

企业级分布式应用服务 EDAS

组件中心

组件中心

自动化持续集成

接口自动化测试平台是一个提供可视化接口调试、接口测试用例编写、沉淀，以及接口测试用例回归的平台，可以用来测试服务化应用的底层服务化接口。

互联网背景下，产品的迭代速度越来越高。以往 HSF/Dubbo 接口测试都需要写测试代码，一个被测接口要写几个甚至十几个测试函数（方法），工作重复，效率不高，且易出错。被测接口修改之后需要修改相关的接口测试代码，维护成本高。

接口自动化测试平台提供了更简捷、方便、高效的测试方式，能脱离代码编写，提升接口测试效率，且方便维护。同时，开发在调试接口时，可以通过界面化的方式随时调用接口，查看接口返回，提高了前后端开发的沟通效率。

平台目前支持的接口框架包括 HSF、Dubbo、HTTP、EDAS。使用这些框架的用户，在接口自动化测试平台上，可以脱离编写代码的传统模式，方便、高效地完成项目过程中的接口调试、接口测试用例编写以及项目、预发布的接口测试回归；同时还支持同模式下主干接口用例的测试与回归。



接口自动化测试是接口测试平台最重要的使用路径，其主要功能就是给开发和测试在项目过程中做接口调试用。

接口测试分为四步：**选取应用>选取服务和方法>编写用例>执行用例。**

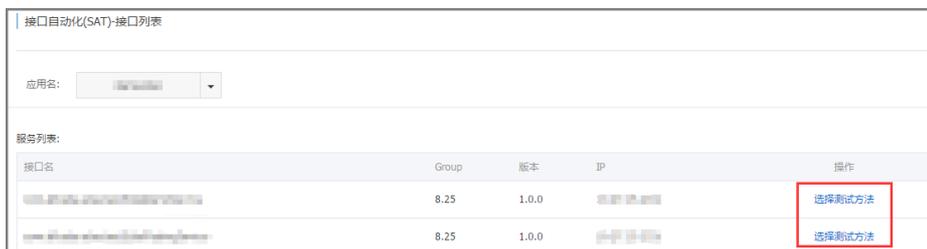
具体操作如下：

在 EDAS 控制台左侧菜单栏选择**持续集成>接口自动化**，进入到接口自动化测试界面。

在接口列表页，选择需测试的应用，如下图所示：



在服务列表中选择需测试的接口，单击右侧的**选择测试方法**，如下图所示：



在 EDAS 服务方法列表对话框，单击需要操作的测试方法右侧的**新增用例**选项。

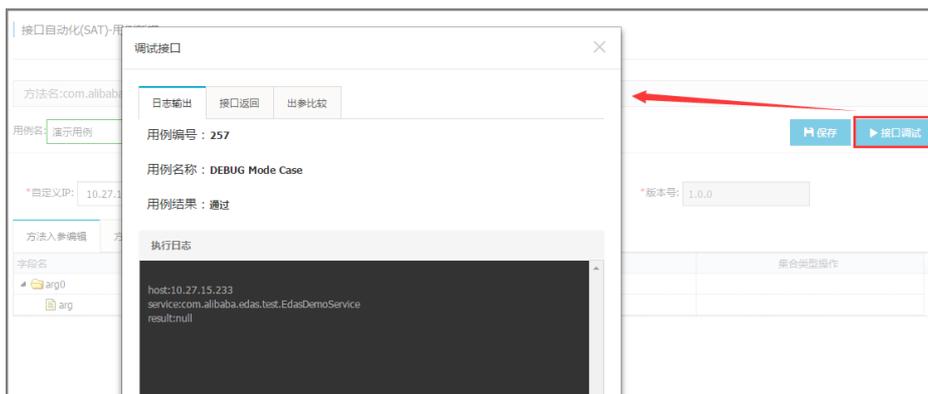


在新增用例页面，输入用例名、选择 IP 等信息，并输入入参出参等参数，单击**保存**。

- 用例名称：无限制，可任意取
- 自定义 IP：选择执行机地址



编辑好用例后，单击**接口调试**执行用例，可查看到执行日志和结果，如下图：



您也可以在**用例列表**页，选择需要执行的用例，单击右侧的执行箭头或者页面右上角的**执行**按钮执行用例：



本文详细介绍接口自动化测试平台的使用方法。

接口列表

接口列表主要用于选择应用进行用例编写。

1. 登录接口测试平台

在 EDAS 控制台左侧菜单栏选择**持续集成**>**接口自动化**，进入接口自动化测试平台。

企业级分布式应用服务

概览

▶ 资源管理

应用管理

全局配置推送

▼ 持续集成

接口自动化

UI自动化

进接口自动化测试平台，界面会有相应的提示：



2. 选择应用

进入接口自动化测试平台，首页展示可以选择应用，如下图所示：



说明：应用的添加方法，建议参考 EDAS 说明文档。

3. 查找服务和方法

选取好应用后，页面会显示应用对应的全部服务。

找到被测接口的被测版本，点击右侧的**选择被测方法**，如下图所示：



服务方法列表对话框会列出该接口的所有测试方法：

EDAS服务方法列表

服务名: [模糊输入]

方法名	参数列表	返回类型	操作
now	[模糊输入]	[模糊输入]	新增用例
sleep	[模糊输入]	[模糊输入]	新增用例
echo	[模糊输入]	[模糊输入]	新增用例

选择被测方法，单击**新增用例**按钮，可新增用例并进行调试：

接口自动化(SAT)-用例新增

方法名: [模糊输入]

用例名: 保存 接口调试

*自定义IP: [模糊输入] *分组: 8.25 *版本号: 1.0.0

方法入参编辑 方法出参校验

字段名	字段类型	入参值	集合类型操作
arg0	[模糊输入]		
arg	[模糊输入]		

4. 输入入参和选择调用方式

在新增用例页面，输入接口测试需要的入参。

字段名	字段类型	入参值	集合类型操作
arg0	ROOT		
arg	LIST		
ListArgs	OBJECT(com.alibaba.edas.testcase.api.dto.DbCon		复制 删除
port	STRING	123456789	
db	STRING		
type	STRING		
host	STRING		
user	STRING		
pwd	STRING		

SAT 支持 Telnet 调用，泛化调用和单元调用。

泛化调用和单元调用后面的 Dubbo 和 HSF 是指 HSF 的 Group，调用前需要先查看好该接口的 Group。

Telnet 调用、泛化调用、单元调用三者的不同:

Telnet 调用：Telnet 调用只适合基于 Pandora 容器的 HSF，且 Pandora 版本最好 2.0.5.2 以上。需要在参数配置中设置 Pandora 端口。

泛化调用：适合 HSF 版本 2.0 以上的接口，且接口方法的参数不能太复杂。DTO 必须是标准的 pojo，不支持多态。

单元调用：适合所有版本的 HSF 接口，支持多态以及各种复杂参数。

单元调用兼容性最好，但是单元调用需要消耗的资源比较大。执行调用时间需要 15s 以上，而 Telnet

调用和泛化调用只需要1s。单元调用适用于 Telnet 调用和泛化调用不能支持的场景。

入参函数模板 RANDOMINT(nMin,nMax) :

如果某个参数需要在1-100之间随机一个整数，则入参值填写RANDOMINT(1,100)
RANDOMSTRING(string1,string2,string3...).

如果某个参数需要在字符数组中随机一个，则填写RANDOMSTRING(aaa,bbb,ccc)。

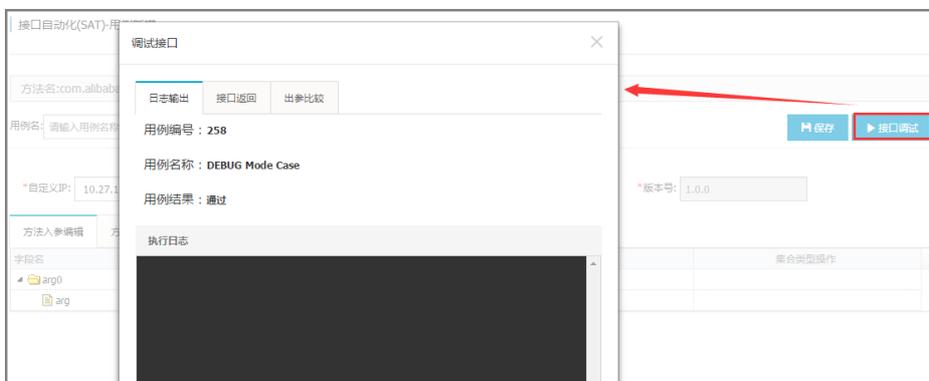
5. 出参校验

出参值校验包括**相等**和**包含**两种类型，依据接口返回的具体情况选择。对于接口调试来说，出参校验不是必须的。您可以自己查看接口返回值来判断出参是否正确，也可以通过配置出参校验，让平台来判断接口返回是否正确。



6. 接口调试和结果查看

点击**接口调试**按钮，执行接口调试，系统弹出执行结果查看窗口。



7. 保存用例

输入用例名称，点击**保存**按钮，可以保存用例。保存之后，该用例会出现在用例列表中。保存用例主要用于帮助开发保存入参，方便下一次的调试。



用例列表

单击左侧菜单栏的**用例列表**，进入主干用例管理页面。

在此页面可以批量执行选中的用例，也可以重跑失败用例，也可查看用例执行结果。

执行用例

方法一：如下图所示，该方法可以对一个或多个用例进行执行。



方法二：如下图所示，该方法只可对某一个用例进行执行。



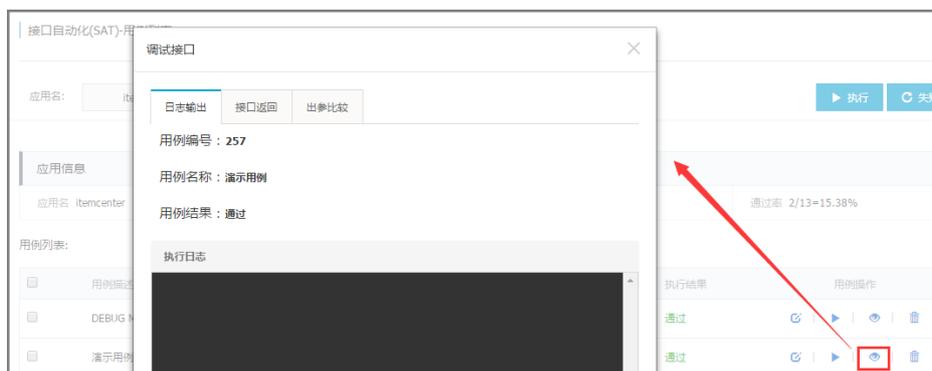
编辑用例

单击**编辑**按钮后可进入编辑页面，修改参数:



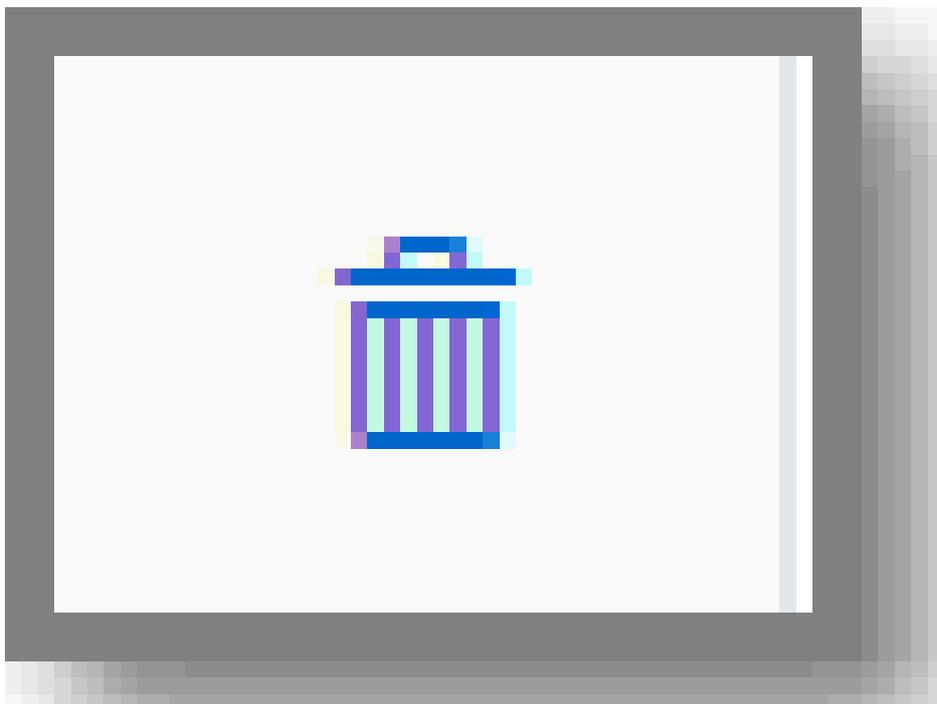
查看执行结果和日志

如下图所示：



删除用例

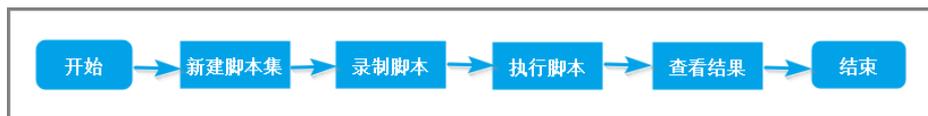
点击删除按钮，如图所示：



UI 自动化是 Web 页面自动化测试的解决方案，可以用来测试应用的前端页面。自动化插件能够记录下应用的界面功能或流程，并用于进行自动化测试。

UI 自动化主要包括在线脚本录制、脚本调试与维护等功能。脚本可以在本机回放进行功能测试，也可以在云端执行，还可定时在后台回放脚本进行项目的持续集成、业务功能的回归以及线上功能的监控等。

