

基于 Python+Django+Easyui 实现的接口自动化测试框架

by: 授客 QQ: 1033553122

博客: <https://www.cnblogs.com/shouke/>

欢迎加入全国软件测试交流 QQ 群: 7156436

目录

1、 开发环境	3
2、 框架功能简介	8
3、 服务端部署	9
a) Apache 配置	9
检查 Apache 版本是否正确	11
修改 Apache 配置	12
启动 Apache	13
b) Django 配置	14
Django 访问 IP 配置	14
创建库配置	14
c) APIRunner 配置	15
数据库配置	15
https SLL、TSL 版本配置	15
d) 系统环境变量配置	15
e) 数据库的创建	16
f) 创建表	16
执行初始化 sql	16
g) 重启 Apache 并启动 Django 应用	16
h) 启动 daphne	17
i) 修改 apache 配置	18
j) 开启 daphne 监控	18
4、 客户端部署	18
a) 数据库配置	18
b) https SLL、TSL 版本配置	19
c) 服务器端日志配置	19
d) 客户端日志配置	19
5、 大致操作流程	20
a) 环境配置	20
b) 新增项目	20
c) 系统配置	20
d) 新增用例	20
e) 新增 API 计划	20
f) 关联 API 测试用例	20

g) 新增并执行“运行计划”	20
h) 查看测试报告	21
6、测试任务可视化管理功能介绍	21
7、用例步骤填写规范	22
a) 接口请求	22
1. 执行操作	22
2. 请求头	23
3. URL/SQL	23
4. 输入参数	24
5. 检查响应	25
6. 校验规则及校验模式	25
7. 输出	25
8. 协议	25
9. 主机地址	25
10. 端口	25
11. 运行次数	25
12. 失败重试次数	25
13. 重试频率	25
b) 操作数据库	25
① 执行操作	26
② URL/SQL	26
③ 输入参数	26
④ 校验规则及校验模式	26
⑤ 输出	26
c) 操作 Redis	27
① 执行操作	27
② 输入参数	27
③ 校验规则及校验模式	27
④ 输出	27
d) 执行函数	27
① 操作对象	27
② 输入参数	27
③ 校验规则及校验模式	27
④ 输出	27
e) 关于用例及步骤运行顺序说明	27
8、参数化	27
① 全局变量	27
② 局部变量	28
③ 变量的引用	28
④ 变量使用范围	28
⑤ 插件函数	31
9、运行与调试	32
a) 服务器在线(调试)运行	32
① 项目级别的(调试)运行	32

② 测试计划级别的调试.....	33
③ 用例(套件)级别的调试.....	33
b) 客户端本地调试.....	34
① 项目级别的调试.....	34
② 用例(套件)级别的调试.....	35
10、 数据备份	35

1、 开发环境

服务端:

win7 64\Windows Server 2008 R2 x64

python-3.5.4-amd64.exe

数据库

MariaDB 5.5.45, MySQL Server 5.7

Django-1.11.4.tar.gz

下载地址:

<https://pan.baidu.com/s/1hsc1V5y>

官方下载地址:

<https://www.djangoproject.com/download/>

<https://github.com/django/django/releases>

<https://codeload.github.com/django/django/tar.gz/1.11.4>

pytz-2018.9.tar.gz

(Django 依赖包, 因公司网络限制, 安装 Django 时自动下载该软件包时超时, 如果安装过程中没报错, 可忽略下载该软件包, 下同, 不再赘述)

<https://files.pythonhosted.org/packages/af/be/6c59e30e208a5f28da85751b93ec7b97e4612268bb054d0dff396e758a90/pytz-2018.9.tar.gz>

Easyui 1.5.3

下载地址:

<https://pan.baidu.com/s/1i6E597R>

下载地址:

<http://www.jeasyui.com/download/v153.php>

Jquery-3.2.1.min.js

下载地址:

<https://pan.baidu.com/s/1bqP7jeZ>

下载地址:

<http://www.jq22.com/jquery/jquery-3.2.1.zip>

jquery-easyui-datagrid-dnd

下载地址:

<http://www.jeasyui.net/demo/193.html>

下载地址:

<https://pan.baidu.com/s/1EtiFA1W3-u3T5USCLNE0eQ>

channels-2.1.7.tar.gz

下载地址:

<https://pypi.org/project/channels/#files>

<https://files.pythonhosted.org/packages/b9/a8/d4fef151a93b7c2f3ae66553db5a9d8cec41f7743e35a70f369b4c5a5800/channels-2.1.7.tar.gz>

daphne-2.2.5.tar.gz

(channels 依赖包)

<https://files.pythonhosted.org/packages/ba/dd/7ac857e4eed08ee6d3c569d8aab275aa8ee5dfe49dcd0351eb9df123fb82/daphne-2.2.5.tar.gz>

pytest-runner-4.4.tar.gz

(daphne 依赖包, 安装 daphne 时提示找不到该软件包)

<https://files.pythonhosted.org/packages/15/0a/1e73c3a3d3f4f5faf5eacac4e55675c1627b15d84265b80b8fef3f8a3fb5/pytest-runner-4.4.tar.gz>

autobahn-19.3.3.tar.gz

(daphne 依赖包, 安装 daphne 时提示找不到该软件包)

<https://files.pythonhosted.org/packages/46/8b/c329cf7a0f591805d8a9fe4b6e88467b8c6e09022b9dfb2242577b983898/autobahn-19.3.3.tar.gz>

txaio-18.8.1.tar.gz

(autobahn 依赖包, 安装 autobahn 时提示找不到该软件包)

<https://files.pythonhosted.org/packages/c1/99/81de004578e9afe017bb1d4c8968088a33621c05449fe330bdd7016d5377/txaio-18.8.1.tar.gz>

six-1.12.0.tar.gz

(txaio 依赖包, 安装 txaio 时提示找不到该软件包)

