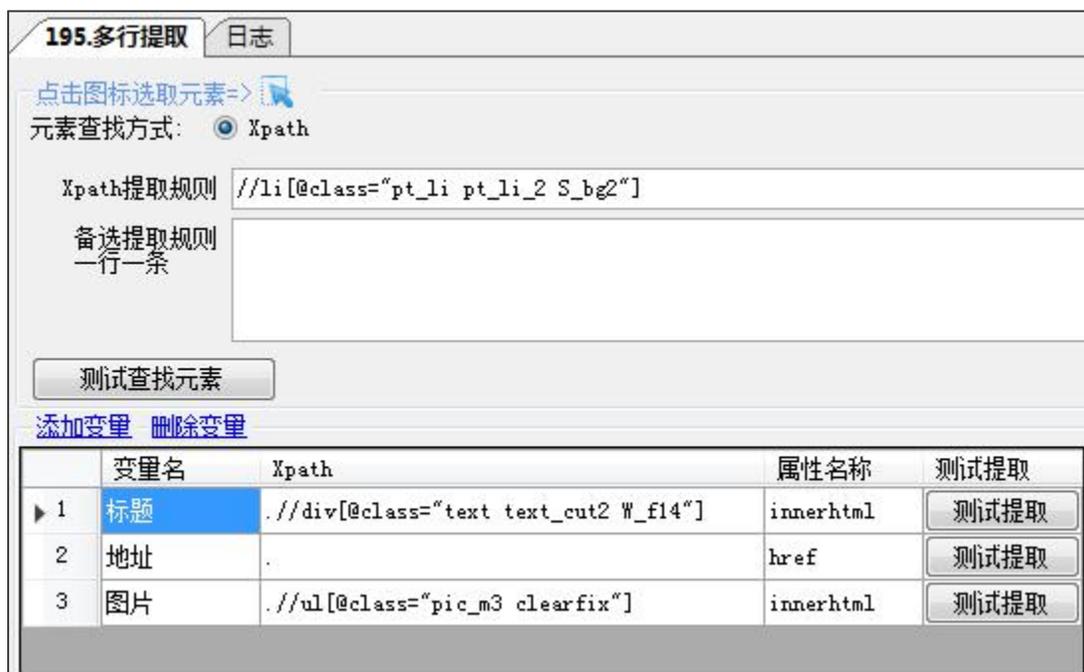
```
//li[@class="pt_li pt_li2 S_bg2"]//div[@class="text text_cut2 W_f14"]
```

则 xpath 提取规则中写：

```
//li[@class="pt_li pt_li2 S_bg2"]
```

添加变量中的 xpath，则需要在剩余 xpath 最前面加一个 “.”

```
./div[@class="text text_cut2 W_f14"]
```



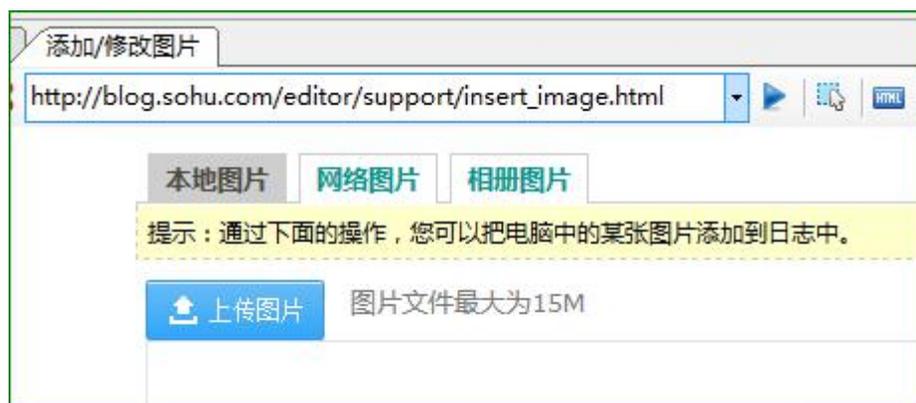
(图 3.3.6.1-2 多行提取)

7、文件上传

需要点击按钮上传文件的情况下，则可以使用此步骤。

7.1 操作指导

(1) Xpath 提取规则：选择上传按钮的 xpath，如图 3.3.7.1 “上传图片”按钮。



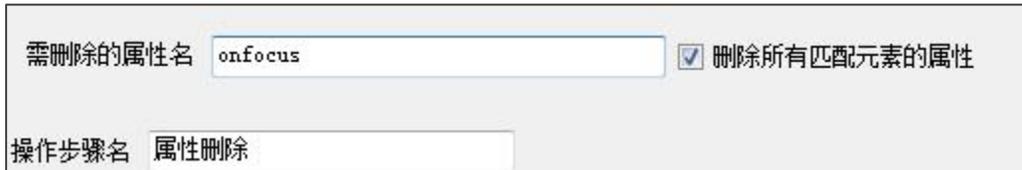
(图 3.3.7.1 点击上传图片)

(2) 选择上传文件的窗口标题：根据在火车浏览器中，点击上述“上传图片”后弹出的选择文件框的名称填写。

(3) 上传文件的路径：选择需上传文件在本地的存储路径，可以直接赋值，也可以调用文本型变量。

8、属性删除

属性删除功能即根据 xpath 定位，删除用户想要删除的属性。比如用户在使用火车浏览器输入验证码时，触发了某个网站 onfocus 属性中的 refresh_code() 事件，导致验证码图片刷新无法输入正确，所以需要把 onfocus 属性删除。



需删除的属性名 删除所有匹配元素的属性

操作步骤名

(图 3.3.8 属性删除)

8.1 操作指导

- (1) Xpath 定位该属性所在的位置。
- (2) 正确填写需要删除的属性名称。
- (3) 确定需要删除的属性是否有多个，如果有多个，并且全部都要删除的情况下，则勾选“删除所有匹配元素的属性”选项。

9、键盘操作

该功能是模拟真实的键盘输入，在仿真速度设定时，数字越大则输入速度就越慢。

9.1 操作指导

- (1) 发送文本：在向输入框中输入数据时，先用点击给输入框一个 focus 点击操作，给输入框一个焦点，然后再使用键盘操作。
- (2) 指定按键：向页面发送指定的按键。
- (3) 使用键盘操作向页面中输入快捷键：如图 3.3.9.1



发送文本 仿真速度 (字/分钟)

指定按键 延时等待 (毫秒):

(图 3.3.9.1 输入快捷键操作)

注意：快捷键的格式为：大括号“{}”中加上快捷键名称。

可以在内容中掺杂快捷键，也可以直接发送快捷键以及组合键，组合键并列写入即可，例如：`{BACKSPACE}{DOWN}`。其中键盘操作中支持的快捷键有：`{(BACKSPACE)|{DOWN}|{END}|{ENTER}|{ESC}|{HOME}|{LEFT}|{PGDN}|{PGUP}|{RIGHT}|{TAB}|{UP}|}`。

除了上面列出的快捷键，其他如 `ctrl`，`shift` 等都是不支持的。

10、鼠标拖拽

该功能是模拟真实的鼠标拖拽操作，例如较为热门的滑动型验证码，就是需要滑动拖拽才能实现的。

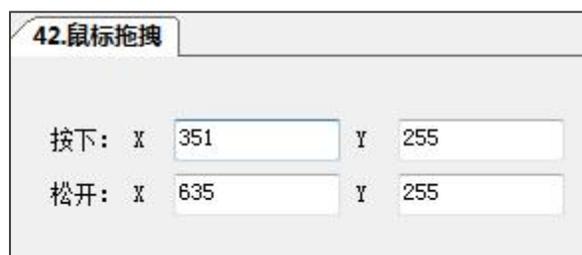
10.1 操作指导



(图 3.3.10.1-1 鼠标拖拽示例)

(1) 改变浏览器窗口大小：随着浏览器窗口大小的改变，坐标也随之改变，这会导致鼠标拖拽的设置失效，所以在拖拽前需要设定好窗口大小。(浏览器窗口具体的大小，可按实际情况设置，最好根据实际打开的页面大小比例进行缩小，并且保存你滑动的元素，在当前页面显示框中可见。)

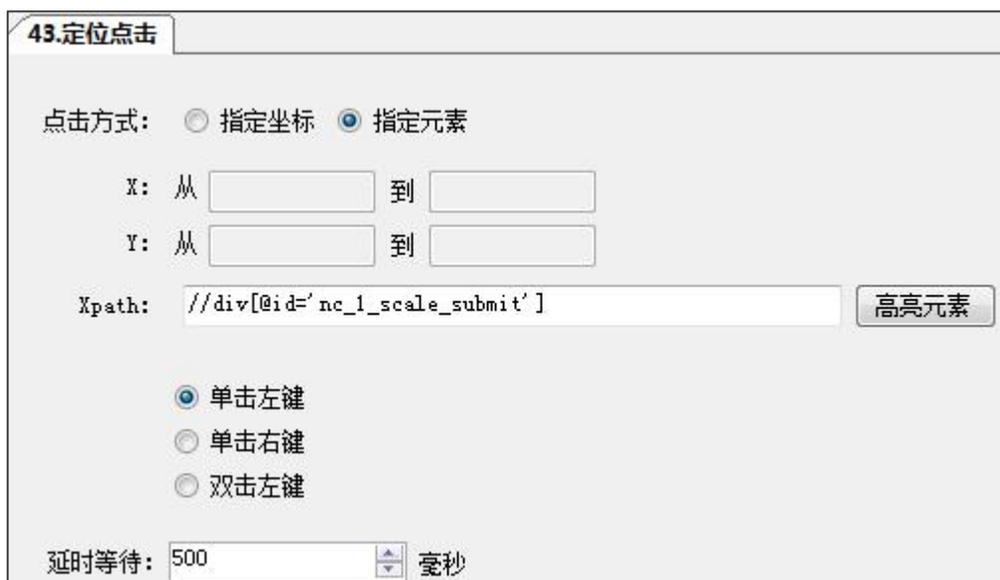
(2) 窗口设置好后，移动鼠标到拖拽目标的按下以及松开处，分别获得坐标，如图 3.3.10.1-2，然后测试运行就可以看到具体的效果。