UI 自动化流程图：



相比其他 Web-UI 自动化测试框架，UI 自动化测试脚本的创建和维护都不需要接触脚本代码，所有交互都是在 Web 页面上在线可视化使用。具体优势如下：

- 简易的页面元素选择
- 页面中非 HTML 元素的录制，包括控件元素录制
- 纯录制产生脚本
- 强大的测试点检查
- 测试中的数据修改
- 脚本支持多浏览器
- 高级的扩展功能
- 详细的失败分类
- 参数设置
- 公共脚本

本文主要描述 UI 自动化测试的完整流程，希望以最简单明了的方式引导您快速上手 UI 自动化测试。

UI 用例脚本编写

使用 UI 自动化前需要安装 Chrome 插件，具体插件安装方法见 UI 自动化帮助文档。

步骤一：新增 UI 脚本集

UI 自动化脚本用例需要存储到某一脚本集中。脚本集即脚本的一个归类，也是用例系统里一个测试件的概念。操作如下：

1. 登录 EDAS **管理控制台**，在左侧菜单栏依次单击**持续集成**>**UI 自动化**。

在 UI 自动化-脚本集管理页面，点击右上角**新建脚本集**。



名称	用例总数	失败数	未执行数	执行时间	操作
演示脚本	1	0	1	--	编辑 删除

在新建信息页面，输入脚本集名称并单击**确定**。新增的脚本集会显示在列表中。

说明：成员可以不填。默认是当前用户，也可添加其它用户的内部账号。一般为工作邮箱的前缀，多个用户用逗号分隔。

新建信息

* 脚本集名称：
长度最长为1024个字符

成员：
成员最长1024个字符，多个成员以“,”分隔

确定 取消

步骤二：录制脚本

请按以下步骤操作：

在脚本集列表页，单击刚刚创建好的脚本集名称，进入脚本集详情页面。

在页面中间的新增脚本下拉框选择新录制选项。

在新建脚本对话框输入脚本名并单击确定。

在脚本列表中找到刚刚创建的脚本，单击录制按钮。若已安装 UI 自动化插件，会自动打开新浏览器，进入录制界面。

在浏览器地址栏或者输入框内输入需要测试的 URL，如输入 taobao.com，并单击开始测试。页面会如同正常打开，同时加载出 UI 自动化的录制框。



- 和功能测试一样，可以根据需要进行点击、输入等鼠标键盘操作。

- 如果需要断言等后台操作，可单击鼠标右键，或在操作框中将鼠标移动到相应动作上，在出现的按钮中设置相应操作也可。
- 所有动作都会录制进 UI 自动化的录制框中。
- 页面所操作的元素左上角会有对应的动作序号。如下图所示：



操作完成后，单击 UI 自动化录制框右下角的脚本保存按钮，进行二次确认后，再单击关闭窗口。



回到脚本集页面中，UI 自动化录制栏会出现**编辑**和**回放**按钮，这样一个脚本就生成完毕，详情页如下图所示：



说明：只有本地电脑安装了 UI 自动化本地执行程序，才会出现回放按钮。具体可看 UI 自动化帮助文档的“回放执行”小节。用鼠标 hover 用例名称后的详情可看刚才录制的脚本大纲。

步骤三：维护脚本

1. 在脚本集详情页，找到对应脚本，单击**编辑**按钮，AUI 会自动打开脚本录制的界面。
2. 把鼠标移动到工具框中相应动作，就会出现该动作的设置区，一般支持上移、修改、设置和删除操作。具体动作的说明请参见 UI 自动化帮助文档。

执行 UI 用例

在脚本集详情页，勾选需要执行的脚本，并选择浏览类型进行执行。

可选择全部或者批量执行（几台执行机并发执行）当前脚本集所有用例，或者勾选相应 case，再选择 **脚本执行** 下拉列表中的执行浏览器类型，进行单个 case 的执行，如下图：



脚本执行完之后，在该脚本下执行结果栏可查看到执行成功，如下图：



UI 自动化测试主要是指 Web 界面自动化测试，在使用过程中需要通过 AUI 插件进行可视化的界面录制及执行。本文主要描述插件的安装和卸载方式。

UI 自动化插件安装

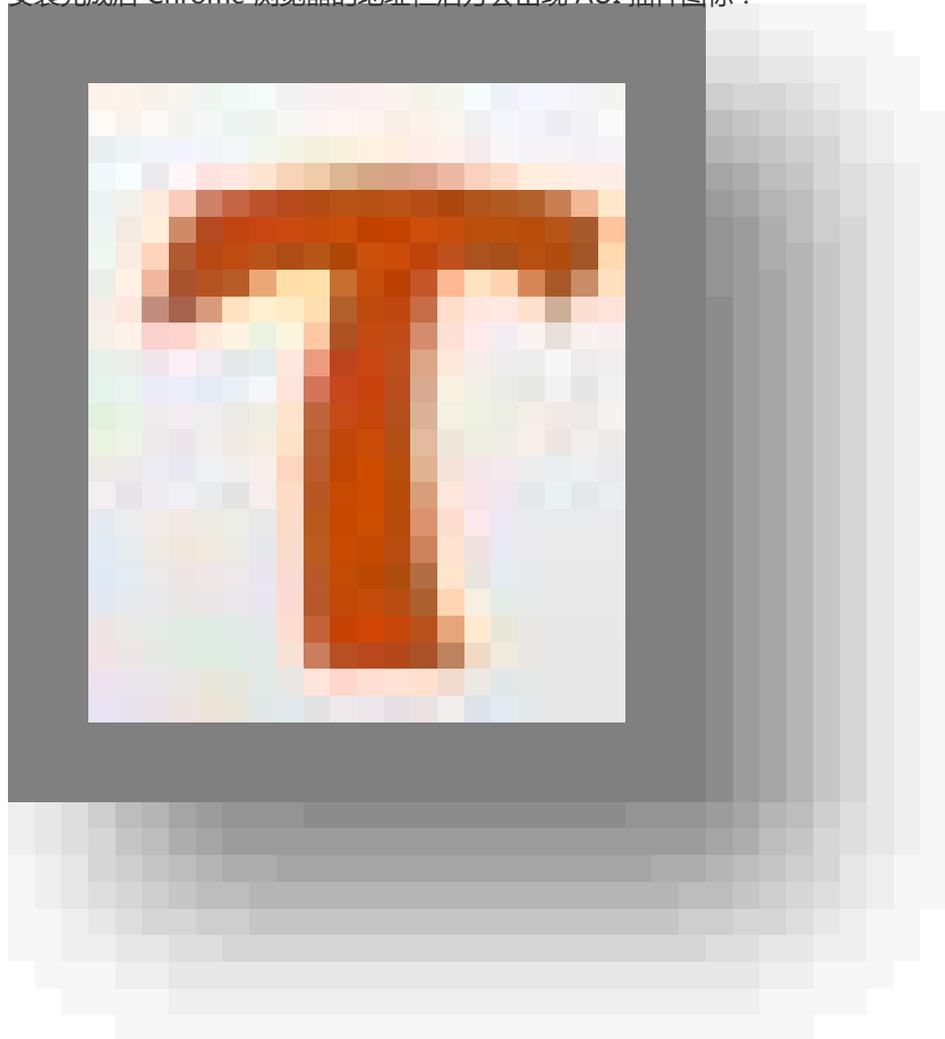
开始使用 UI 自动化测试之前，首先需要安装 AUI 插件。

注意：在安装之前，请确认已经安装了 Chrome 浏览器。

插件安装方法如下：

- 方法一：前往 Chrome 应用商店进行在线安装。
- 方法二：若 Chrome 应用商店无法访问，请参考安装说明进行手动安装。

安装完成后 Chrome 浏览器的地址栏后方会出现 AUI 插件图标：



如需卸载插件，请在 Chrome 浏览器地址栏中输入 `chrome://extensions`，找到“UI 自动化录制工具”后，即可进行关闭和卸载操作。如下图：



本文以录制一个最简单的脚本（用户登录 Aliexpress.com）为例来介绍 AUI 中最基本最常用的功能。

脚本录制

步骤一：创建脚本集

操作如下：

在 EDAS 控制台左侧菜单栏选择**持续集成>UI 自动化**。

在脚本集管理页面右上角单击**新建脚本集**，如下图：



输入脚本集名称，单击**确认**。创建好的脚本会显示在脚本集列表里。

说明：成员具有该任务的操作权限，可不填，默认自己，阿里内部用户输入域账号。



单击脚本集名称可以查看详情。如下图：

UI自动化(AU)-脚本集管理--演示脚本

执行配置 执行全部 失败重跑 当前集成频率: 无

模块列表

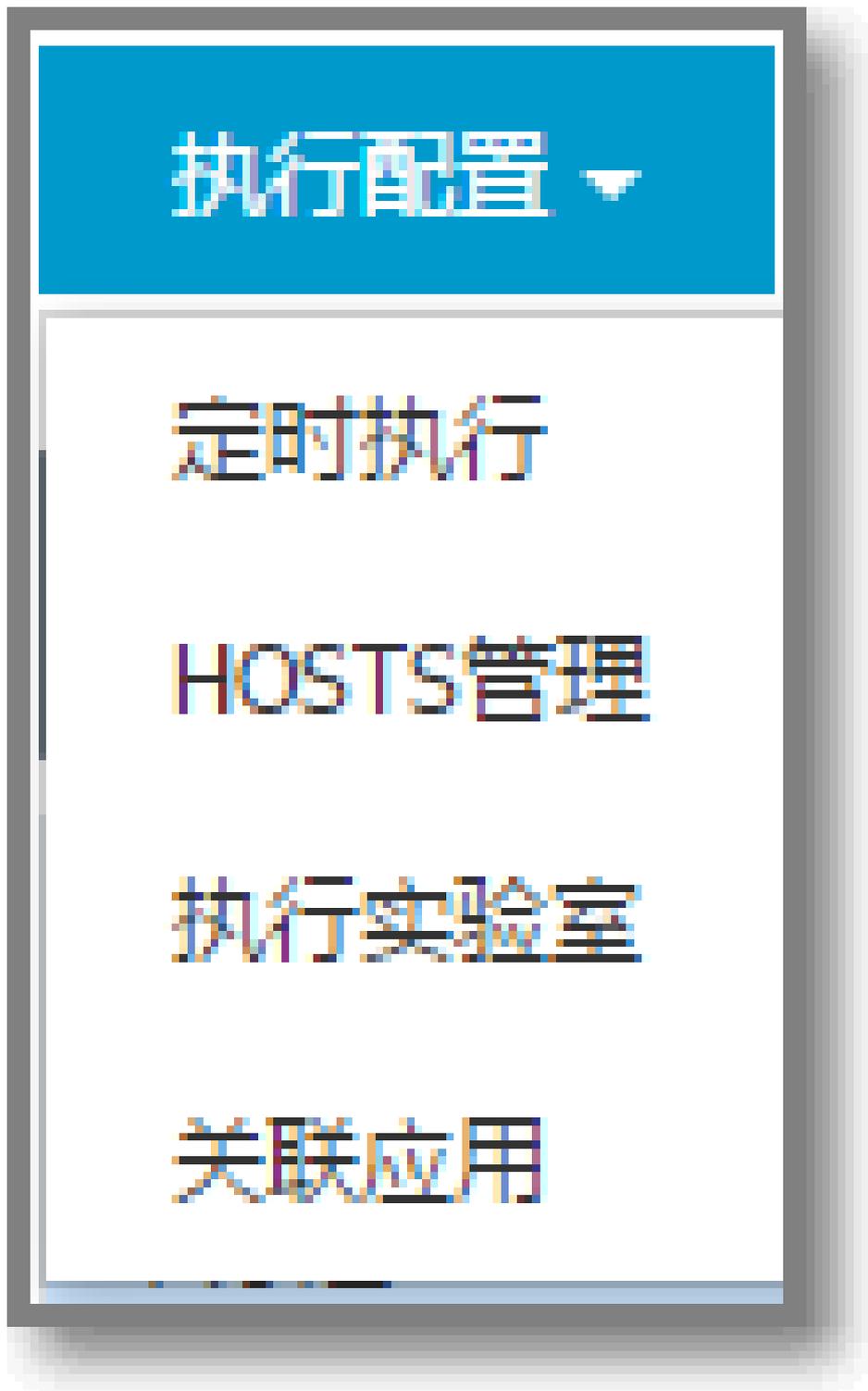
模块名称	脚本总数	成功脚本数	失败脚本数	操作
未分组	1	0	0	修改名称 删除模块 执行模块

新增脚本 脚本修改 脚本执行

脚本列表

类型设置	脚本名称	AU录制	创建者	执行结果	浏览器	执行时间	执行耗时	执行人	操作
公共	演示脚本1	录制		未运行					删除 执行趋势

说明：定义 hosts，集成设置等在**执行配置**下拉框中。



步骤二：录制脚本

本节通过一个简单的录制用例（login 过程）来演示 AUI 录制脚本的操作流程。

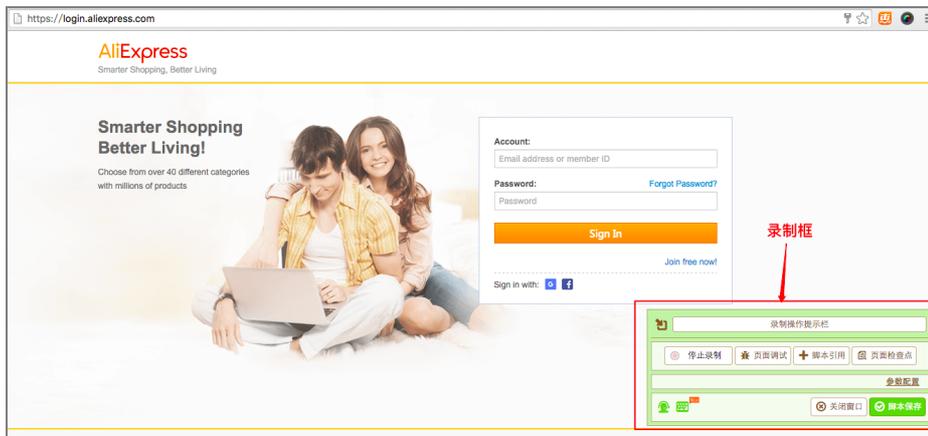
完成步骤一创建脚本集后，请按以下步骤操作：

在脚本集详情页的脚本列表里，单击**录制**。

在 AUI 浏览器页面输入录制页面 “login.aliexpress.com ” ，单击**开始录制**。如下图：



AUI 进入录制状态，待页面加载完后，显示录制工具栏。如下图：



在登录页面输入登陆账号、密码，单击**登录**。录制工具栏会记录所有录制步骤，如下图所示：

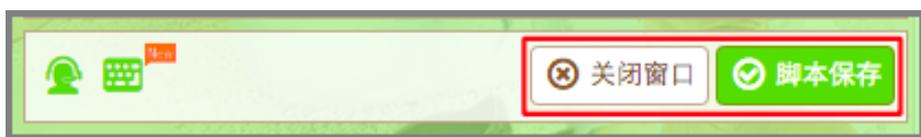


在录制工具框里，可以选中动作后可对其进行相关设置，包括修改、上移、参数配置、删除，如下图所示：



说明：特性配置详细介绍请参考下文。

单击脚本保存>关闭窗口，完成脚本录制。如下图所示：



在脚本列表上方单击脚本执行，选择浏览器进行执行。这时录制状态已更新为编辑状态。如下图所示：



点击运行结果查看详情。如下图所示：



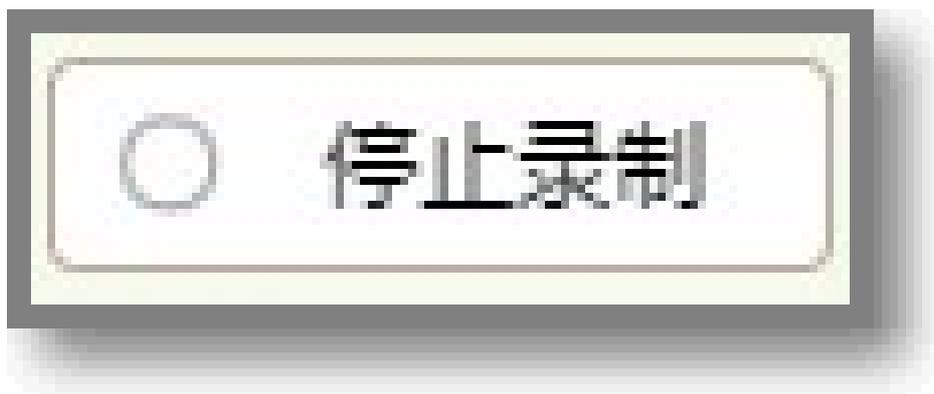
脚本录制基本功能

1. 录制功能开关

打开录制功能（未录制状态）



停止录制功能（录制状态）



使用说明：新建时默认为录制状态，编辑时默认未录制状态。可以用空格快捷键切换状态。

2. 选择一个页面元素（控件）

在录制模式下，将鼠标移动到页面上可视的页面元素，就会出现黄色的选择框。单击选择这个元素，即可进一步录制针对该元素的操作过程。



3. 录制工具栏（toolbox）的说明

录制工具栏如下图所示：



1. **页面调试**：录制/编辑状态，快速的进行步骤回放，默认从选择的步骤调试到最后一页的最后一个步骤，也可进入指定页进行回放；
2. **脚本引用**：录制/编辑状态，快速引入公共脚本，以及参数。引入后自动执行到最后一步；
3. **页面检查点**：录制/编辑状态，创建页面级的检查点：页面 URL 检查、数据库表内容检查、cookie 值检查、弹出框检查；
4. **参数配置**：录制/编辑状态，可查看 / 新增 / 编辑四种参数类型：静态参数 / 数据库参数 / 驱动参数 / 页面参数，支持脚本中各输入内容的数据替换；
5. **自定义编码**：录制/编辑状态，支持查看已录制动作代码或插入编码动作；
6. **问题反馈**：录制/编辑状态，快速反馈问题到后台；
7. **录制操作提示栏**：录制中的操作提示。

4. 右键高级功能说明

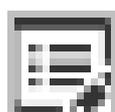
录制过程中，单击右键可以看到15个高级功能选项，如下图所示：



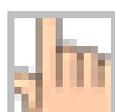
元素检查点



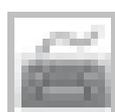
数据中心使用



参数使用



mouseover录制



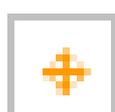
模拟键盘输入



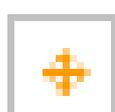
模拟左键单击



右键单击录制



元素拖拽录制



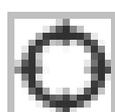
滚动录制



文件上传录制



日期控件录制



元素精确选择



元素参数创建

1. **元素检查点**：对元素设置检查点包括 URL 检查（正则匹配），cookie 值检查、弹出框检查；
2. **数据中心使用**：调用数据中心的测试数据；
3. **参数使用**：直接引用已经创建的参数；
4. **Mouseover 录制**：对 mouseover 事件进行录制，系统自动默认录制；
5. **模拟键盘输入**：录制富文本框（带有输入格式的文本）和嵌入式对象的输入操作；
6. **模拟左键单击**：录制选中控件的左键单击操作；
7. **右键单击录制**：模拟右键单击录制；
8. **元素拖拽录制**：用于录制控件的拖放操作；
9. **滚动录制**：用于录制界面滚动操作；
10. **文件上传录制**：用于录制文件上传操作；
11. **日期控件录制**：用于录制日期控件，支持相对日期和固定日期两种录制；
12. **元素精确选择**：可视化的列出选中控件的父结点链或子结点链内容，进行精确选择录制；
13. **元素变量创建**：用于对元素的部分内容设置为变量，或对元素属性值的部分内容设置为变量；
14. **鼠标左键单击**：模拟左键单击录制；
15. **鼠标 mouseover**：对 mouseover 事件进行录制，系统自动默认录制。

回放执行是指将录制好的脚本在本机进行回放调试，包括代码回放和单步调试两种模式。

下面介绍回放执行操作的步骤：

安装 AUI 本地执行程序。

如果系统没有安装或者开启 agent，则会自动的弹出提示要求安装。



安装执行软件。

- Windows 64 位操作系统：下载安装包，完成后解压 ZIP 包，右键管理员身份运行 AuiAgent.exe。
- Windows 32 位操作系统：下载安装包，完成后解压 ZIP 包，右键管理员身份运行 AuiAgent.exe。
- MAC 或 Linux 操作系统：下载 JAR 包（32.5MB），执行以下命令进行安装（需要自行安装 Java 执行环境）：

```
nohup java -agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=5600 -jar aui.agent-1.0.0.jar &
```

注意：安装路径务必要包含非英文字符。如果出现问题，请删除 `auinstall` 注册表项，然后重新安装，就会重新选择路径。

如需卸载执行软件，请打开 UI 自动化的默认安装目录，即“用户根目录”下的 `auinstall` 文件夹，双击 `uninstall.exe` 进行卸载。

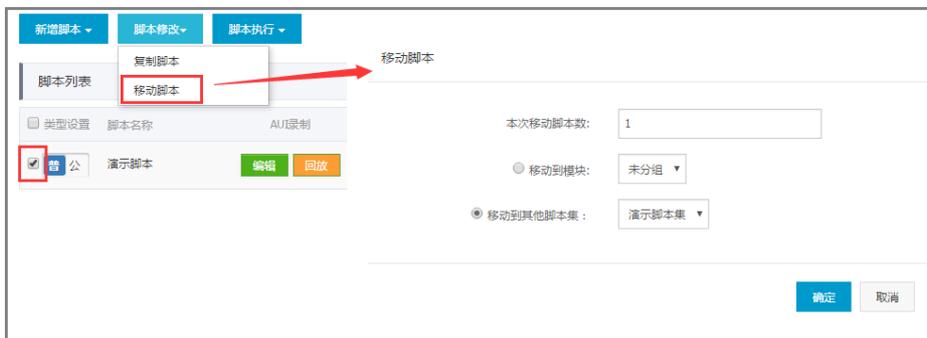
UI 页面中可对脚本进行统一管理，本文主要对界面的每个功能项进行一个说明。

页面模块

可以对脚本分组管理，如下图：



脚本可在不同模块中移动，也可在不同脚本集中移动，如下图：



选中脚本集（高亮显示），可在该脚本集中新增脚本。

注意：脚本默认在未分组中。

1. 新增脚本集

对脚本进行分组管理，可添加多个成员，如下图所示：



The screenshot shows a dialog box titled "新建信息" (New Information) with a close button in the top right corner. It contains two input fields: "脚本集名称" (Script Set Name) and "成员" (Members). The "脚本集名称" field has a red asterisk and a note below it stating "长度最长为1024个字符" (Maximum length is 1024 characters). The "成员" field has a note below it stating "成员最长1024个字符，多个成员以','分隔" (Maximum length is 1024 characters per member, multiple members separated by commas). At the bottom right, there are two buttons: "确定" (Confirm) and "取消" (Cancel).

2. 新增脚本

新增脚本包括两部分：新录制和引入脚本。如下图所示：



- 新录制脚本，只需要输入脚本名称；
- 引入脚本，可将其它脚本集中的脚本引入进来，达到多处维护的作用。

3. 修改脚本

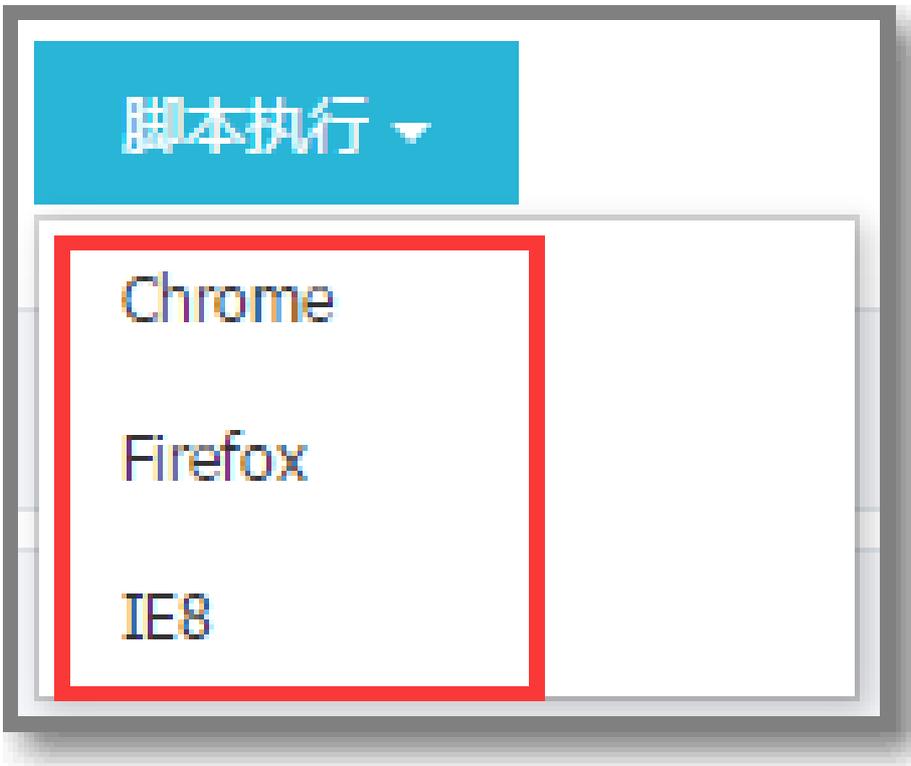
修改脚本包括两部分：复制脚本和移动脚本。

- 复制脚本，只能选择一个脚本进行复制，复制到当前任务集中；
- 移动脚本，支持脚本在不同脚本集，不同模块之间移动，选择对应的复制目标地址即可；但是脚本集移动，不支持到目标脚本集的某一模块。



4. 脚本执行

脚本支持三种浏览器的运行，Chrmoe/Firefox/IE8。运行前先勾选脚本，详见下图：



5. 执行配置

脚本集的执行配置包括4部分：定时执行、hosts 管理、执行实验室、应用关联，如下图所示：



定时执行：针对整个脚本集，设置定时执行时间、频率、以及通知方式。

定时设置

是否集成:

运行时间: 星期一 星期二 星期三 星期四 星期五 星期六 星期日
 点 分 间隔 小时执行

邮件通知:

执行浏览器:

执行机组:

其他邮件收件人:

Hosts 管理：添加任务在云端执行时的环境 hosts 地址。

执行实验室：可配置不同与脚本集的 hosts、定时执行时间、执行浏览器，建议用作回归或集成任务。



6. 运行结果通知

脚本立即执行后会发旺旺弹出框消息（干牛暂不支持），集成运行结果会根据通知设置发送通知。

公共脚本库

1. 新增公共脚本库

公共脚本支持被其他脚本引用，但公共脚本不能引用公共脚本。

方法一：新增脚本时，可设置是否为公共脚本，如下图所示：

新建脚本

脚本名称:

模块:

是否设定为公共脚本:

确定 取消

方法二：支持在脚本列表中直接选择，如下图所示：

脚本列表

类型设置	脚本名称
<input type="checkbox"/>	
<input checked="" type="checkbox"/>	演示脚本

2. 公共脚本引用

录制中，可直接引用公共脚本包括，个人的公共脚本和公共库的公共脚本。



3. 公共脚本管理

对公共脚本进行管理步骤如下：

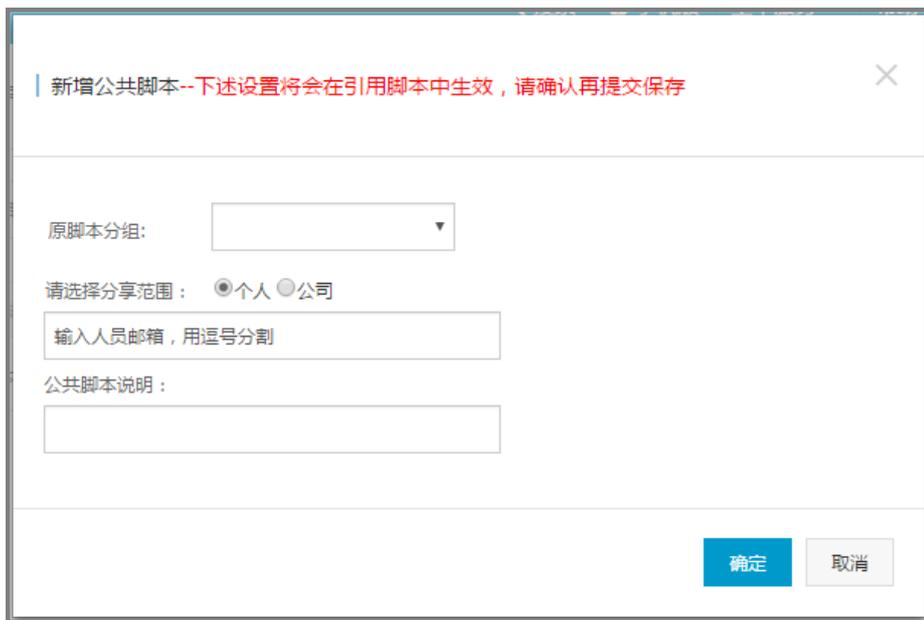
点击公共脚本库。



进入公共脚本管理页面，包括个人和公司的公共脚本。



点击制作公共脚本可快速设置公共脚本。



新增公共脚本--下述设置将会在引用脚本中生效，请确认再提交保存

原脚本分组:

请选择分享范围: 个人 公司

输入人员邮箱，用逗号分割

公共脚本说明:

确定 取消

点击**用例管理**页面可快速进入 UI 任务管理页面。

UI 脚本录制时录制成功的步骤在录制提示框中都会有对应的动作提示，下面我们进行详细的讲解下功能的使用及操作提示。

录制操作步骤

1. 单击操作步骤

录制单击操作步骤。

操作如下：

- 方法一（常用）：录制中，点击选中的控件，“录制工具栏”中显示步骤信息：步骤 N：点击控件【xxx】；
- 方法二(个别控件常用方式录制失败时)：
 - 录制中，选择一个页面元素；
 - 鼠标右键选择**模拟左键单击**；
 - “录制工具栏”中显示步骤信息：步骤 N:点击控件【xxx】。

方法一和方法二的主要区别在于：

- 方法一点击页面元素后，AUI 会先执行点击操作再录制脚本；
- 途径二则是先录制脚本再执行点击。

在某些特殊场景中，点击操作结束后，页面元素可能会消失，就会导致途径一会失败，只能选择途径二。

2. 填写操作步骤

录制填写操作步骤。

前提：被录制输入框（保持原始内容，如空内容）的状态下被选中。

操作如下：

1. 录制中，选择一个输入框控件，保证输入框内容为原始内容状态下被选中；
2. 在输入框中输入数据；
3. 点击页面其它部位，使输入框失去焦点，“录制工具栏”中显示步骤信息：步骤 N: 【xxx】控件输入值【aaa】。

3. 数据中心使用

录制填写操作，使用的数据由数据中心生成（依托数据银行）。

操作如下：

1. 录制中，选择一个输入框控件；
2. 鼠标右键选择**数据中心使用**，显示数据中心任务信息；
3. 选择数据任务，如“会员 ID(ifm)”，显示任务执行中；
4. 数据中心生成的数据自动填写到输入框，“录制工具栏”中显示步骤信息：步骤 N: 【xxx】控件输入值【DB[[aaa]]】，其中“DB[[]]”表示内容为数据中心生成。

4. 下拉框选择操作

录制下拉选择框的选择操作步骤，只支持“select”元素的选择操作。

操作如下：

1. 录制中，选择一个下拉选择框；
2. 点击下拉选择框，选择其中某个值；
3. “录制工具栏”中显示步骤信息：步骤 N: 【xxx】控件选择值【aaa】。

5. 双击操作步骤

录制某控件的双击操作步骤。

操作如下：

1. 录制中，选择一个控件；
2. 双击选中的控件，“录制工具栏”中显示步骤信息：步骤 N: 双击控件【xxx】。

6. 文件上传录制

录制文件上传的操作步骤，只支持【input type=" file" 】实现的文件上传功能。

操作如下：

1. 录制中，选中上传文件控件；
2. 鼠标右键—>打开**文件上传录制**；
3. 点击**文件上传**，从本地选择相应的文件；
4. 点击**确定**按钮；
5. 录制成功后，“录制工具栏”中显示步骤信息：步骤 N: 控件【xxx】上传文件【yyy】。

7. Mouseover 事件操作

录制选中控件的 mouseover 事件操作步骤。操作如下：

- 方法一（自动录制）
 - i. 录制中，直接点击选择控件；
 - ii. 录制成功后，“录制工具栏”中显示步骤信息：步骤 N: Mouseover 执行【系统自动创建】。
- 方法二（选择录制）
 - i. 录制中，鼠标右键单击选择的控件；
 - ii. 选择**mouseover 录制**；
 - iii. 录制成功后，“录制工具栏”中显示步骤信息：步骤 N: Mouseover 执行。

8. 模拟键盘操作步骤

录制文本输入操作步骤，模拟键盘输入。

操作如下：

1. 录制中，先确保选中目标元素；
2. 鼠标右键->选择**模拟键盘录制**；
3. 输入内容后，点击**确定**；
4. 录制成功后，“录制工具栏”中显示步骤信息：步骤 N: 【xxx】输入值【text】。

9. popup 弹出框检查

录制系统弹出框操作步骤，支持 popup 弹出框是否存在的校验。

操作如下：

1. 录制中，出现 popup 弹出框；
2. 点击录制工具栏中 **页面操作>页面检查点>弹出框检测** 选择是否存在；
3. 录制成功后，“录制工具栏”中显示步骤信息：+系统弹出框（PoPuP）检测。

10. 日期控件操作

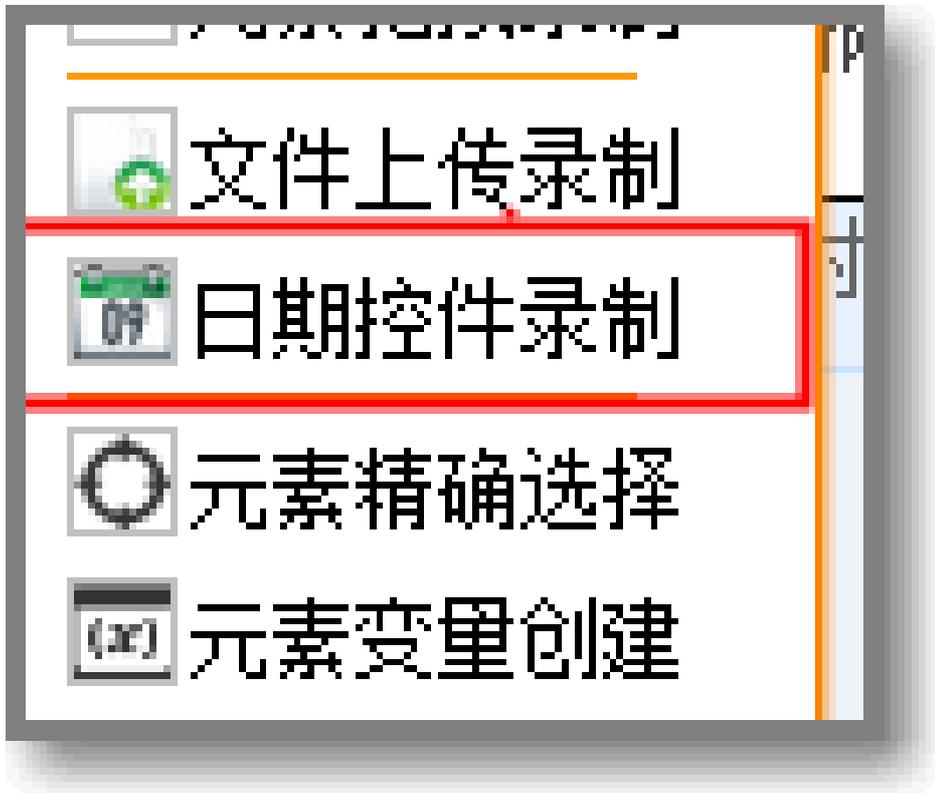
录制日期控件。

操作如下：

停止录制的状态下，选择好日期；

开始录制，选择日期控件；

右击菜单中，选择日期控件录制；



显示固定值，也可以自动以设置；

点击确认后，已录制动作记录页面显示。



11. 控件精确路径选择

采用直观可视化的方式列出选中控件的父结点链或子结点链内容。

操作如下：

右键选中页面中控件元素；

选择元素精确选择，对话框显示当前选中控件结点及父结点列表；



点击**获取子节点**按键，路径栏会显示当前节点的子节点列表：



点击**获取父节点**，路径栏会显示当前节点的父节点列表；

点击路径栏列表项，切换被选中的控件。

录制检查点



1. 页面 URL 检查

录制页面 URL 检查步骤。

前提：当前页面无被选中控件。

操作如下：

1. 将鼠标移动到工具箱中**页面操作**上方，点击**页面检查点**按键，显示创建变量对话框；
2. 点击**页面 URL 检查**；
3. 编辑预期结果内容，点击**确定**，“用例录制信息”中当前步骤信息增加“检查点”次数。

2. 数据库表内容检查

录制数据库表查询结果检查步骤。

注意：SQL 字符串必须以 ‘`’` 结束。

操作如下：

1. 将鼠标移动到工具箱中**页面操作**，点击**页面检查点**按键，显示创建变量对话框；
2. 点击**数据库表内容检查**；
3. 选择数据库名称,显示 SQL 填写文本框；
4. 编写 SQL，可以同时编写多条查询 SQL 语句；
5. 点击**执行**，显示执行结果列表；
6. 点击**确定**，“用例录制信息”中当前步骤信息增加“检查点”次数。

3. Cookie 检查

录制 Cookie 的内容检查，主要检查当前 Cookie 中是否包含特定项。

操作如下：

1. 将鼠标移动到工具箱中**页面操作**，点击**页面检查点**按键，显示创建变量对话框；
2. 点击**Cookie 值检查**；
3. 弹出面板中展示所有的 Cookie 值；
4. 选中要检查的 Cookie 项（可以多选）
5. 点击**确定**，“用例录制信息”中当前步骤信息增加“检查点”次数。

4. 弹出框检查

录制弹出框检查，主要检查当前是否有弹出框弹出。

操作如下：

1. 将鼠标移动到工具箱中**页面操作**，点击**页面检查点**按键，显示创建变量对话框；
2. 点击**弹出框检查**；
3. 弹出面板中有单选框：**存在/不存在**；
4. 选择**存在** 或者 **不存在**；
5. 点击**确定**，“用例录制信息”中当前步骤信息增加“检查点”次数。

参数设置



支持五种参数类型：元素参数 / 静态参数 / 数据库动态参数 / 页面动态参数 / 脚本驱动参数。其中元素参数需要选中对应的控件元素创建，不在上图中添加。

1. 元素参数

创建一个变量，值来源于页面某控件元素。

操作如下：

1. 选中一个页面控件；
2. 右键单击此控件，选择**元素变量创建**；
3. 根据元素属性，展示其可创建参数的属性；
4. 选择需要创建的属性，点击**下一步**；
5. 在“控件属性值”栏，点击选择属性值部分内容，则“正则表达式配置”栏自动生成正则表达式；
6. 修改正则式，直到执行结果显示需要的文本内容；
7. 填写变量名称（在 case 范围唯一），点击**确认添加**；

8. “用例录制信息”中当前步骤信息增加“变量”次数。

2. 页面元素参数

创建一个变量，值来源于页面相关特征值。

操作如下：

1. 点击**参数配置**按钮；
2. 点击**新增参数**，选中 **页面动态参数**；

目前支持三种：页面 URL、随机值、数据中心（依托数据银行）；



在“控件属性值”栏，点击选择属性值部分内容，则“正则表达式配置”栏自动生成正则表达式；

修改正则式，直到执行结果显示需要的文本内容；

填写变量名称（在 case 范围唯一），点击**确认添加**；

“用例录制信息”中当前步骤信息增加“变量”次数。

3. 数据库动态参数

创建一个变量，值来源于数据库表的查询结果。

数据库操作
✕

数据库: 编辑

sql语句: 执行

只展示sql执行结果中的第一条数据

sql中建议用单引号，双引号请用\转义

结果值变量设置:

参数值	参数名	操作

取消
保存

操作如下：

1. 点击**参数配置**按钮；
2. 点击**新增参数**，选中 **数据库动态参数**；
3. 选择数据库，没有数据库可点击新增选项进行新增；
4. 新增后，选中对应数据库；
5. 编写一条 SQL，点击**执行**，显示执行返回值；
6. 删除不需要的参数，修改参数名，点击**保存**；
7. “用例录制信息”中当前步骤信息增加“变量”次数。

4. 脚本驱动参数

创建一个脚本驱动参数，可支持同一脚本根据参数值运行多次。

操作如下：

1. 点击**参数配置**按钮；
2. 点击**新增参数**，选中**脚本驱动参数**；
3. 在参数列表的空行中，添加参数名，并输入对应的驱动值，如“玩具|游戏”；
4. 鼠标离开焦点即保存。

5. 静态参数

创建一个变量，值可自定义字符，可配合公共脚本功能，作为脚本对外暴露可修改的参数。

操作如下：

1. 点击**参数配置**按钮；
2. 点击**新增参数**，选中**脚本驱动参数**；

3. 在参数列表的空行中，添加参数名，并输入对应的参数值；
4. 鼠标离开焦点即保存。

6. 使用参数

录制页面中所有输入框中，可使用参数。

操作如下：

1. 如打开编辑文本，修改 URL 等输入框；
2. 将需要修改为参数的地方，修改为参数列表中的参数，按照格式：`${参数名}`；
3. 保存后，脚本调试与运行时，会将此处用参数值替代。

7. 删除参数

普通参数

参数列表中，有“删除”图标的普通参数。

操作如下：

1. 点击**参数配置**按钮；
2. 点击操作栏中的“删除”图标，鼠标移开，删除成功。

特殊参数

参数列表中，没有“删除”图标的普通参数。

操作如下：

1. 点击**参数配置**按钮；
2. 操作栏中有“修改”图标的参数，可点击“修改”图标，弹出参数设置框，在参数设置框中删除；
3. 操作栏中没有操作的参数，为页面或元素参数，需要在录制工具框中相应的步骤中点击删除。

可视化维护



1. 插入步骤

在录制好的脚本中插入一个步骤。

操作如下：

1. 在“已录制动作记录”页面，选中(单击)需要插入的位置之前的一个步骤；
2. 点击**开始录制**，录制你需要插入的步骤；
3. 录制完成后，点击**脚本保存**。

2. 修改步骤中输入值

在已有的步骤上修改输入值。

操作如下：

在“已录制动作记录”页面，选中



改的内容。

后（单击）需要修

提交，完成。

3. 设置步骤

在已有的步骤上自定义设置。

操作如下：

在“已录制动作记录”页面，在【动作 X】后，点击



, 进行步骤上自定义设置。



4. 设置页面初始化

页面在用例中的名称更改。

操作如下：

在“已录制动作记录”页面，在【页面 X】，后点击



始化；

, 选择页面数据初

根据三种不同的初始化步骤进行操作。

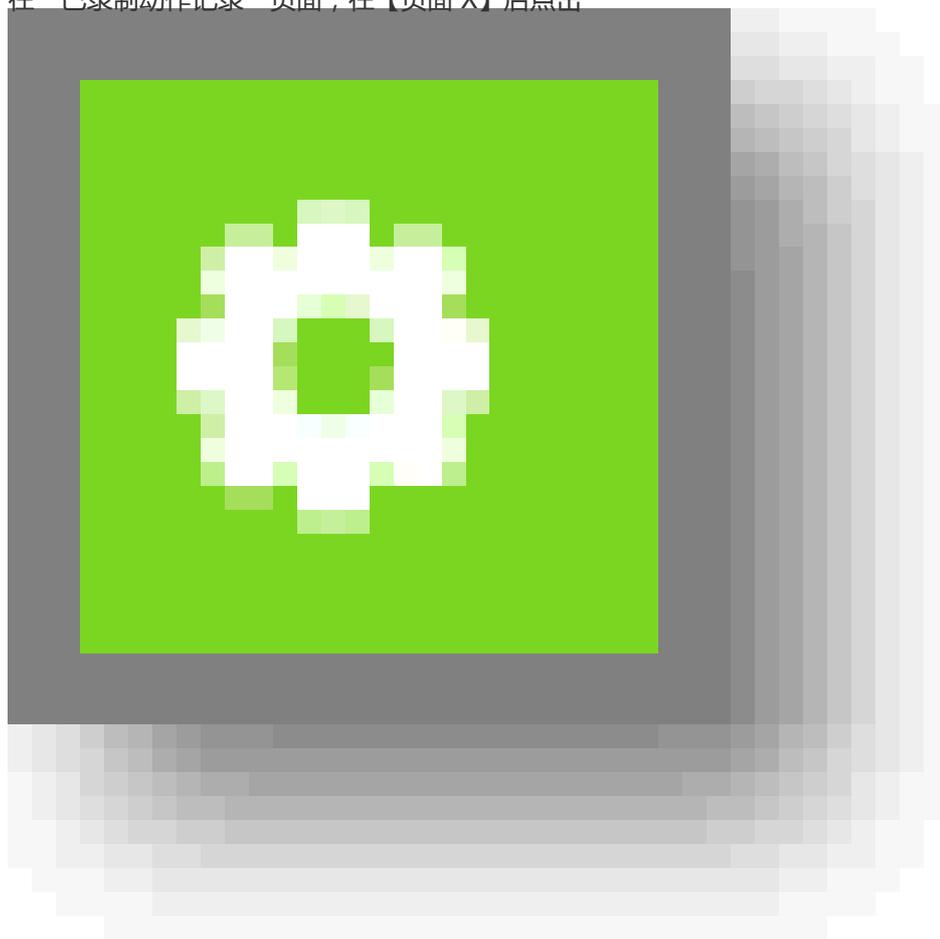


5. 设置页面名称和信息

页面在用例中的名称更改。

操作如下：

在“已录制动作记录”页面，在【页面 X】后点击



设置。

, 进行页面自定义

页面设置

是否手动输入URL?

页面URL:

页面名称:

自定义描述:

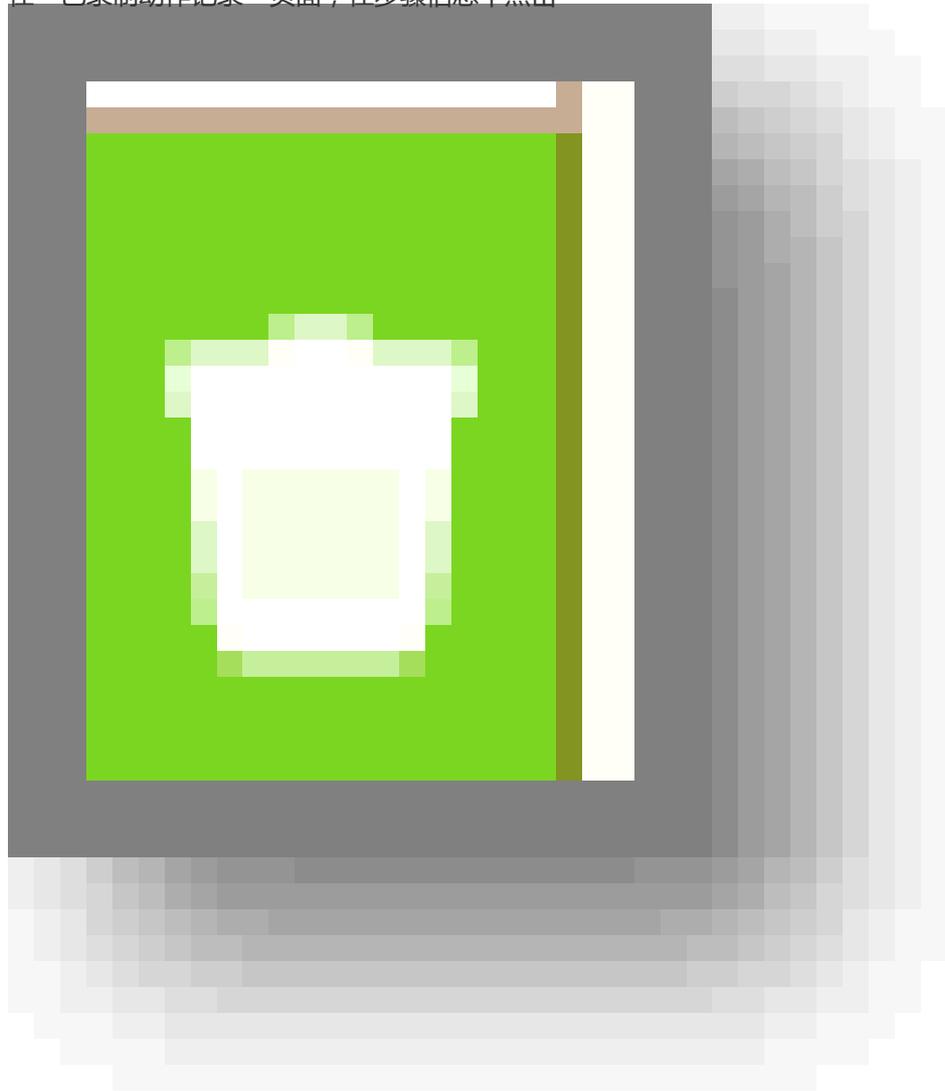
修改其中的页面名称和描述即可。

6. 删除步骤

删除一个已经有的步骤。

操作如下：

在“已录制动作记录”页面，在步骤信息中点击



，即可删除一个步骤。

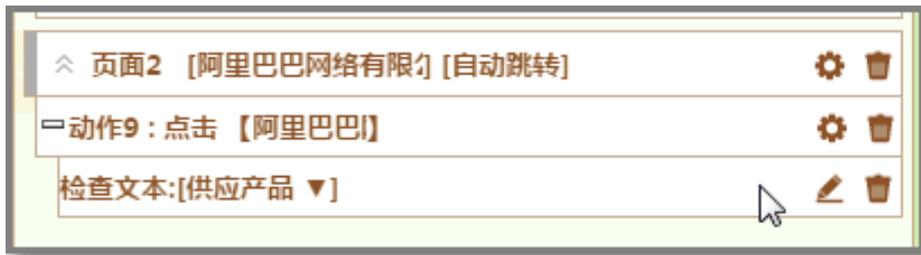
7. 删除检查点

重新获取某页面元素的定位数据，场景：对于执行失败中，应控件定位信息变化找不到控件的情况

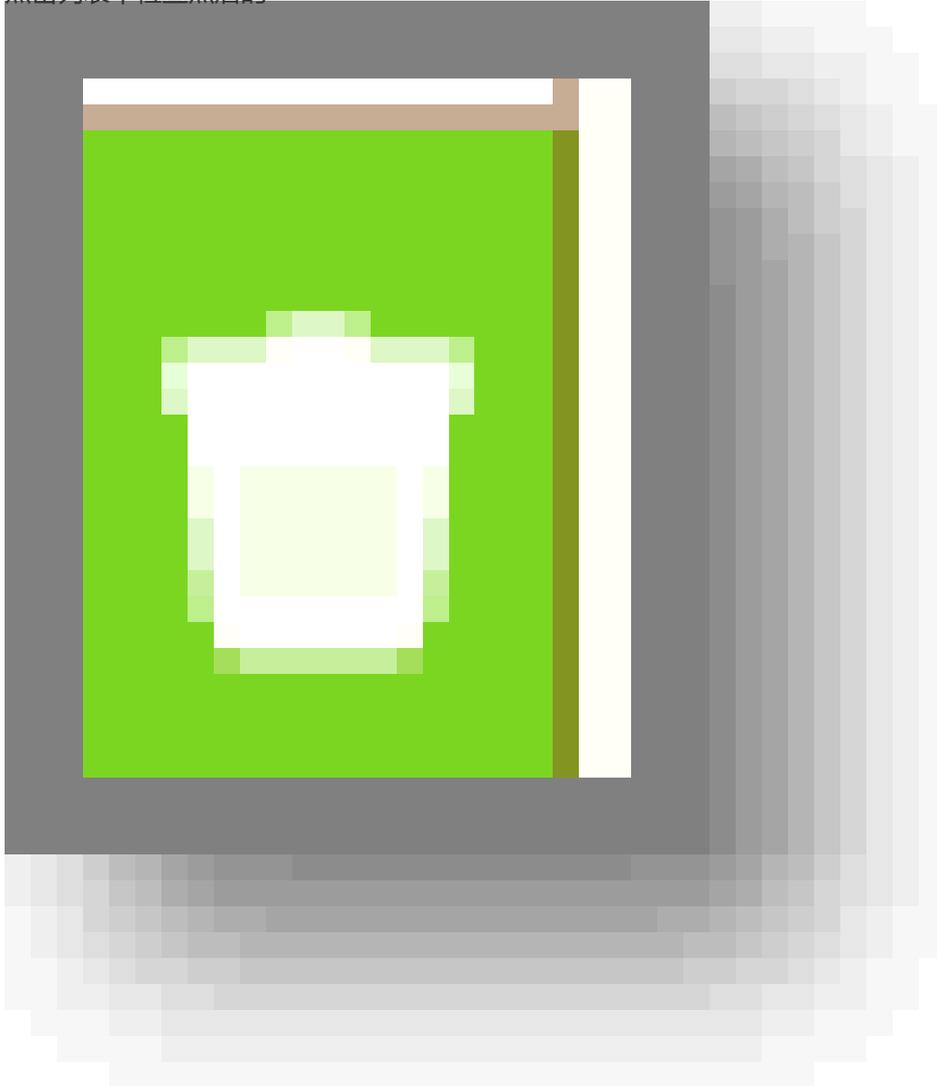
操作如下：

1. 在“已录制动作记录”页面，选中（单击）需要删除检查点的一个步骤。

弹出检查点列表，如下图：



3. 点击列表中检查点后的



来删除检查点。

8. 删除参数

删除一个已有的参数，删除动作和删除检查点类似

操作如下：

1. 在“已录制动作记录”页面，选中（单击）需要删除变量的一个步骤；

弹出变量列表，如下图：



3. 点击列表中变量点后的[X]来删除变量。

公共脚本

引用公共脚本

引用现有的公共脚本，减轻录制工作量与维护成本。

操作如下：

在录制工具框中，点击**脚本引用**按钮，跳出下图对话框：



选择对应的公共脚本，点**确定**，就会在当前步骤所在的页面后，引入公共脚本，并自动触发页面调试，帮助执行。

若遇到错误，会红色高亮注明，如下图，即公共脚本区域。



问题一

- 【描述】操作过程中，一个页面出现 2 次，为什么在维护时，当前步骤总是停在第一次出现的页面上？需要在第二次出现的页面修改，怎么操作？
- 【说明】这个操作主要是解决维护和创建时，针对页面的操作需求不一样才产生的，维护时基本不会新增页面，需要页面信息和当前打开的页面之间有个同步，方便维护操作，录制时，才会产生同一个页面不断加入到录制流程中的情况，那时，就不会有当前打开页面和页面信息之间的自动同步功能了，只有在刷新时才会识别。所以在录制页面上有状态切换功能，见（操作入门 - 基本功能 - 录制状态的说明）
- 【解决】双击第二次出现的页面信息