(图 3.3.10.1-2 按下与松开坐标设定)

11、定位点击

在指定坐标内或指定元素实现点击，点击方式包含单击左键、单击右键，双击左键三种，可以满足操作中不同的点击需求。



(图 3.3.11 定位点击设置)

四、变量处理

火车浏览器中经常需要用到变量，变量处理是实现对变量进行重置、计数、转化等功能的操作。



(图 3.4 变量处理)

1、变量处理

除了对变量列表进行添加、删除、排序之外，对单个变量还可以进行前后截取、正则替换等处理。

2、变量重置

对文本型变量、计数器变量及 list 变量重新赋值。

文本型变量进行赋值的操作，可以将初始值填写在“变量内容”框中直接变量，对 list 变量的赋值，如果一行一个书写（如图 3.4.2），则赋值多行，也可以将赋值内容保存在文本型变量中，然后调用变量。



(图 3.4.2 变量重置)

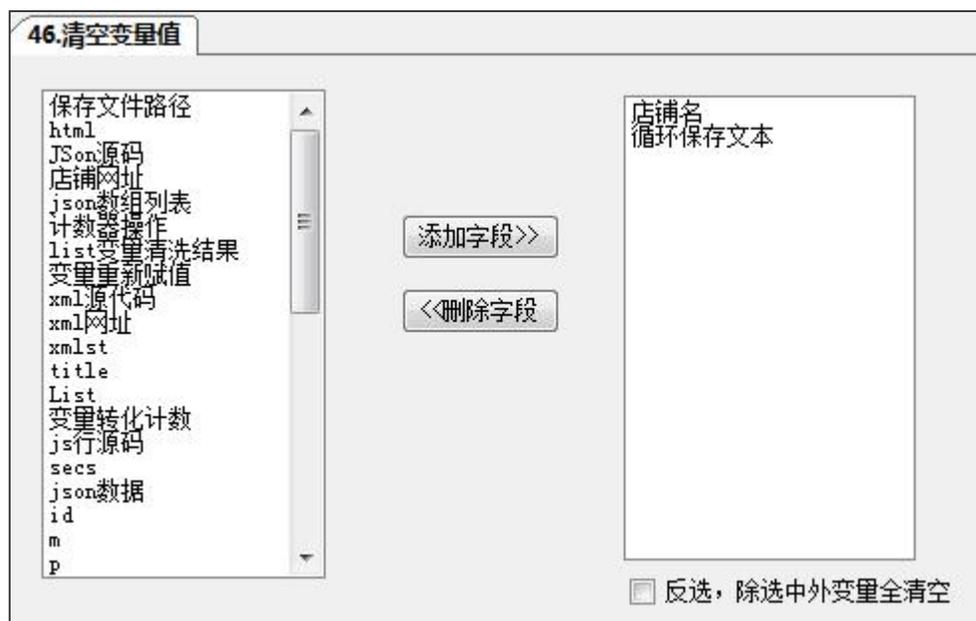
3、清空变量

用于清空变量中的内容。

3.1 操作指导

(1) 如图 3.4.3.1 所示，左边为当前脚本中的所有变量，右边则为需要清除的变量。

(2) 如果勾选了“反选，除选中外变量全清空”这一选项，那么图示的右边是不需要清空的变量，左边为需要清空变量值的变量。



(图 3.4.3.1 清空变量值)

4、变量清理

对 List 变量进行清空、去重等操作，操作完成后可将变量保存至新变量或直接在当前进行修改。

操作对象 list变量 保存至新变量:

操作方式 清空整个变量 去掉为空的记录 移除最后一条记录

移除重复记录 移除第一条记录

内容过滤删除

内容必须包含 | 为只包含多个中的一个即可, ↑ 为多个条件都必须包含。两个标记不能同时使用

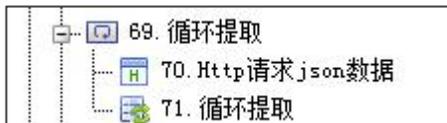
内容不得包含 | 为只包含多个中的一个即可, ↑ 为多个条件都必须包含。两个标记不能同时使用

排序 升序排序 (asc) 降序排序 (desc)

(图 3.4.4 变量清理)

5、循环提取

从文本变量中循环提取相同格式的数据保存至 list 变量，便于操作调用，如图 3.4.5。循环提取的方式包含前后截取和正则提取两种。



(图 3.4.5 循环提取流程)

6、计数处理

对计数器变量进行重置以及进行加减乘除等操作。

45.计数器操作

操作类型 增加 减少 乘以 除以(进一法) 除以(四舍五入) 重置

操作变量 计数器操作

操作数值 使用变量 计数器操作

(图 3.4.6 计数器操作)

6.1 操作指导

(1) 操作数值：是对计数器变量进行加减等操作时的数值。

(2) 使用变量：操作数值，可以使用变量。选择使用变量时，操作数值中设置的数字就不再起效。

注意：计数器变量只能保存正整数，计数器操作中也只能操作正整数相关的操作。对计数器变量进行简单的计算可以从类型中选择，但如果计算比较复杂，需要使用 "自定义代码 C#或者是 js"。

7、变量转化

将文本变量、计数器变量、list 变量之间进行转化。

7.1 操作指导

(1) 文本变量转化为计数器变量：文本变量中是正整数时，才能成功转化为计数器变量，否则失败返回 0。

(2) List 变量合并成文本变量：合并字符串是自定义 list 值的间隔符。

47. 变量转化

文本变量转化为计数器变量 文本变量添加到List变量 List变量合并成文本变量

List变量个数转化为计数器变量

原变量名 合并字符串 [换行]

新变量名

字符转化为数字失败后返回0

(图 3.4.7.1 变量转化)

8、提取列表

将文本型变量中的内容按规则处理成 list 变量并保存。

8.1 操作指导

(1) 字符串分割：将按一定方式间隔的字符串，提取成列表变量。

例字符串：1, 2, 3, 4, 5，该示例使用逗号间隔，因此可将逗号作为分隔字符串如图 3.4.8.1-1。

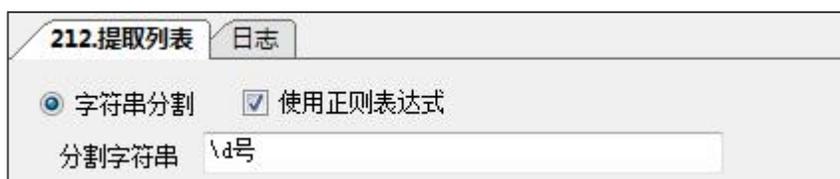
212. 提取列表 日志

字符串分割 使用正则表达式

分割字符串

(图 3.4.8.1-1 字符串分割)

例字符串：苹果 1 号梨子 2 号香蕉 3 号猕猴桃 4 号，该示例以“数字+号”的方式间隔，且数字是会变化的，因此可以使用正则表达式如图 3.4.8.1-2。



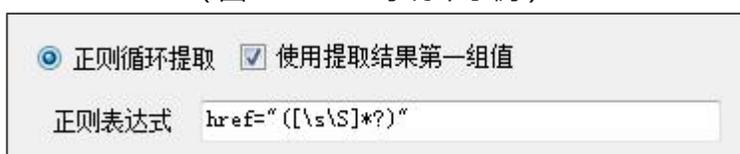
(图 3.4.8.1-2 正则表达式)

(2) 正则循环提取：提取符合正则表达式的字符串。

例字符串：如图 3.4.8.1-3

```
<a href=" http://www.locoy.com/" title=" 火车头采集器 "target="_blank" >火车头采集器&nbsp;</a>
<a href=" http://www.tuwenba.com/" title=" 图文吧 "target="_blank" >图文吧&nbsp;</a>
<a href=" http://www.qiduowei.com/" title=" 企多维 "target="_blank" >企多维&nbsp;</a>
<a href=" http://www.shuju.net/" title=" 大数据观察 "target="_blank" >大数据观察&nbsp;</a>
```

(图 3.4.8.1-3 字符串示例)



(图 3.4.8.1-4 正则循环提取)

按图 3.4.8.1-3 中的字符串，填写好正则表达式如图 3.4.8.1-4 即可正则循环提取。注意：如勾选“使用提取结果第一组值”，也就是提取匹配组 1 的内容。

(3) 楼层类型分割：使用前后截取的方式提取内容，前后的字符也可以使用正则表达式，参照上述正则循环提取的示例，楼层类型分割可以如图 3.4.8.1-5 写：



(图 3.4.8.1-5 楼层类型分割)

9、时间处理

进行时间转化，得到系统时间等操作。



(图 3.4.9 时间处理)

9.1 操作指导

- (1) 得到系统时间：确定时间格式后，测试可得符合格式的系统当前时间。
- (2) 将指定时间转换为时间戳：给出指定时间，并勾选时间戳转换选项。
- (3) 将指定格式数据进行时间的增加、减少等操作参考图 3.4.9.1。

198.时间处理 日志

指定时间: 2012-10-10 12:12:12 转换指定时间, 默认当前时间

时间格式: yyyy-MM-dd HH:mm:ss 选中时间戳时可为空

时间差转换: +1|0|0|0|0|0 为空则不进行转换

时间差转换参考格式
年|月|日|时|分|秒
当前年份增加1年: +1|0|0|0|0|0 当前月份增加1月: 0|+1|0|0|0|0

时间戳转换: 转换为时间戳 保存至变量: 时间

测试结果: 2013-10-10 12:12:12 测试

操作步骤名 时间处理

(图 3.4.9.1 时间差转换)