问题二

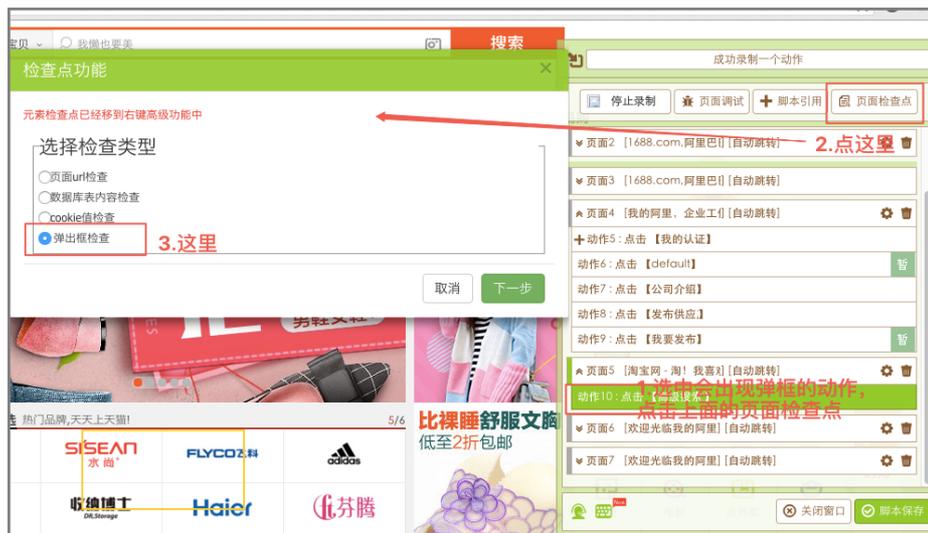
- 【描述】为什么有些脚本不能切换成公共脚本？
- 【说明】公共脚本切换成普通脚本的前提是该脚本没有被引用。普通脚本切换成公共脚本的前提：
 - 该脚本没有引用其它脚本；
 - 设置公共脚本的参数和权限。

问题三

- 【描述】支持图片验证码么？
- 【说明】目前还不支持；建议添加一些自动化测试账号，这些账号不会出现图片验证，绕开这个问题。

问题四

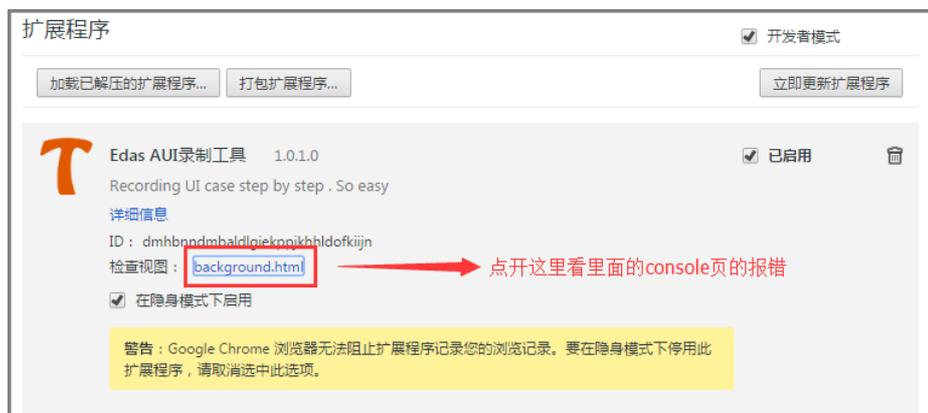
- 【描述】支持页面弹框检查么？
- 【说明】支持，如下图：



问题五

- 【描述】怎么检查问题？

【说明】在 Chrome 浏览器，扩展程序中选中“开发者模式”；点击【检查视图】后的“background.html”。如下图：



问题六

- 【描述】某个页面有多个商品，每个商品的途经地都不同，怎么查看每一个商品的途经地都包含自己指定的几个地点之一？



【说明】如下：

可以将下图中的 `${addr}` 直接改成指定的地点，如“长崎”。



也可以使用参数设置，如下图：



问题七



如上图就是使用错误。页面动态参数基于页面有前后逻辑，在设置后的页面都会显示，设置前的页面是看不到的。

问题八

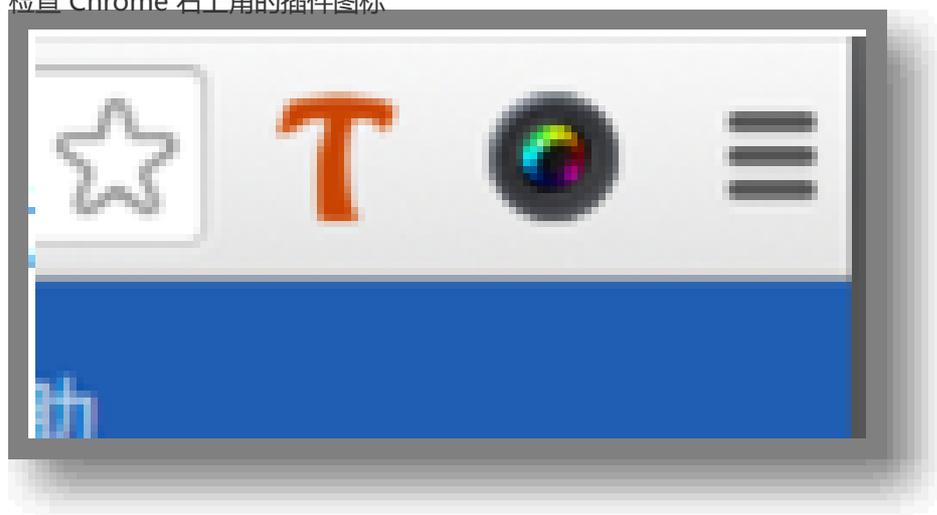
【描述】脚本内容出不来，对话框选项重复。如下图:





【解决方案】如下：

- i. 查看是不是开启了两种 AUI（目前有三种：云效，itest，PaaS 内部版）。
- ii. 检查 Chrome 右上角的插件图标



有两个或

两个重复的就有问题。

- iii. 打开 `chrome://extensions/`，只启用对应的版本即可（依据使用的用例管理系统）。

问题九

【描述】下图的动作二中的密码为什么不能设置参数？



【说明】密码可以设置参数，只是设置后密码框中的密码是已“*”代替，是页面控件自动的行为。



分布式任务调度 SchedulerX

SchedulerX 简介

SchedulerX 是阿里中间件团队开发的一款分布式任务调度产品，在阿里内部有着广泛的使用，经过集团内上千个业务应用历经多年打磨而成。每天非常稳定的运行着集团内几十万个任务以及完成每天几亿次的任务调度。在未来 SchedulerX 将支持更多特性以满足集团内外不断增长的用户需求。

用户在应用中依赖 SchedulerX-Client，并在 SchedulerX 控制台创建定时任务，进行相应的参数配置后，启动该应用就可以接收到定时任务的周期调度。SchedulerX-Server 集群为调度触发提供高可用性和高稳定性的保证，并且可以实现对用户客户端机器集群进行分布式调度。

功能特性

分布式

不管是 SchedulerX 服务端还是客户端都是分布式架构设计，任务可以在多台客户端机器里的任何一台机器执行，如果客户端出现宕机的情况，服务端会自动选择正常运行的客户端（业务 App 机器）去执行 Job。

高可用

每个 Job 在服务端三台不同机器有一个备份，SchedulerX 服务端任意宕掉两台机器仍能保证 Job 正常调度。

轻量服务

用户不需要关心调度逻辑，只需要实现业务逻辑即可。

简单易用

用户只需要在业务应用里添加上 SchedulerX 的客户端 JAR 包，实现一个 JobProcessor 接口，然后再自主运维控制台配置上一个 Job 即可完成使用。

支持 Quartz 时间表达式

支持国际标准的 Quartz 时间表达式，该表达式精确到秒级别。

友好的用户界面

SchedulerX 提供了非常友好的自主运维控制台方便用户创建，删除，修改 Job。提供了立即触发执行一次的功能，方便用户测试以及关键时刻手动立即执行一次。还为用户提供了历史执行记录查询的功能，用户可以看到任何一个 Job 过去 100 次的历史执行记录。

应用场景

固定时间点触发的任务

例如：2016年11月11日0点执行的一次任务。

周期性触发的任务

例如：每秒钟（或者每小时、每天、每星期、每月等）执行一次的任务。

通过控制台手动触发的任务

例如：可以通过控制台手动触发任务的调度执行。任务触发执行后，由用户实现的 Job 处理器接口中的代码决定具体要完成的业务逻辑功能（例如扫表、触发 RPC 调用、入库、执行本地脚本等）。

本文介绍了利用 SchedulerX 完成几个简单 Job 的示例。

每秒输出 Hello world！（时间表达式：0/1？）

```
public class HelloWorldJobProcessor implements ScxSimpleJobProcessor {
    public ProcessResult process(ScxSimpleJobContext context) {
        System.out.println("Hello World!");
        return new ProcessResult(true);//true 表示执行成功，false 表示失败
    }
}
```

每天凌晨 1 点执行一个 Shell 脚本删除日志文件。（时间表达式：0 0 1 ? ?）

```
public class ExecuteShellJobProcessor implements ScxSimpleJobProcessor {
    public ProcessResult process(ScxSimpleJobContext context) {
        try {
            Runtime.getRuntime().exec("rm -rf /home/admin/logs/*.log");
        } catch (IOException e) {
            e.printStackTrace();
        }
        return new ProcessResult(true);//true 表示执行成功，false 表示失败
    }
}
```

每年一次的生日提醒。（时间表达式：0 0 0 11 11 ? *）

```
public class BirthdayJobProcessor implements ScxSimpleJobProcessor {
    public ProcessResult process(ScxSimpleJobContext context) {
        // 此处省略生日提醒业务逻辑
        return new ProcessResult(true);//true 表示执行成功，false 表示失败
    }
}
```

本文主要介绍如何快速上手使用 SchedulerX，包括如何创建和配置一个定时任务、定时任务处理器的编写以及定时任务执行客户端 Bean 配置。

主要内容如下：

创建定时任务处理器。

创建定时任务并配置任务触发时间周期。

运行客户端，启动调度任务。]

下文以建立一个每隔 1 分钟执行一次 Hello World 输出的定时任务为例进行说明。

1、创建定时任务处理器

在工程中添加 SchedulerX -client 依赖。

```
<dependency>
<groupId>com.alibaba.edas</groupId>
<artifactId>schedulerX-client</artifactId>
<version>2.1.5</version>
</dependency>
```

实现 Job 处理器接口。

```
package com.schedulerx.test;
import java.util.Date;
import com.alibaba.edas.schedulerx.ProcessResult;
import com.alibaba.edas.schedulerx.ScxSimpleJobContext;
import com.alibaba.edas.schedulerx.ScxSimpleJobProcessor;
public class ExecuteShellJobProcessor implements ScxSimpleJobProcessor {
    public ProcessResult process(ScxSimpleJobContext context) {
        System.out.println("Hello World! " + new Date());
        return new ProcessResult(true);//true 表示执行成功，false 表示失败
    }
}
```

2、创建定时任务并配置任务触发时间周期

登录 EDAS 控制台, 在左侧菜单栏选择 **任务管理** > **分组管理**，在右上角单击 **新建分组**。

在 **新建分组** 对话框输入 **分组描述**，单击 **确定**。

在左侧菜单栏选择 **任务管理** > **任务列表**，在右上角单击 **新建 Job**。

在 **创建任务** 对话框输入相应信息，单击 **确定**。

新建Job

*Job 分组：华东1

*Job 处理接口：请输入Job处理接口实现类的全类名（完整包路径加类名，如XXX）

*类型：简单Job单机版

*定时表达式：0 0 0 * * ? 使用工具生成

Job 还在执行中，到了下次触发时间点Job照常触发执行

Job 描述：请输入Job描述

确定 取消

Job 分组：在右侧选择 Job 所在的区域及分组。

在选择任务分组时可选择已经存在的分组，也可使用新创建的任务分组。

Job 处理接口：输入接口实现的全类名（完整包路径加类名）。

此处填入第 2 步创建的 Job 处理器 ExecuteShellJobProcessor 的全类名（com.schedulerx.test.ExecuteShellJobProcessor）。