注意：时间戳转换为 10 位，若要 13 位，需用 JS 实现时间戳转换。

10、随机取值

功能是从 list 变量中随机提取一行值并赋值给文本型变量。

11.随机取值 日志

处理的变量: list

按索引取值: 第一
 最后
 随机
 指定 0

取值使用后删除

保存至变量: 文本变量

操作步骤名 随机取值

(图 3.4.10 随机取值)

10.1 操作指导

(1) “取值使用后删除”：勾选即表示已经取出来的值，将从 list 变量中删除，下次不会再重复取值。

(2) 随机取值删除，指的是删除缓存中的值，如果 list 变量中的值是从文本中读入的，则不会删除文本中的内容。

(3) 注意：如果“循环操作”中的“循环遍历 list”已经调用了该 list 变量，那么此步骤中就不能勾选“随机取值删除”，否则运行中会报错。

11、提取 Xml

即提取 Xml 列表中的内容。（这里需对 Xml、根元素、父元素、子元素有所了解。）

11.1 操作指导

(1) xpath 类似于 xpath，以 “/” 连接从根元素开始扩展到底部的不同级元素。而且 xpath 支持 “//”，即可以写相对路径。

比如提取 id 的值，其绝对路径显示为/feed/entry/id，如果 entry 在 feed 元素中是唯一的，那么就可以写成//entry/id，这样也能提取到 id 元素的值。



(图 3.4.11.1 提取 xml 变量)

(2) 属性名：属性名称代表用户要取该元素中的什么属性，如果为空默认取 InnerText，其余属性可以自定义填写。

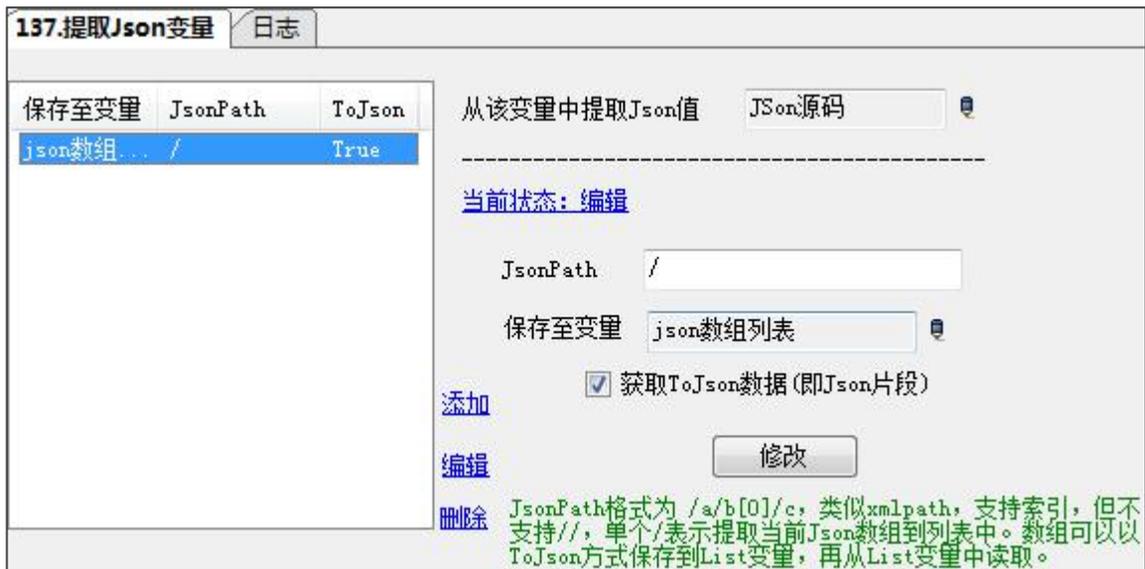
(3) 保存至变量：可以选择文本型变量，也可以选择 list 变量。

12、提取 Json

提取 json 变量中的内容。（这里需对 json 语法、名称/值对、对象、数组有所了解。）

12.1 操作指导

(1) 明确 json 含义和格式 (数据由逗号分隔, 花括号保存对象, 方括号保存数组) 后, 需要确认数据是否符合 json 格式。



(图 3.4.12.1 提取 Json 变量)

(2) 如果符合, 就提取数组中的内容保存至 list 变量, 数组就是[]中的内容, 这里 JsonPath 就是"/", 并需要勾选“获取 Tojson 数据”。

(3) 现在数组中的对象就被保存至 list 变量, 获得对象中值对的值, 则循环遍历 list, 然后再进行一次提取 json 变量处理, 这样循环遍历对象, 就能获得不同对象中名称的值对, 用户可以选择不同的保存方式将值对保存下来。

12.2 注意事项

(1) 只有数组中的内容才可以通过本步骤的 ToJson 方式保存至 list 变量, 值对以及对象只能保存在文本型变量中。

(2) 这里的 Jsonpath 与 xpath 类似, 支持数组进行索引。但不同的是, Jsonpath 不支持//, 即必须按级向下写。

(3) 保存数组内容至 list 变量需要勾选“获取 Tojson 数据”, 保存值对内容则不需要。

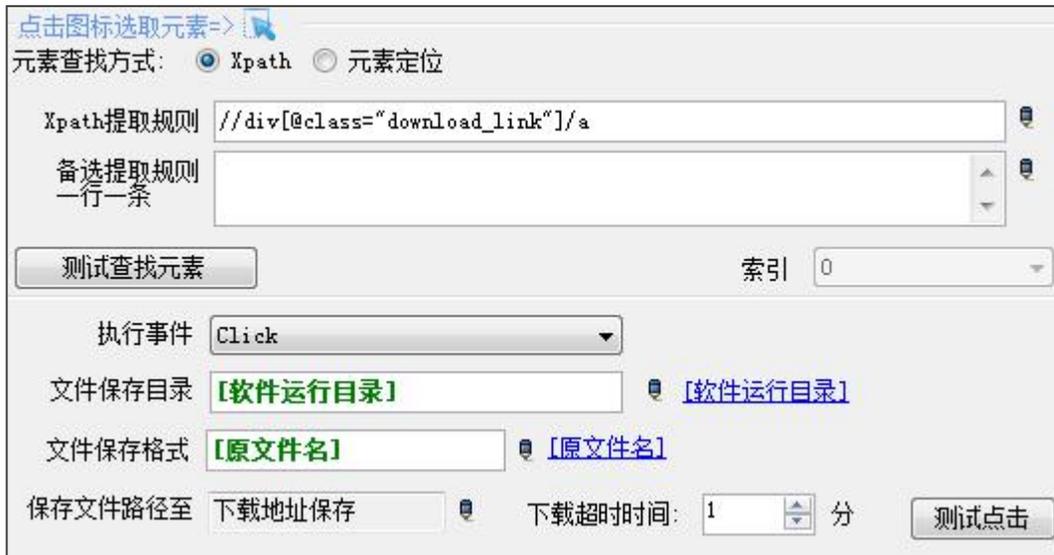
五、文件下载

文件下载是使用火车浏览器来下载用户所需要的文件, 其中包括有真实地址的文件下载和找不到真实下载地址只能点击后下载的点击下载。

1、点击下载

有时用户无法找到真实的下载地址, 需要点击下载按钮才能下载文件。这就用到了第四章第三部分的元素操作中的“点击”来完成下载。

1.1 操作指导



(图 3.5.1.1 点击下载)

- (1) xpath 提取规则、备选提取规则、测试查找元素、索引、执行事件与点击中的定义均可参考元素操作——点击。
- (2) 文件保存目录：下载文件保存的目录。
- (3) 文件保存格式：下载文件命名方式。
- (4) 保存文件路径至：将文件下载后保存在本地的路径保存至变量。
- (5) 下载超时时间：在超时时间内文件还没有下载完成，就强制停止下载，进行下面的步骤。

2、文件下载

将文件的真实下载地址写入步骤实现文件下载。

2.1 操作指导



(图 3.5.2.1-1 填写下载地址)

- (1) 如图 3.5.2.1-1 将下载地址写入步骤，可保存在文本型变量中，然后再调用该变量。



(图 3.5.2.1-2 文件下载保存)

(2) 填写保存目录、格式。下载地址保存至变量是指将文件下载后保存在本地的路径保存至变量。

2.2 注意事项

(1) 文件下载步骤地址栏中，写的是下载的地址，而非网页地址。

(2) 如果有多个下载地址，请保存至 list 变量中，然后循环遍历 list 保存至文本型变量，再调用该文本型变量。

(3) 其余 http 请求的设置，按照 get 设置即可。如果地址有特殊的编码要求，再根据实际情况修改。

六、数据库操作

1、执行 sql 语句

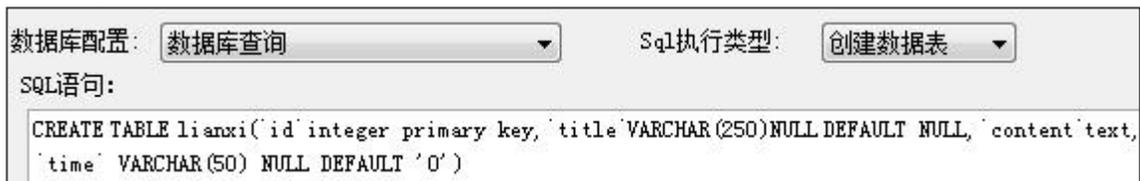
即执行 sql 语句的增删、修改和查询。

1.1 操作指导

(1) 配置数据库配置，请参照本文第三章第三部分数据库配置的用法。

(2) sql 语句的示例以及注意事项。

① 创建表语句：



(图 3.6.1.1-1 创建表语句)

② 查询语句：



(图 3.6.1.1-2 查询语句)

注意：选择“查询结果保存至变量”，一次只可以保存一行数据，可以保存至文本型和 list 变量。如果不勾选“查询结果保存至变量”，则将列值自动赋值给与列同名的文本型变量。

③ 更新语句



The screenshot shows a configuration window for an SQL update statement. At the top left, there is a label '数据库配置:' followed by a dropdown menu set to '数据库查询'. To the right, there is a label 'Sql执行类型:' followed by a dropdown menu set to '更新数据'. Below these is a text area labeled 'SQL语句:' containing the SQL code: `update lianxi set title='新标题' where title='标题'`.