类型：类型包括 简单 Job 单机版 和 简单 Job 多机版。

定时表达式：可使用工具生成，也可手工修改时间表达式字符串。

工具生成：



手工修改时间表达式字符串：



此处时间表达式兼容 Quartz 时间表达式，Quartz 提供七个域（分别对应：秒>分>小时>日>月>周>年），通常年可以省略。

详细时间表达式说明请参考 EDAS SchedulerX 使用指南-创建 JOB。

3、运行客户端，启动调度任务

初始化 SchedulerX Client。

独立 Java 应用方式

```
package com.schedulerx.test;

import com.alibaba.edas.schedulerx.SchedulerXClient;

public class schedulerxTestMain {

    public static void main(String[] args) {
        SchedulerXClient schedulerXClient = new SchedulerXClient();
        schedulerXClient.setGroupId("101-1-1-52");
        schedulerXClient.setRegionName("cn-hangzhou");
        /*
        //如果使用的是上海 Region 集群，需要设置 domainName 属性，同时指定 RegionName 为 cn-shanghai
        schedulerXClient.setRegionName("cn-shanghai");
        schedulerXClient.setDomainName("schedulerx-shanghai.console.aliyun.com");
        */
        try {
            schedulerXClient.init();
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

```
}
```

EDAS 应用管理(WAR 包部署)方式

在项目工程中增加 spring-web 的依赖。

```
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-web</artifactId>
<version>4.1.5.RELEASE</version>
</dependency>
```

在 Spring 配置文件中增加 SchedulerXClient bean 的配置。

```
<bean id="schedulerXClient" class="com.alibaba.edas.schedulerx.SchedulerXClient"
init-method="init">
<property name="groupId">
<value>101-1-1-77</value>
</property>
<property name="regionName">
<value>cn-hangzhou</value>
</property>
<!--如果使用的是上海 Region 集群，需要设置 domainName 属性，同时指定
RegionName 为 cn-shanghai
<property name="regionName">
<value>cn-shanghai</value>
</property>
<property name="domainName">
<value>schedulerx-shanghai.console.aliyun.com</value>
</property-->
</bean>
```

在 web.xml 配置 Spring 监听器。

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://java.sun.com/xml/ns/javaee" xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/web-app_2_5.xsd" version="2.5">
<display-name>scxWarTest</display-name>
<context-param>
<param-name>contextConfigLocation</param-name>
<param-value>WEB-INF/applicationContext.xml</param-value>
</context-param>
<listener>
<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
</web-app>
```

注意：以上两种方式，schedulerXClient 配置的属性值必需和您在控制台的配置信息保持一致，如下图所示。



JobID/描述	分组	区域	定时表达式	Job上次触发时间	Job状态	是否启用	操作
...	0 0 0 * * ?	未执行	已启用	详情 编辑 监控配置 触发一次 更多	
...	0 0 0 * * ?	未执行	已启用	详情 编辑 监控配置 触发一次 更多	

- **分组：**对应 schedulerXClient Bean 中的 groupId 属性。
- **区域：**对应 schedulerXClient Bean 中的 regionName 属性。

启动调度任务。

独立 Java 应用方式

在运行应用的机器安装 EDAS Agent。

把应用打好的 JAR 包上传至应用服务器。

运行主函数 schedulerxTestMain，同时在程序启动时加上参数 -Dspas.identity=/home/admin/.spas_key/default

EDAS 应用管理(WAR 包部署)方式

将应用打好 WAR 包。

在 EDAS 控制台应用管理中创建应用，上传 WAR 包。

在应用管理中启动调度任务应用。

启动调度任务后，系统将会根据您的设定（每隔1分钟）调用实现 Job 处理器接口的类代码（ExecuteShellJobProcessor），完成打印输出 Hello World 的任务。

开发者旺旺交流群：1644245735

在 EDAS 控制台左侧菜单栏选择 **任务管理 > 任务列表**，然后在 **Job 列表** 右上角单击 **新建 Job**。

在 **新建 Job** 对话框输入相应信息，单击 **确定**。

新建Job

*Job分组： 8.25 华东1

*Job处理接口： 请输入Job处理接口实现类的全类名（完整包路径加类名，如XXX）

*类型： 简单Job多机版 ? 最小触发时间间隔10秒

*定时表达式： 0 0 0 * * ? 使用工具生成

Job还在执行中，到了下次触发时间点Job照常触发执行

Job描述： 请输入Job描述

确定 取消

任务参数说明：

Job 分组：在右侧选择 Job 所在的区域及分组。

在选择任务分组时可选择已经存在的分组，也可使用新创建的任务分组。

Job 处理接口：输入接口实现的全类名（完整包路径加类名）。

用户填写的全类名，客户端会自动新建出对象，如果是 Spring 环境并且对象有其他类的依赖，客户端会根据成员变量名称在 Spring 环境找到对应的 Spring Bean，然后自动注入相应的依赖对象。

类型：类型包括 简单 Job 单机版 和 简单 Job 多机版。

简单 Job 单机版

在已经启动的连接到该分组的所有机器中，随机选一台机器触发执行。

简单 Job 多机版 (需要 schedulerX-client 1.6.6 以上版本)

同时触发所有已经启动的连接到该分组的机器执行。

定时表达式：可使用工具生成，也可手工修改时间表达式字符串。

使用工具生成

手工修改时间表达式字符串

此处时间表达式兼容 Quartz 时间表式，Quartz 提供七个域（分别对应：秒>分>小时>日>月>周>年），通常年可以省略。

时间表达式的时间字段除允许设置数值外，还可使用一些特殊的字符，提供列表、范围、通配符等功能。

星号 (*)：可用在所有字段中，表示对应时间域的每一个时刻，例如，星号在分钟字段时，表示“每分钟”。

问号 (?)：该字符只在日期和星期字段中使用，它通常指定为“无意义的值”，相当于点位符。

减号 (-)：表达一个范围，如在小时字段中使用“10-12”，则表示从10到12点，即10，11，12。

逗号 (,)：表达一个列表值，如在星期字段中使用“MON, WED, FRI”，则表示星期一，星期三和星期五。

斜杠 (/)：x/y 表达一个等步长序列，x 为起始值，y 为增量步长值。如在分钟字段中使用0/15，则表示为0，15，30和45秒，而5/15在分钟字段中表示5，20，35，50，你也可以使用*/y，它等同于0/y。

Quartz 时间表达式的格式与 Linux Crontab 格式的区别：

Quartz 的格式向下支持到秒级别的计划，Linux Crontab 是到分钟级。

Linux Crontab 给出五个域(分、时、日、月和周)，Quartz 提供七个域。

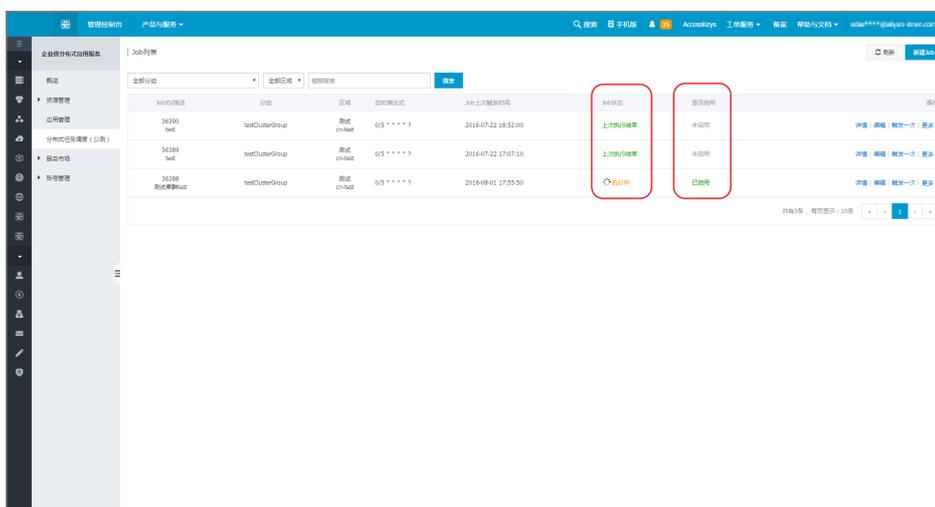
在 Job 列表页面，您可以查看 Job 的运行状态及配置信息，并对某个 Job 的配置信息进行编辑和操作。

查看 Job 运行状态

在 EDAS 控制台左侧菜单栏选择 **任务管理 > 任务列表**。

在 Job 列表页面查看 Job 运行状态。

如果列表太长，也可以根据服务分组、所在区域进行筛选，进行关键字模糊搜索。



Job 列表除了 Job 的一些基本信息外，主要是如下两个状态信息：

Job 状态:

- 执行中：代表本次 Job 触发正在执行过程中（在用户实现的代码中运行）。
- 上次执行结束：上次触发运行已经结束，等待下一个触发点。
- 未执行：Job 创建完成后，还没有被调度执行过。

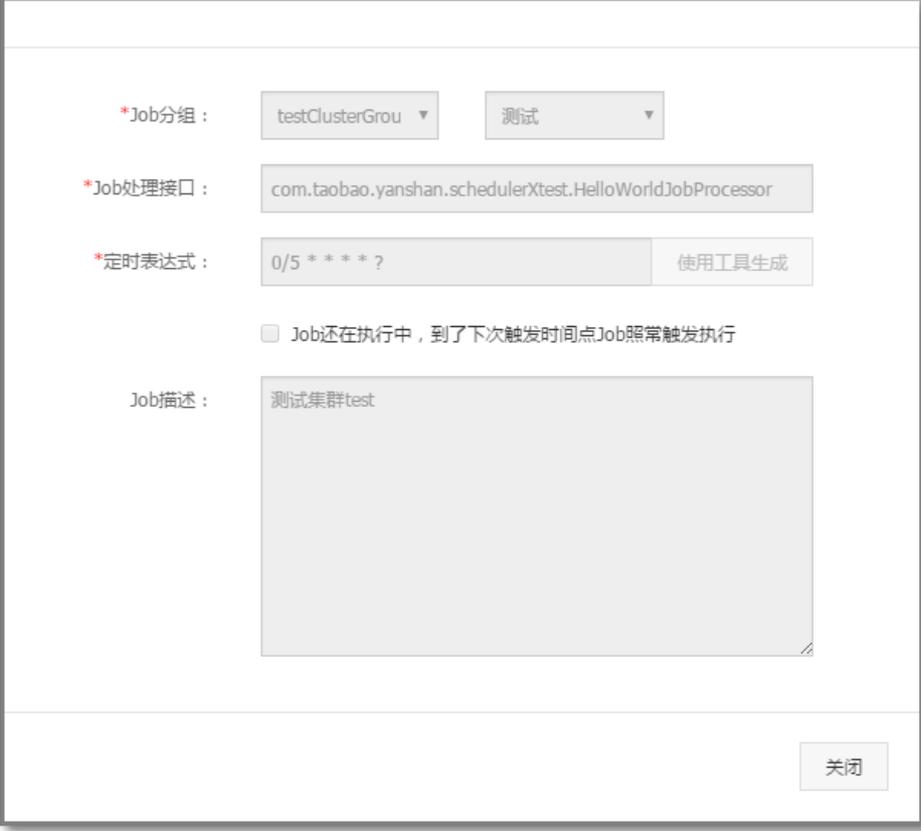
是否启用：

- 已启用：Job 为正常启用状态，到达用户设定的触发时间点后，系统会触发执行该 Job。
- 未启用：Job 为停用状态，系统不会对该 Job 进行触发执行。

查看与编辑 Job 配置

查看 Job 详细配置信息

在 Job 右侧操作选项单击 **详情**，即可查看配置信息。



The screenshot shows a configuration dialog box for a job. It contains the following fields and controls:

- *Job分组:** A dropdown menu with 'testClusterGrou' selected and another dropdown with '测试'.
- *Job处理接口:** A text input field containing 'com.taobao.yanshan.schedulerXtest.HelloWorldJobProcessor'.
- *定时表达式:** A text input field containing '0/5 * * * * ?' and a button labeled '使用工具生成'.
- Job还在执行中，到了下次触发时间点Job照常触发执行
- Job描述:** A text area containing '测试集群test'.
- 关闭** button at the bottom right.

修改 Job 配置信息

单击 Job 右侧的 **编辑** 选项，即可编辑 Job 的配置信息。

编辑Job

*Job分组：

*Job处理接口：

*定时表达式：

Job还在执行中，到了下次触发时间点Job照常触发执行

Job描述：

触发 Job

单击 Job 右侧的操作选项里的 **触发一次**，会立即确触发 Job 运行1次，不管 Job 此时是否为启用状态，此操作通常可用于 Job 调试。

启用、停用、删除 Job

在 Job 操作选项单击 **更多**，根据 Job 状态会有 **启用、停用、删除、触发记录** 选项。

监控报警配置功能，是用户可以针对自己的 Job 任务进行监控报警配置，并指定报警信息接收人。在 Job 运行达到报警条件时，系统会将报警信息通知给配置好的联系人，提示用户对 Job 任务进行检查维护。

在 Job 列表中，选择需要监控的 Job，单击右侧的 **监控配置**。

在监控配置页面输入相关信息，单击 **确定**。

监控配置

*使用报警：

*当满足下述任一条件时，发出报警：

Job连续超过 次没有准时触发 1

正在执行的Job，执行时间超过 秒 超过后强制中止Job 2

Job执行结束后有超过 %的机器执行失败发出报警 3

*报警通知对象：

全部联系人

请输入关键词查询联系人

- edas.guyi
- edas.yinshi
- zhendongtest@aliyun-inner.com
- edas.xuanye

已选联系人

监控参数说明：

- **使用报警**：可以打开、关闭报警。关闭报警后配置参数会保留，但报警信息不会发送。
- **未准时触发报警**：设置此项报警后，如果 Job 连续未准时触发的次数达到用户设置的阈值时，将触发报警。
- **执行超时报警**：设置此项报警后，如果用户的 Job 执行的时间超过用户设置的阈值时，将触发报警。同时如果选择了后面的**超过后强制中止 Job**，系统会将会在发送报警的同时终止 Job，将 Job 运行状态修改为结束运行，但不会强制终止用户 client 中运行的线程。同一 Job 实例首次报警后，后续将以5分钟、10分钟、20分钟、40分钟的时间间隔发送报警信息。**注意**：此项报警是应用在运行时间较长的 Job 监控上，所以应用此报警配置的 Job 运行时间需在 1 分钟以上。
- **执行失败报警**：单机版 Job 此处为复选框，选择即可；多机版 Job 此处需要输入失败的百分比（即运行失败的机器数/总机器数*100）。当 Job 运行失败且达到此阈值时，将触发报警。
- **设置报警通知对象**：Job 报警信息分为两种形式发送，短信方式和电子邮件方式。报警通知对象可以选择多个联系人，联系人的电话和邮箱信息在帐号管理中的个人资料处进行设置和维护。

本文档介绍如何在开发测试区域（Region）新建调度任务，并对任务进行试运行测试。

注意：开发测试区域（Region）是为方便用户在开发环境中进行任务调度测试，并不提供稳定性和安全性保证。请在测试结束后，删除 Job 配置并将 Job 迁移到正式区域中。

创建 Job

登录 EDAS 控制台，在左侧导航栏中选择 **任务管理 > 分组管理**。

在任务分组页面中选择 **区域**，再选择某个 **分组**。

或者点击 **新建分组**，创建一个新的分组。

注意：

- 如果用于测试，通过开发环境（没有 ECS）连接 SchedulerX 时，需要选择 **测试** 区域。
- 如果用于正式使用时，则只能通过该区域内的 ECS 连接。

在任务管理页面右上角单击 **新建 Job**。

注意：要在某个区域下新建 job，则该区域下必须存在分组。

此处，在 schedulerxTestGroup 分组的测试区域中建立了一个每30秒触发一次，Job 处理器实现类为 com.schedulerx.test.ExecuteShellJobProcessor 的 Job，具体参数解释可参考 **创建 Job**。

在工程中编写代码

在工程文件 pom.xml 中增加下面依赖：

```
<dependencies>
<dependency>
<groupId>com.alibaba.edas</groupId>
<artifactId>schedulerX-client</artifactId>
<version>1.6.6</version>
</dependency>
</dependencies>
```

初始化 SchedulerX-Client。

第一种方法：

```
public class SchedulerxTestMain {
```

```
private static final Log logger = LoggerFactory.getLog(SchedulerxTestMain.class);

public static void main(String[] args) {

    SchedulerXClient schedulerXClient = new SchedulerXClient();
    schedulerXClient.setGroupId("101-1-1-77");
    schedulerXClient.setRegionName("cn-test");

    try {
        schedulerXClient.init();
    } catch (Throwable e) {
        e.printStackTrace();
    }
}

}
```

第二种方法 (通过 Spring 方式) :

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE beans PUBLIC "-//SPRING//DTD BEAN 2.0//EN"
"http://www.springframework.org/dtd/spring-beans-2.0.dtd">
<beans>

<bean id="schedulerXClient" class="com.alibaba.edas.schedulerX.SchedulerXClient" init-
method="init">
<property name="groupId">
<value>101-1-1-77</value>
</property>
<property name="regionName">
<value>cn-test</value>
</property>

</bean>

</beans>
```

上面两种方式对应的 2 个参数分别为分组和区域。

编写 Job 处理器接口。

注意：Job 处理器要实现接口 ScxSimpleJobProcessor，并且要与在页面创建 Job 时填写的 Job 处理器接口名称对应上。

```
package com.schedulerx.test;

import java.util.Date ;

public class ExecuteShellJobProcessor implements ScxSimpleJobProcessor {
    public ProcessResult process(ScxSimpleJobContext context) {
        System.out.println("Hello World") +new Date();
    }
}
```

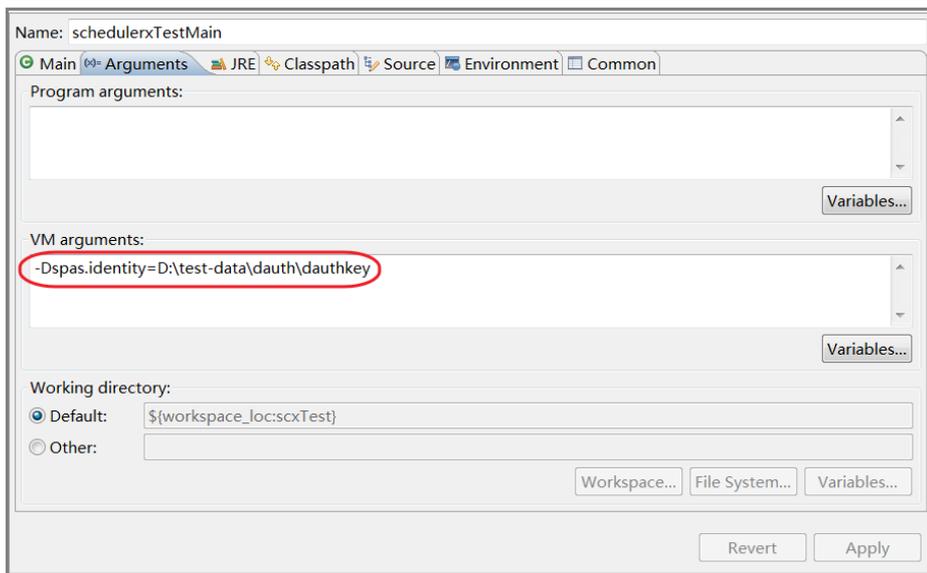
```
return new ProcessResult(true);//true 表示执行成功，false 表示失败
}
}
```

The screenshot shows a configuration window for a job. It contains the following elements:

- *Job分组:** A dropdown menu with 'schedulerxTest' selected and a '测试' button.
- *Job处理接口:** A text input field containing 'com.schedulerx.test.ExecuteShellJobProcessor', which is circled in red.
- *定时表达式:** A text input field with '0/30 * * * * ?' and a '使用工具生成' button.
- Job还在执行中，到了下次触发时间点Job照常触发执行
- Job描述:** A text area containing 'schedulerX'.
- 关闭** button at the bottom right.

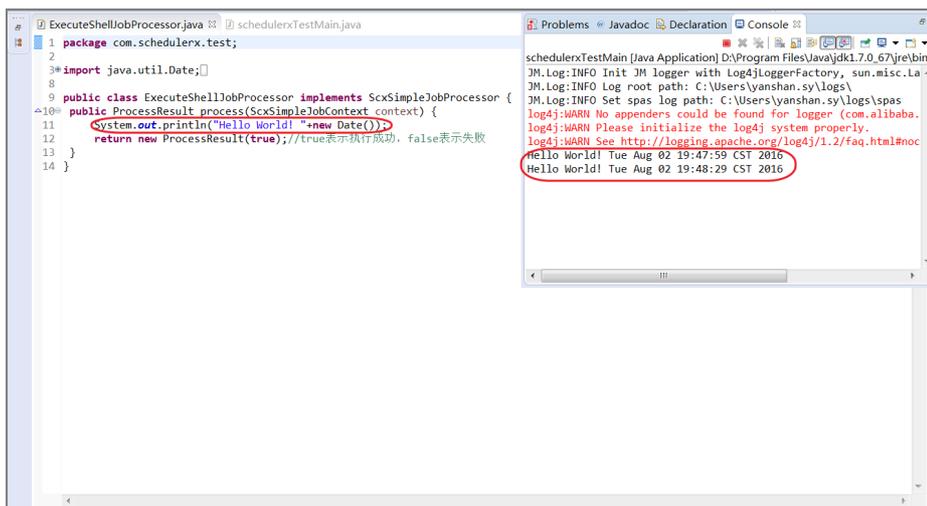
运行测试

在运行测试程序前需要设置一下运行参数：



文件路径和文件名根据实际情况填写，开发测试区域不进行鉴权。

设置好参数后运行程序，每 5 秒钟会接收到一次 Job 触发，输出处理器逻辑中打印的信息。



当前客户端的最新版本为 2.1.5。新增了如下特性：

- 支持网格任务
- 支持脚本任务

升级到最新版本的客户端，在工程中添加对应的版本依赖即可。

```
<dependency>
<groupId>com.alibaba.edas</groupId>
<artifactId>schedulerX-client</artifactId>
<version>2.1.5</version>
</dependency>
```

注意：

1.6.6 及以下版本客户端升级时，com.alibaba.edas.schedulerX 包路径变更为 com.alibaba.edas.schedulerx。SchedulerXClient 等类需要重新导入。

2.1.3 以上版本要求运行环境JDK1.8版本

本文通过示例介绍如何使用 SchedulerX 的脚本任务。

注意：使用脚本任务需要 schedulerx-client-2.1.5 及以上版本。

创建脚本任务

1. 创建 shell 脚本任务

在 EDAS 控制台左侧导航栏选择 **任务管理** > **任务列表**，在 **Job 列表** 页面右上角单击 **新建 Job**。

在 **任务类型** 中选择 **脚本任务**，然后在 **脚本内容** 中录入 shell 脚本。

```
echo "Hello,"  
echo "EDAS"
```

任务执行后，会在执行日志中输出 “Hello , EDAS” 信息。

2. 创建带参数的 shell 任务

编辑一个 shell 脚本，保存为 /Users/xxx/shellJobTest/hello.sh。

```
echo $1  
echo $2
```

编辑一个 shell 任务，然后在 **脚本内容** 中录入 shell 脚本。

```
/Users/xxx/shellJobTest/hello.sh Hello , EDAS
```

任务执行后，会在执行日志中输出 “Hello , EDAS” 信息。

3. 创建 Python 脚本任务

编辑一个 Python 脚本，保存为 /Users/xxx/shellJobTest/hello.py。

```
print('Hello edas, python!')
val1 = 1
val2 = 0
val3 = val1/val2
print('val3 is ' + val3)
```

编辑一个 shell 任务，然后在 **脚本内容** 中录入 shell 脚本。

```
python /Users/xxx/shellJobTest/hello.py
```

说明：这里故意写了一个错误的逻辑，1/0会输出异常，并记录在异常日志中。

4. 任务执行结果查看

在 **Job 列表** 页面对应的 Job 记录右侧选择 **更多 > 触发记录**。可以查看到任务的执行详情、执行日志和执行结果。

运行脚本任务

运行脚本任务有两种方式：**依赖 Client 的方式** 和 **Agent 方式**。

依赖 Client 的方式

在应用工程中添加 SchedulerX-client 依赖。

```
<dependency>
<groupId>com.alibaba.edas</groupId>
<artifactId>schedulerX-client</artifactId>
<version>2.1.5</version>
</dependency>
```

初始化 SchedulerX Client。

初始化 SchedulerX Client 有两种方式：**独立 Java 应用方式** 和 **Spring 应用方式**。

独立 Java 应用方式

```
package com.schedulerx.test;
import com.alibaba.edas.schedulerx.SchedulerXClient;

public class schedulerxTestMain {

    public static void main(String[] args) {
        SchedulerXClient schedulerXClient = new SchedulerXClient();
        schedulerXClient.setGroupId("111-1-1-11");
        schedulerXClient.setRegionName("cn-hangzhou");
        schedulerXClient.setNewVersion(true);
        /*
        //如果使用的是上海 Region 内的集群，需要设置 domainName 属性，同时指定 RegionName 为 cn-
        shanghai
        schedulerXClient.setRegionName("cn-shanghai");
        schedulerXClient.setDomainName("schedulerx-shanghai.console.aliyun.com");
        */

        try {
            schedulerXClient.init();
        } catch (Throwable e) {
            e.printStackTrace();
        }
    }
}
```

Spring 应用方式

```
<bean id="schedulerXClient" class="com.alibaba.edas.schedulerx.SchedulerXClient" init-
method="init">
<property name="groupId">
<value>101-1-1-77</value>
</property>
<property name="regionName">
<value>cn-hangzhou</value>
</property>
<!--如果使用的是上海 Region 内的集群，需要设置 domainName 属性，同时指定 RegionName 为 cn-
shanghai
<property name="regionName">
<value>cn-shanghai</value>
</property>
<property name="domainName">
<value>schedulerx-shanghai.console.aliyun.com</value>
</property>-->
<property name="newVersion">
<value>true</value>
</property>
</bean>
```

Agent 方式

下载 SchedulerX-Agent。

可在应用机器上通过 **wget** 命令下载 SchedulerX Agent。

SchedulerX Agent 下载地址：<http://edas.oss-cn-hangzhou.aliyuncs.com/SchedulerX/schedulerx-client-2.1.4.tar.gz>

运行 SchedulerX-Agent。

解压下载的压缩包，进入 SchedulerX-Agent/conf 目录，编辑 agent.ini 文件。

```
[baseSection]
groupId=111-1-1-0001 //任务分组编号
domainName=schedulerx.console.aliyun.com //域名，如果是上海 Region 请填写 schedulerx-shanghai.console.aliyun.com
regionName=cn-test //填写对应的 regionName
newVersion=true //缺省参数，不用修改
```

进入 SchedulerX-Agent/bin 目录，执行 **start-1g.sh**。

根据任务负载及机器配置情况可对应使用 **start-2g.sh**、**start-4g.sh** 或 **start-8g.sh**。

Agent 提示启动成功，就可以接收定时任务调度了。