(图 3.6.1.1-3 更新语句)

④ 删除语句：



The screenshot shows a configuration window for an SQL delete statement. At the top left, there is a label '数据库配置:' followed by a dropdown menu set to '数据库查询'. To the right, there is a label 'Sql执行类型:' followed by a dropdown menu set to '删除数据'. Below these is a text area labeled 'SQL语句:' containing the SQL code: `delete from lianxi`.

(图 3.6.1.1-4 删除语句)

七、文件操作

保存文本、目录、导出 excel 等文件相关操作。



(图 3.7 文件操作)

1、保存文本

保存文本、list、计数器变量中的内容。

1.1 操作指导

(1) 变量类型以及内容格式模板。

①文本类型：保存文本类型以及计数器类型，格式可以自定义。



(图 3.7.1.1-1 文本类型内容格式)

②List 类型：直接保存 List 中的内容。List 合并字符串可使用换行，一行一个，也可以自定义连接字符串。



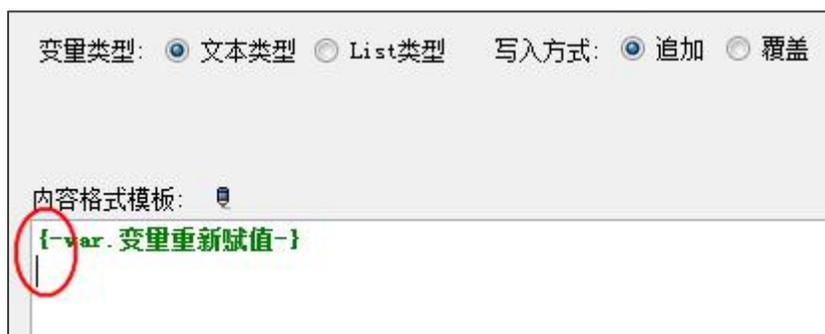
(图 3.7.1.1-2 list 类型内容格式)

(2) 写入方式：追加是指新内容保存在原有内容后面。覆盖是指新内容将文本中原有内容覆盖掉，即不保存原有内容。

(3) 保存路径:保存文本的路径以及名称。

(4) 文本编码：文本的编码格式。

(5) 文本类型变量如需要一行一个保存数据, 则在模板的末尾加一个 ENTER (换行) 即可。



(图 3.7.1.1-3 文本变量换行保存数据)

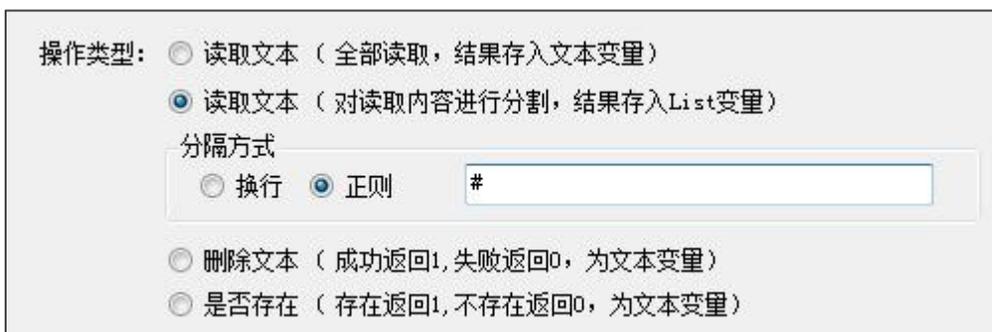
2、文件操作

读取文本中的内容, 删除文本, 判断文本是否存在等。

2.1 操作指导

(1) 读取文本保存至文本型变量: 只能选择非只读的文本型变量保存文本内容。

(2) 读取文本保存至 list 变量: 按照间隔方式保存至 list 变量, 可选换行和正则间隔。换行: 文本中内容一行一个。正则: 文本中内容按一定规律间隔, 比如:



(图 3.7.2.1 正则分隔读取文本)

(3) 删除文本: 将已经存在的文本删除, 删除成功则返回 1 并保存至非只读的文本型变量, 失败则返回 0。

(4) 是否存在: 判断文本是否存在, 存在则返回 1 保存至非只读的文本型变量, 不存在则返回 0。

3、目录操作

创建目录, 读取目录中内容, 删除目录。

3.1 操作指导

(1) 创建目录: 创建新的目录。

(2) 读取目录: 读取目录中内容。



(图 3.7.3.1 读取目录操作)

- ①目录路径：读取哪一个目录下的内容。
- ②保存至变量：读取目录中的文件路径保存至 list 变量。
- ③指定文件后缀：读取该目录下哪一种后缀的文件，形似 *.txt，如果为空不写，则默认读取所有类型的文件路径。
- ④路径保存方式：保存至 list 变量中路径的形式。注意，不能读取目录中嵌套目录的路径信息。

(3) 删除目录：删除已存在目录及其中的文件。

4、导出 Excel

将数据库中的数据导出为 excel 文件。

4.1 操作指导



(图 3.7.4.1 导出 excel 操作)

导出 excel 需要先配置数据库文件路径，可以使用 sqlite, mysql, sqlserver 等数据库，如果只是简单的数据导出，可以使用 sqlite 内存数据库，因为是虚拟数据库不会产生实体的数据库文件。

(1) 数据库中需要导出的列，在 select 语句中写上该列名。注意：如果要导出的是数据库中所有的列，不能单纯使用 “*” 代替，需要将列名写出。

(2) 文件保存地址中，需要加上路径以及保存文件的名称。

(3) 文件格式支持 2003 以及 2007。

5、Ini 配置文件

ini 格式的配置文件，让用户可以通过操作文本的方式设置、保存或读取脚本中变量的值。

5.1 操作指导

(1) 节点名称：定义 ini 文件中的节点，可以保存相同变量的不同值到不同的节点中，如果有两个节点比如[config]、[user]，读取时选择[user]节点，则读取的是[user]节点变量中的值，保存配置同理。注意：文本型变量的值与 list 变量的值格式不同，变量按行间隔，文本型变量的值一行一个，List 变量中各行的值使用“，” 隔开。

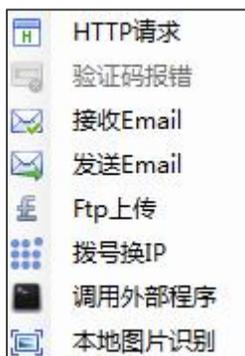
(2) 文件路径：可以访问的绝对路径，或是使用火车浏览器的系统变量[软件运行目录]。配置文件一定是 ini 格式文件。



(图 3.7.5.1 读取配置操作)

八、其他操作

包括 http 请求、邮件收发、调用程序等其他操作。



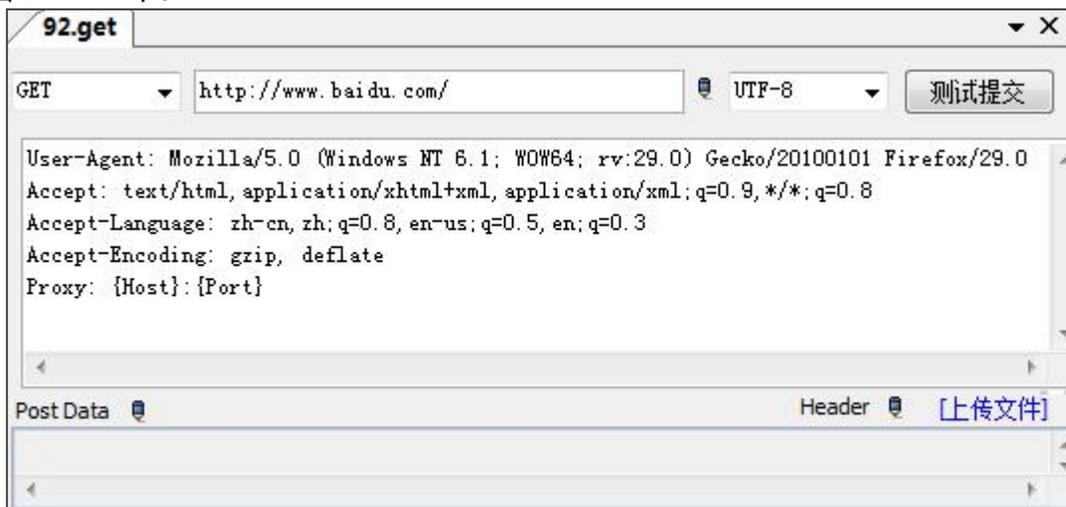
(图 3.8 其他操作)

1、http 请求

http 请求网址得到源代码，可以进行 get、post 请求。

1.1 操作指导

(1) 模拟 get 请求，可以使用 fiddler 抓包获取头信息，然后将信息填在如图 3.8.1.1 中。



(图 3.8.1.1 模拟 get 请求操作)

(2) 模拟 post 请求，方法参考 get 请求，post 请求需要填写 Post Data。

(3) 若是 http 需要设置代理，只需在头信息中加上有效代理即可。如 Proxy: {Host}:{Port}，形如 Proxy: 127.0.0.1:8888。

2、验证码报错

在使用自动打码功能时，打码平台对验证码图片可能存在识别错误，但是账户中的积分还是会扣除，针对这种情况，用户可以做一个条件判断的步骤，运行验证码报错功能后会将结果以及验证码图片发送给打码平台，当打码平台审核确认后，会将积分返还账户。

注意：只可在打码错误的情况下使用此功能，也就是在使用此功能前要加上条件判断的步骤。否则频繁提交会导致打码平台判断为恶意提交，并禁止提交报错。

3、接收 Email

接收邮箱中符合条件的邮件。

3.1 操作指导

(1) 登录协议需要选择 POP。

(2) 发邮件的邮箱必须要开通 POP/IMAP。

(3) 邮箱密码处填写的是邮箱的授权码(授权码是用于登录第三方邮件客户端的专用密码)。

The screenshot shows a configuration window for mailbox settings. It is divided into two main sections: 'Mailbox Information Settings' and 'Mail Retrieval Settings'. In the first section, the login protocol is set to POP3. Fields for mailbox account, password, and server address (pop.163.com) are present, along with a port field set to 110 and an unchecked SSL checkbox. A 'Test Login' button is also visible. The second section allows setting the number of mail items to retrieve (set to 1), the content to save (set to 'Mailbox Content'), the sender's mailbox, and a filter for mail titles.

(图 3.8.3.1-1 接收邮件设置)

(4) 服务地址、端口以及是否勾选 ssl，不同的邮箱有不同的选择，接收邮件服务器是 pop，如图 4.8.3.1 以 163 邮箱为例：

类型	服务器名称	服务器地址	SSL协议端口号	非SSL协议端口号
收件服务器	POP	pop.163.com	995	110
收件服务器	IMAP	imap.163.com	993	143
发件服务器	SMTP	smtp.163.com	465/994	25

(图 3.8.3.1-2 服务地址、端口等设置)

(5) 发件人邮箱以及邮箱包含的字符可以不全部填写，仅填写其中一项也可以正常保存。

4、发送 Email

即向邮箱发送邮件。

4.1 操作指导

- (1) 登录协议选择 IMAP。
- (2) 发邮件的邮箱必须要开通 POP/IMAP。
- (3) 邮箱密码处填写的是邮箱的授权码(授权码是用于登录第三方邮件客户端的专用密码)。

(图 3.8.4.1 发送邮件设置)

(4) 服务地址、端口以及是否勾选 ssl，不同的邮箱有不同的选择。发送邮件服务器一般是 smtp，如上文接收 email 内容中的图 3.8.3.1-2。

5、ftp 上传文件

即向 ftp 中上传文件，执行删除文件、列出文件等命令。

6、拨号换 IP

为防止 IP 被禁，用户可以使用拨号换 IP 的功能，如图 3.8.6 所示选择拨号类型并填写账户信息即可。

(图 3.8.6 拨号换 IP)

7、调用外部程序

调用外部可执行程序来处理内容，可以将火车浏览器的参数传递给外部程序，也可以获得外部程序处理后返回的值。

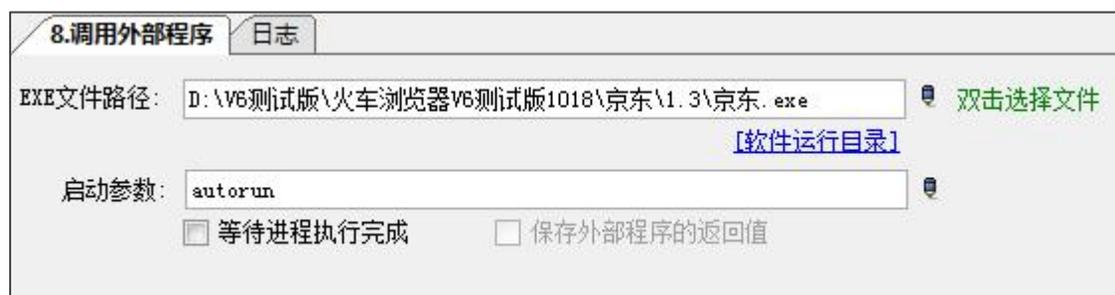
7.1 操作指导



(图 3.8.7.1-1 调用外部程序)

(1) 文件路径：填写完整的可执行文件的路径（如果是放在火车浏览器安装目录下，也可以用[软件运行目录]来代替软件安装目录的路径）。

(2) 启动参数：启动可执行文件运行的参数，也可以向可执行文件中传递参数，如图 3.8.7.1-2。



(图 3.8.7.1-2 调用 exe 直接启动)

(3) 等待进程执行完成：勾选此选项，则必须等待可执行文件运行完成，才能运行脚本中的下一个步骤。

(4) 保存外部程序的返回值：将外部程序的返回值保存至火车浏览器中的变量中，比如在火车浏览器生成的 exe 中保存的是结束流程中返回的值。若是其他程序就是 return 返回的值。

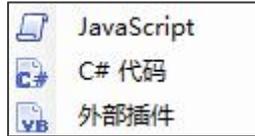
(5) 强制关闭外部程序：外部程序最长运行时间，由超时时间定义。超过了超时时间，如勾选该项，则强制关闭外部程序。

8、本地图片识别

将本地图片、或由 http 文件下载和网页截图所得的图片进行识别。选择识别配置后，识别结果可保存至变量。

九、自定义代码

火车浏览器可自定义 js、c#、外部插件代码来操作脚本变量。



(图 3.9 自定义代码)

1、JavaScript

javascript 步骤, 可以操作火车浏览器变量, 可以操作当前页面中的 DOM 元素, 也可以添加 js 文件。

1.1 操作指导

操作火车浏览器中的变量, 可以对变量进行加减乘除等操作, 最后需要返回的值, 需要加 return, 然后才能保存至变量, 否则返回的是 undefind 字符。

简单代码示例

①对变量进行加法: var a={var.变量 a}; return a;

②JS 实现时间戳的转换: 如图 3.9.1.1。

```
function getLocalTime(nS) {  
    return new Date(parseInt(nS) * 1000).toLocaleString().replace(/年|月/g, "-").replace(/日/g, " ");  
}  
return getLocalTime({-var.时间戳-});
```

(图 3.9.1.1 JS 实现时间戳转换代码示例)

2、C#代码

使用 C#来操作脚本中的变量。

2.1 操作指导

(1) 在新建步骤的时候会自动在代码片段中添加默认的方法, 这个方法名不能修改, 默认调用的方法如图 3.9.2.1-1 所示。

```
C#代码: 复制默认代码到剪切板 查看可用的程序集  
  
public static void Execute(Dictionary<string, string> stringDic, Dictionary<string, List<string>>  
listDic, Dictionary<string, int> intDic)  
{  
  
}
```

(图 3.9.2.1-1 默认调用方法代码)

(2) 在这个方法中可以对变量的值进行操作, key 是变量名, value 是变量的值。注意: 默认引用的程序集是无法自定义添加的。

(3) 对文本型变量, List 变量, 计数器变量的示例如下。

①修改变量 Test 的值为 III, 如图 3.9.2.1-2。

C#代码: [复制默认代码到剪切板](#) [查看可用的程序集](#)

```
public static void Execute(Dictionary<string, string> stringDic, Dictionary<string, List<string>> listDic, Dictionary<string, int> intDic)
{
    stringDic["Test"] = "111";
}
```

(图 3.9.2.1-2 修改变量 text 的值)

②判断文本型变量[c]的值为 a 时清空 List 变量[a]的值，如图 3.9.2.1-3。

C#代码: [复制默认代码到剪切板](#) [查看可用的程序集](#)

```
public static void Execute( Dictionary<string, string> stringDic, Dictionary<string, List<string>> listDic, Dictionary<string, int> intDic)
{
    if (stringDic.ContainsKey("c") == false) return;
    if (stringDic["c"] == "a" && listDic.ContainsKey("a")) listDic["a"] = new List<string>();
}
```

(图 3.9.2.1-3 判断文本型变量值)

3、外部插件

支持 C#、php、python 语言编写的三种插件类型，火车浏览器中自带示例，可以参照示例学习插件的编写方法。

3.1 操作指导

插件类型 C# PHP Python

选择插件 中文分词 [插件设置](#)

注意，PHP和Python插件只能处理变量，C#插件可以操作浏览器对象

插件参数

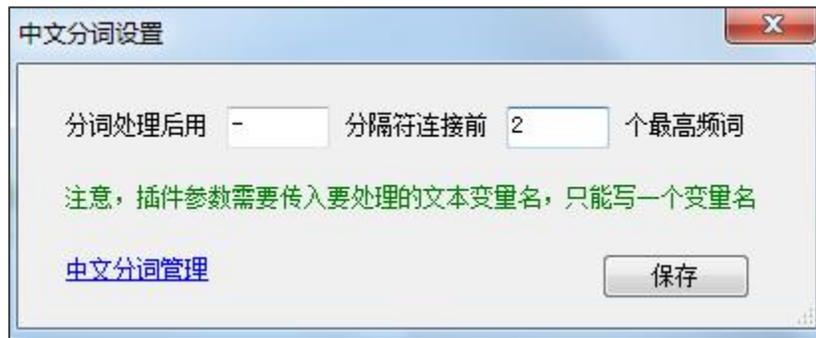
软件会传所有变量给插件，插件参数是额外的字符串变量，用来告诉插件它所处的位置或状态

[点击这里，下载更多插件](#) [点击这里复制并打开示例插件源代码](#)

(图 3.9.3.1-1 外部插件)

(1) 插件类型可选 C#、php、python。

(2) 插件设置：设置不同插件的参数。以中文分词为例，可设置分词的间隔方式，以及个数。



(图 3.9.3.1-2 中文分词插件设置)

(3) 插件参数处写的是需要处理的变量名称, 最后处理的结果也会返回给该变量。

十、互动设置

1、气泡提示

对运行过程中, 弹出的提示框进行标题和内容的设置。

2、输入设置

运行时启用该功能可以弹出窗口让用户输入内容, 以达到交互的目的。

2.1 操作指导

(1) 超时关闭窗口: 如图 3.10.2.1 所示, 超过设定时间就自动关闭窗口, 为 0 秒时, 则不自动关闭窗口。注意: 若输入的名称中有不允许为空的值, 则超时设置无效, 即不得为空的变量没有填值时, 无法自动关闭窗口。

(2) 输入设置的名称有很多种类型, 以图 3.10.2.1 为例:

- ①Label: 文字提示。
- ②Boolean: 布尔值, 返回 true, 或者 false。
- ③Number: 输入框中只可以填写数字。
- ④Text: 填写任意字符串。
- ⑤select: 选择按钮, 以列表形式显示。
- ⑥FileName: 可以保存导入文件的路径。
- ⑦DropDown: 选择按钮, 以下拉框的形式显示。



(图 3.10.2.1 输入设置操作)

(3) 默认值处是可以使用变量的，如果默认值选择某一变量，则正式运行时会显示此变量中的值。

3、暂停执行

正式运行时可以截止该项来暂停步骤的运行，这样用户可以在运行窗口中直接进行操作，便于交互。待其他操作完成后点击“继续执行”，就可以接着运行下面的步骤。

注意：无法单步测试运行，只在正式运行时起效。

4、自定义日志

正式运行过程中，可在日志里加上自己想要输出的日志。

5、播放声音

可播放 wav 格式的声音文件。

6、变量输出

输出变量中的内容。在正式运行的时候会弹出变量输出的值，如不点击确定则不会继续后续的步骤。（此功能可用于调试程序。）

第四章、项目管理器

脚本编辑器写好脚本之后，需要在项目管理器中建立任务来长时间运行脚本。



(图 4 项目管理器菜单)

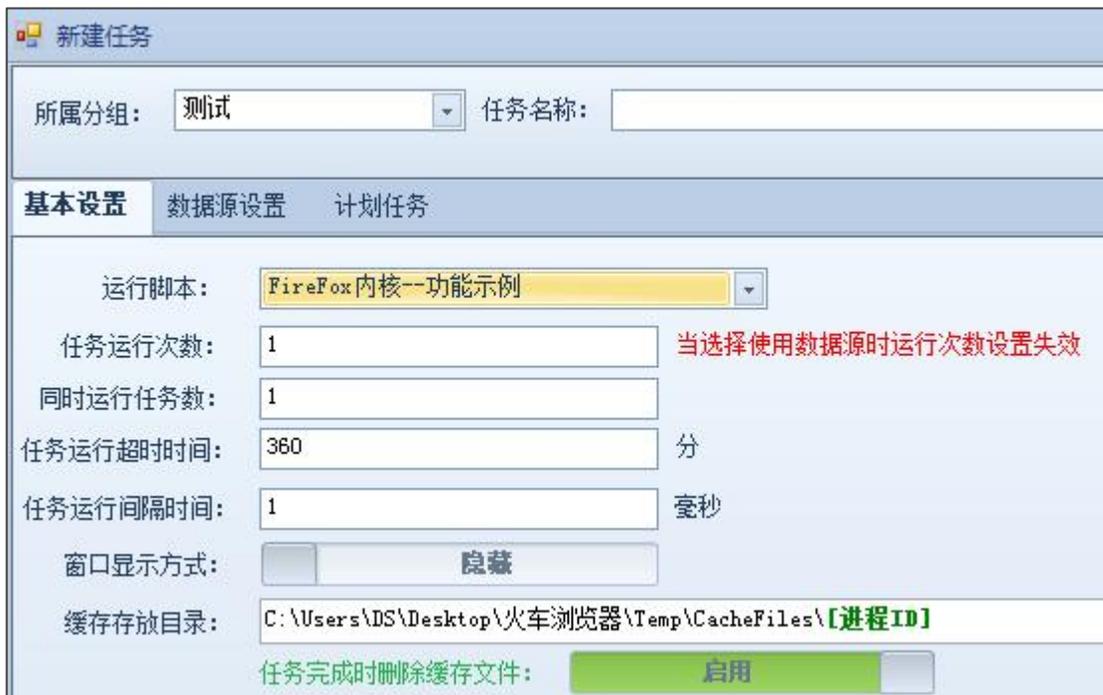
一、任务控制

选中任务可对其进行开始和停止控制。

二、新建分组

即新建一个任务分组。

三、新建任务



(图 4.3 新建任务窗口)

1、基本设置

1.1 操作指导

(1) 运行脚本：选择脚本编辑器中建立的脚本来运行。如果编辑改变脚本内容，项目管理器中无需重新选择运行脚本，可自动调用。但修改脚本内容对项目管理器中当前正在运行的任务不能实时生效，需要停止任务再重新运行。

(2) 任务运行次数和同时运行任务数：如设置运行次数为 6，运行任务数为 4，那么启动运行时，会一次性启动 4 个进程同时运行，等到 4 个进程结束后，再同时启动剩下的两个进程运行。

(3) 任务运行超时时间：当前任务最长运行时间，超过超时时间，则会自动停止。

(4) 任务运行间隔时间：如果任务运行次数有多次，则每一次运行启动的间隔时间。

(5) 窗口显示方式：运行任务时，运行窗口是否显示。

(6) 缓存目录存放方式：运行的缓存目录位置，启用任务缓存删除，则任务完成时会删除缓存文件。

2、数据源设置

可以从数据源中读取存储的数据并传递至脚本变量使用。支持从文件中读取，从 excel 和数据库中读取。

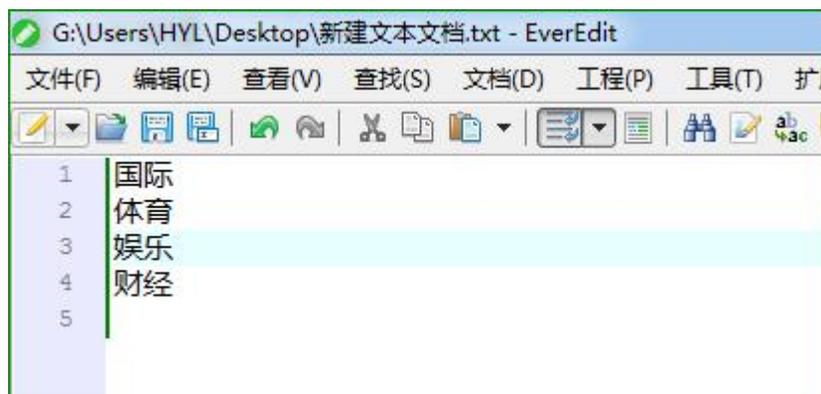


(图 4.3.2 数据源设置)

2.1 操作指导

(1) 从文件中读取：项目管理器按行导入 txt 文本中的内容，赋值给脚本中指定的变量，由项目管理器控制循环运行文本中每行内容。选择从项目管理器中导入文本中内容，可以有效的减少内存的占用。因为每运行完成文本中的一行内容，就会结束当前进程释放占用的内存，然后再启动一个新的进程运行文本中的下一行内容。

①文本地址：导入的文本中内容需一行一个，这样才能按行赋值给选定变量。如图 4.3.2.1-1.

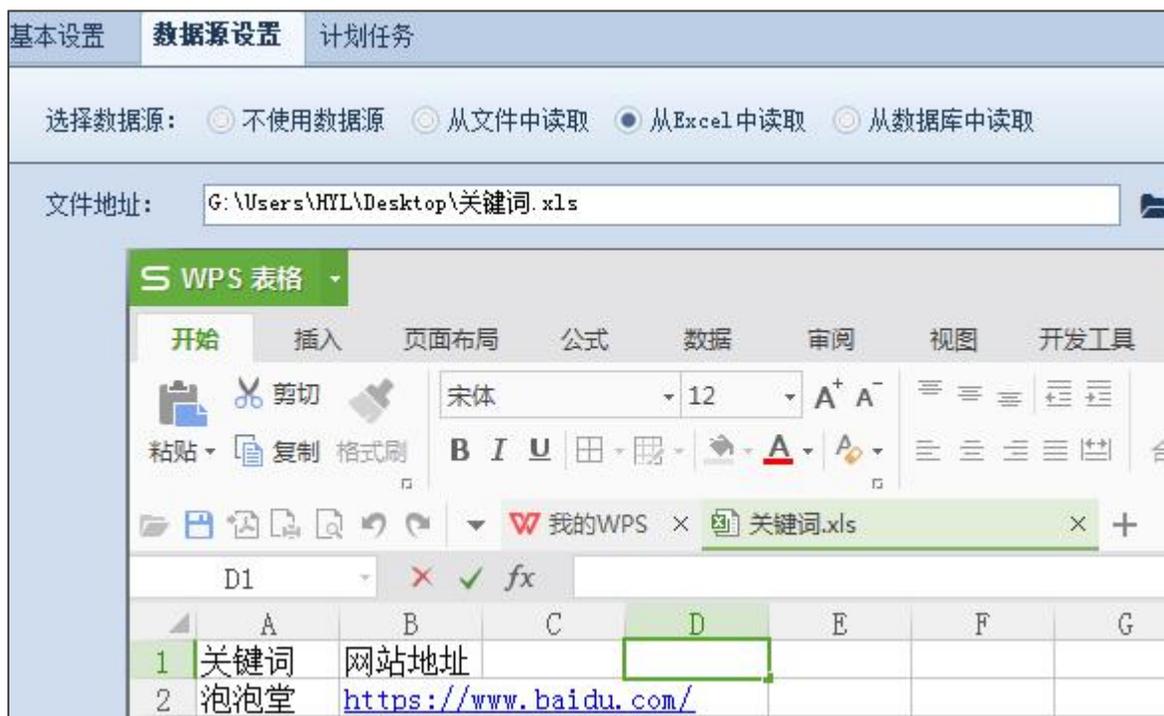


(图 4.3.2.1-1 文本地址导入的内容格式)

②文本内容赋值给变量：文本中按行读取的内容只能赋值给脚本中勾选了只读的文本型变量。

(2) 从 Excel 中读取：项目管理器按行导入 excel 中每行的内容，赋值给脚本中与列同名的变量。

①文件地址：excel 地址。导入 excel 中内容，excel 的格式需如图 4.3.2.1-2，第一行写列名，项目管理器会按行读取列中内容，赋值给与列同名的变量。“关键词”列中内容会赋值给脚本中的文本型变量“关键词”，所以脚本中先建立一个文本型变量“关键词”，这样才能接收到 excel 中传递过来的值。“网站地址”列同理。



(图 4.3.2.1-2 文件地址导入 excel 格式)

②若运行任务时，报下图错误：则需关闭导入的 excel 文件。若是 excel 文件打开，则无法正常读取其中内容。



(图 4.3.2.1-3 报错示例)

(3) 从数据库中读取：读取数据库中列的内容赋值给一系列同名的变量。

①查询语句：写 select 语句，从数据库中查询数据。

②更新语句：写 update 语句，更新数据库中数据。语句中可以调用脚本中的变量，如图 4.3.2.1-4 此语句是脚本运行完之后执行的。

数据库类型: Sqlite Mysql Mssql

数据库连接字符串: server=127.0.0.1;Port=3306; user id=root; password=;Database=table; charset=utf8 [\[软件运行目录\]](#)

查询语句: `select id,网址 from 网址库`

更新语句: `update 网址库 set 已采=1 where id={-var.id-}`

(图 4.3.2.1-4 从数据库读取)

3、计划任务

根据任务需要确定计划任务，勾选启用生效。可选每间隔、每天、每周等，也可使用 cron 表达式来设置。

基本设置 数据源设置 **计划任务**

启用计划任务

每间隔
 每天
 每周
 仅一次
 Cron表达式

每间隔

间隔时间 小时 分钟 秒

限定在以下 ~ 时间段内

(图 4.3.3 计划任务)

四、选项

对任务完成后的提示、时间冲突、软件显示、使用偏好等方面的设置。

第五章、生成 exe

生成 exe 功能是将脚本打包成一个可执行程序，这样就可以直接运行脚本。如需多台电脑都运行脚本，可以将 exe 分发至多台电脑。还可以选择生成绑定硬件以及账号验证的 exe，即可对他人使用进行一些限制。



(图 5 生成 exe 界面)

一、操作指导

1、基本设置

1.1 软件名称

生成的软件名称。

1.2 软件版本

软件的版本号，这里要注意填写，用于日后的更新。

1.3 运行脚本

选择需要生成打包的 exe。

1.4 软件图标

软件显示的图标，软件默认如图 5 所示 Ico 格式的图标，如需自定义图标，可以生成 Ico 格式的图片调用。

1.5 保存配置

保存当前 exe 项目的设置，下次使用可直接调用修改。

1.6 软件预览

预览当前设置下，软件生成后的界面。

1.7 生成 exe

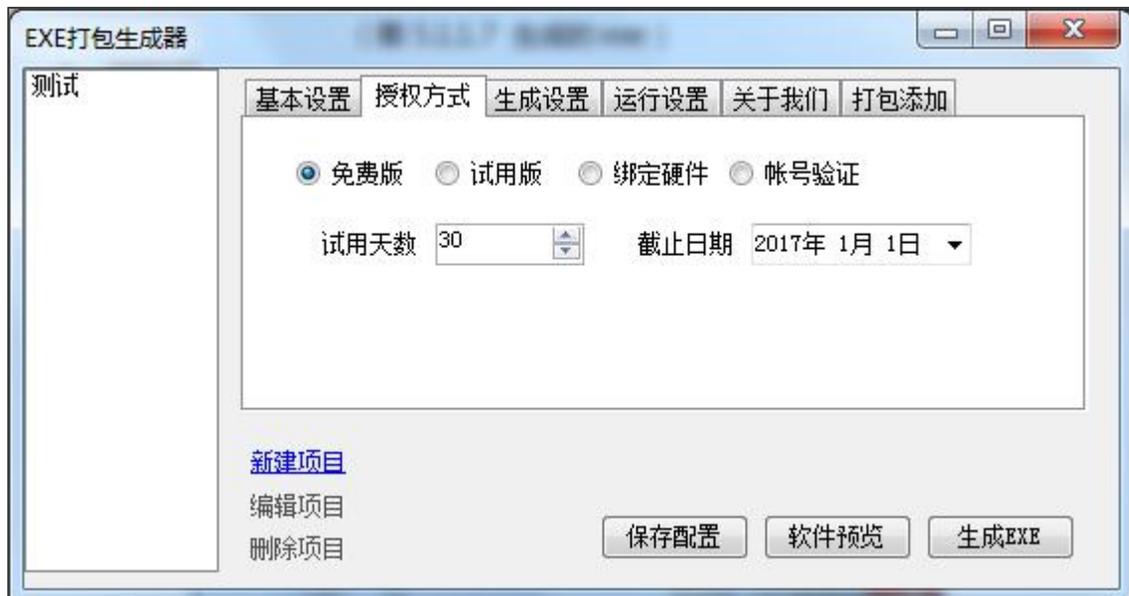
设置完成后，点击此按钮，即可生成 exe，如图 5.1.1.7。

名称	修改日期	类型	大小
Configuration	2016/11/28 16:17	文件夹	
Plugins	2016/11/28 11:54	文件夹	
Projects	2016/11/28 16:17	文件夹	
RunTime	2016/11/28 11:54	文件夹	
Temp	2016/11/28 16:17	文件夹	
Config.ini	2016/12/2 15:39	INI 文件	1 KB
RunTime32.dll	2016/11/25 13:45	应用程序扩展	920 KB
TasksRunner.exe	2016/11/25 13:45	应用程序	19 KB
示例exe.exe	2016/11/28 11:54	应用程序	370 KB

(图 5.1.1.7 生成的 exe)

2、授权方式

生成 exe 一共有四种授权方式，不同的授权方式绑定电脑的方式不同。



(图 5.1.2 授权方式)

2.1 免费版

不绑定电脑，可以分发任意电脑使用。

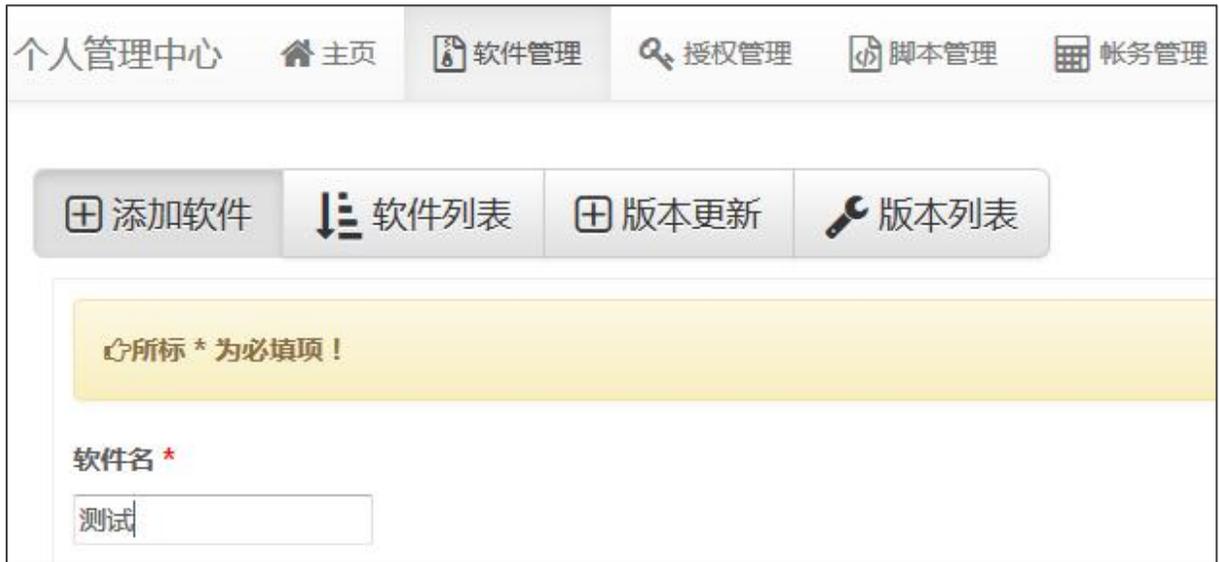
2.2 试用版

试用版本，用户只能使用一定的时间。使用天数以及截至日期都是可以设定的，其中试用天数和截至日期都设置的情况下，日期近选项生效。

2.3 绑定硬件

绑定机器码，不可任意更换电脑，只能在当前绑定电脑使用，使用方式如下。

(1) 在官网(www.locoyposter.com)使用商业版用户名登录个人中心，添加一个软件，软件名称需与该 exe 基本设置的软件名称一致。



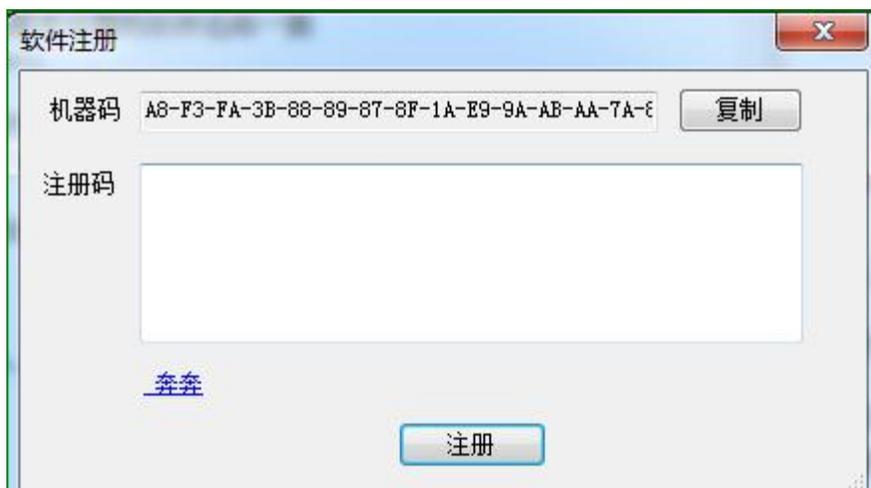
(图 5.1.2.3-1 添加软件)

(2) 选择版本更新栏目，选择对应软件名称，上传生成的 exe 文件，软件版本号需要写一致。



(图 5.1.2.3-2 上传文件填写版本)

(3) 点击生成的 exe 文件，会弹出软件注册框，复制弹出机器码。



(图 5.1.2.3-3 复制机器码)

(4) 选择授权管理，添加机器码栏目，将复制的机器码填入页面，并选择对应的软件名称，写值授权用户名称，选择开始日期以及结束日期。



(图 5.1.2.3-4 添加机器码)

(5) 点击提交后，在管理机器码栏目，复制生成的授权码，填写如 exe 的弹出注册框内，点击注册，即能正常打开 exe。

软件名	机器码	授权码
测试	A8-F3-FA-3B-88-89-87-8F-1A-E9-9A-AB-AA-7A-80-1A-C4-C5-77-B7-CA-DE-0B-FB-95-6D-3B-81-44-4E-34-29	ABZRX1BIVy8CB1FFVywHFgcQDxpaTwxFURYFFNFVWUFa1QzBmEGYwRmUEZxVaVB/ALYFwVeoXrdV2ADgUUBWkgBnUWBQYFczAjRRMVdjB20HMw9pWm4MZVFiBWJTYVUzBWtUOwY3BmUEYIA+V2hQNwA0BWFxNF5iVWMAM1ExVjcAlIEOU DRXMwJgUStXYQdmB3sPb1o+



(图 5.1.2.3-5 复制授权码)

2.4 账号验证

使用用户名以及密码登录，一个用户名同时只能登录一台机器，更换机器登录时，另一台电脑会被自动下线。

(1) 在官网(www.locyoposter.com)使用商业版用户名登录个人中心，添加一个软件，软件名称需跟 exe 基本设置的软件名称一致，如图 5.1.2.3-1。

(2) 选择版本更新栏目，选择对应软件名称，上传生成的 exe 文件，软件版本号需要写一致，如图 5.1.2.3-2。

(3) 授权管理选择添加用户：如图 5.1.2.4-1。

添加机器码 管理机器码 添加用户

软件id *
测试

授权用户 *
youue

授权用户密码 *
ue1234

备注
111

生效时间 *
2016/12/02

过期时间 *
2016/12/03

(图 5.1.2.4-1 添加用户)

(4) 然后使用此用户名密码，登录 exe

测试 - 1.3

用户名 youue

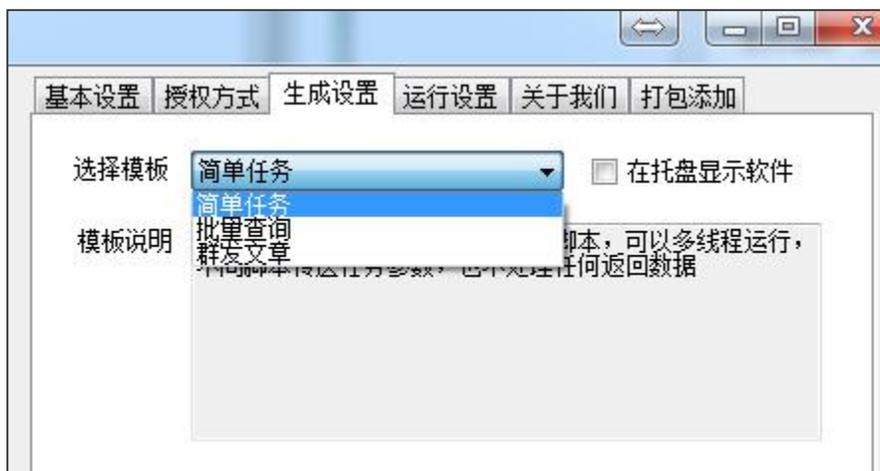
密码 ●●●●●●●

登录

(图 5.1.2.4-2 登录 exe 界面)

3、生成设置

如图 5.1.3 可选择生成的模板有三种，其程序运行的原理都不同。



(图 5.1.3 选择模板)

3.1 简单任务

用户无需从 exe 中，而是从外部导入数据，即可选择简单任务模板。

3.2 批量查询

如需从外部导入文本中数据（文本中数据是一行一个保存），并将保存数据传值给脚本中变量，则选择此模板。

(1) 生成要求：要选择此模板，脚本中必须有一个勾选只读的名为“关键词”的文本型变量。文本中读入的值是一行一个传递给这个文本型变量的。

(2) 运行原理：exe 将文本中内容按行读入内存，然后任务运行，运行一次，将一行内容赋值给关键词变量，脚本运行运行完后，则结束当前进程，然后启动一个新的进程，将下一行内容赋值给关键词变量，直到文本中所有行的内容都运行一遍。

3.3 群发文章

用户如需群发文章，则可以使用群发文章模板。群发文章模板为选择一个文件夹，文件夹中的所有 txt 文本都将做为一个文章来发布，文章标题即为文件名，文章内容即为文件内容。可以选择发完就删除文本，也可以不删除。脚本中必须要有“标题”和“内容”两个变量。

(1) 运行原理：读取目录下的文本，文件名赋值“标题”变量，文件内容赋值给“内容”变量，任务一次读取一个，完成后则自动启动任务读取下一个文本。

4、运行设置

4.1 显示脚本细节

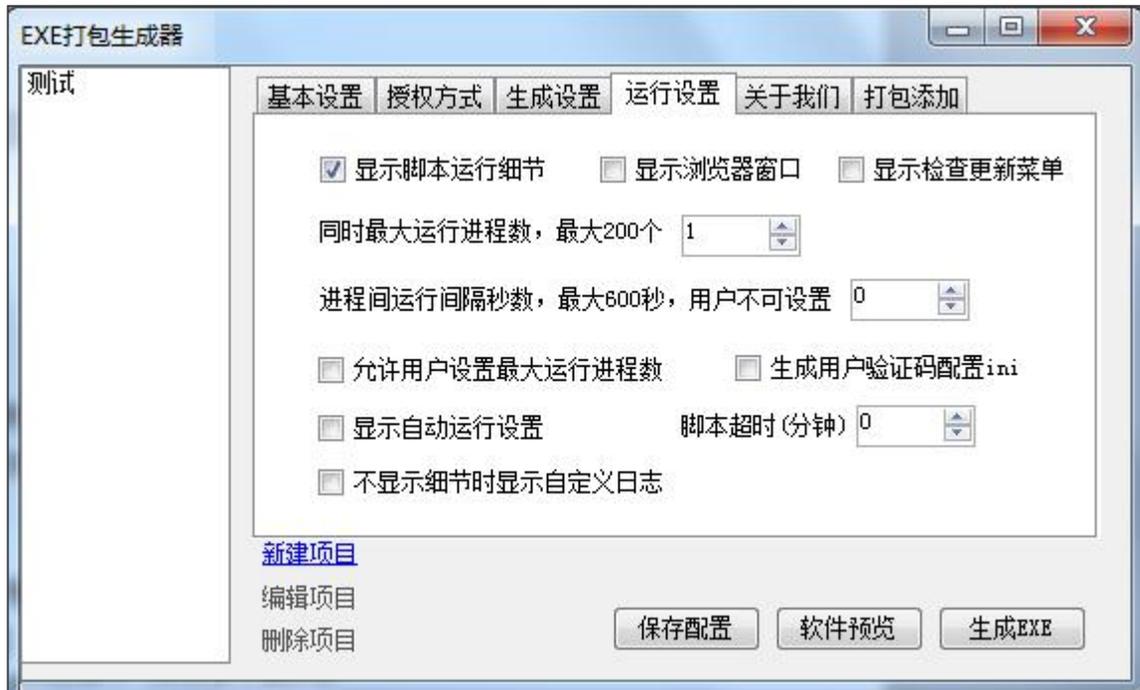
显示步骤的运行日志，不勾选，则只显示任务开始时间，以及运行是否成功及结束时间。

4.2 显示浏览器窗口

是否弹出显示 taskrunner 运行窗口。

4.3 显示检查更新菜单

是否要显示检查更新的窗口。



(5.1.4 运行设置)

4.4 同时最大运行线程数

即当前任务，可以同时运行几个线程。

4.5 运行用户设置最大运行线程数

是否让客户可以自定义同时运行线程数目。

4.6 生成用户验证码配置 INI

在 exe 目录下生成一个 Config.ini 文件，让用户可以填写验证码的用户名密码或者路径。勾选此选项，则在运行 exe 前，需要编辑 ini 文件重新填写验证码用户名密码，编写一次保存既可。

4.7 显示自动运行设置

设置 exe 运行的时间间隔，即每隔一段时间就运行一次。

4.8 脚本超时

exe 运行超过此时间，就会自动停止，为 0 则不限制。

4.8 不显示细节时显示自定义日志

是否显示自定义日志。

5、关于我们

填写注册文字、说明文字，联系信息等开发者的自定义内容。

6、打包添加

填写软件目录或文件可将其内容一并打包到 exe 中。

第六章、软件适应性

一、运行环境

系统环境：Win10/Win8/Win7/Win2012/Win2010/Win2008/Win2003/Vista/Xp

框架支持：要求电脑安装.NET4.0 框架支持

平台选项：Anycpu X64 位开发，兼容 32 位及 64 位系统

二、授权方式

火车浏览器采用网络认证进行授权，通过用户名和密码登录，在使用时需绑定电脑，但可随时自由更换，不同版本支持绑定的电脑数量（即可允许同时在线使用的电脑数）不同，可按需选择。

三、软件升级

火车浏览器官方会定期对软件进行功能的扩充和更新升级，用户在所选择的服务年限内可通过官网发布的地址下载最新版本的软件，如超出服务年限可选择续费后更新，如无需更新，原版本仍旧可以使用。

四、适应性服务

- 1、专属技术支持渠道
- 2、按需制定培训计划
- 3、定期视频培训课程
- 4、可进行 OEM 定制
- 5、支持自行扩展开发插件

五、技术支持

官网地址：<http://www.locoyposter.com>

企业 QQ：800019423

技术交流 QQ 群：257885236

官方论坛：<http://bbs.locoyposter.com>

官方教程：<http://www.locoyposter.com/doc.html>

技术邮箱：tech@lewell.cn