<https://files.pythonhosted.org/packages/dd/bf/4138e7bfb757de47d1f4b6994648ec67a51efe58fa907c1e11e350cddfca/six-1.12.0.tar.gz>

setuptools_scm-3.2.0.tar.gz

(pytest-runner 依赖包, 安装 channels 时提示找不到该软件包)

https://files.pythonhosted.org/packages/54/85/514ba3ca2a022bddd68819f187ae826986051d130ec5b972076e4f58a9f3/setuptools_scm-3.2.0.tar.gz

asgiref-2.3.0.tar.gz

(channels 依赖包)

<https://files.pythonhosted.org/packages/a9/17/52c6eaa07a4babd1c0260d42f659266313ac90be88f1f8805b93f5ec072d/asgiref-2.3.0.tar.gz>

async-timeout-2.0.0.tar.gz

(asgiref 依赖包, 安装 asgiref 时提示找不到该软件包)

<https://files.pythonhosted.org/packages/78/10/7fd2551dc51f6065bdbba07d395865df4582cc18169297e7a5c8d90f5bd2/async-timeout-2.0.0.tar.gz>

Twisted-18.9.0.tar.bz2

(channels 依赖包)

<https://files.pythonhosted.org/packages/5d/0e/a72d85a55761c2c3ff1cb968143a2fd5f360220779ed90e0fadf4106d4f2/Twisted-18.9.0.tar.bz2>

incremental-17.5.0.tar.gz

(Twisted 依赖包)

<https://files.pythonhosted.org/packages/8f/26/02c4016aa95f45479eea37c90c34f8fab6775732ae62587a874b619ca097/incremental-17.5.0.tar.gz>

pywin32-224-cp35-cp35m-win_amd64.whl

(Twisted 依赖包, win7 下运行, 如果缺少该软件包, 会报错: No module named 'win32api')

https://files.pythonhosted.org/packages/83/a5/960c5a714b3c975102031286121db06fe861fdd221b493dce5144d759b90/pywin32-224-cp35-cp35m-win_amd64.whl

attrs-19.1.0.tar.gz

(incremental 依赖包)

<https://files.pythonhosted.org/packages/cc/d9/931a24cc5394f19383fbb3e1147a0291276afa43a0dc3ed0d6cd9fda813/attrs-19.1.0.tar.gz>

PyHamcrest-1.9.0.tar.gz

(attrs 依赖包)

<https://files.pythonhosted.org/packages/a4/89/a469aad9256aedfbb47a29ec2b2eeb855d9f24a7a4c2ff28bd8d1042ef02/PyHamcrest-1.9.0.tar.gz>

hyperlink-18.0.0.tar.gz

(channels 依赖包)

<https://files.pythonhosted.org/packages/41/e1/0abd4b480ec04892b1db714560f8c855d43df81895c98506442babf3652f/hyperlink-18.0.0.tar.gz>

idna-2.8.tar.gz

(hyperlink 依赖包)

<https://files.pythonhosted.org/packages/ad/13/eb56951b6f7950cadb579ca166e448ba77f9d24efc03edd7e55fa57d04b7/idna-2.8.tar.gz>

Automat-0.7.0.tar.gz

(channels 依赖包)

<https://files.pythonhosted.org/packages/4a/4f/64db3ffda8828cb0541fe949354615f39d02f596b4c33fb74863756fc565/Automat-0.7.0.tar.gz>

m2r-0.2.1.tar.gz

(Automat 依赖包)

<https://files.pythonhosted.org/packages/39/e7/9fae11a45f5e1a3a21d8a98d02948e597c4afd7848a0dbe1a1ebd235f13e/m2r-0.2.1.tar.gz>

docutils-0.14.tar.gz

(m2r 依赖包)

<https://files.pythonhosted.org/packages/84/f4/5771e41fdf52aabebbadec9381d11dea0fa34e4759b4071244fa094804c/docutils-0.14.tar.gz>

mistune-0.8.4.tar.gz

(m2r 依赖包)

<https://files.pythonhosted.org/packages/2d/a4/509f6e7783ddd35482feda27bc7f72e65b5e7dc910eca4ab2164daf9c577/mistune-0.8.4.tar.gz>

constantly-15.1.0.tar.gz

(channels 依赖包)

<https://files.pythonhosted.org/packages/95/f1/207a0a478c4bb34b1b49d5915e2db574cad415c9ac3a7ef17e29b2e8951/constantly-15.1.0.tar.gz>

zope.interface-4.6.0.tar.gz

(channels 依赖包)

<https://files.pythonhosted.org/packages/4e/d0/c9d16bd5b38de44a20c6dc5d5ed80a49626fafcb3db9f9efdc2a19026db6/zope.interface-4.6.0.tar.gz>

mysqlclient-1.3.10-cp35-cp35m-win_amd64.whl

下载地址:

https://raw.githubusercontent.com/OskarPersson/testci/master/mysqlclient-1.3.10-cp35-cp35m-win_amd64.whl

redis-3.2.1.tar.gz

下载地址: <https://pan.baidu.com/s/1gWEu4UFFXt9wggnbj1I1kQ>

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

<https://files.pythonhosted.org/packages/24/d4/06486dee0f66ef8c5080dc576fdfb33131fd2e0be3747f2be4e5634088a2/redis-3.2.1.tar.gz>

httpd-2.4.23-win64.zip

下载地址 1:

<https://www.apachelounge.com/download/VC10/>

下载地址 2:

<https://pan.baidu.com/s/1hsc1V5y>

mod_wsgi-4.6.5.tar.gz

下载地址:

https://pypi.org/project/mod_wsgi/#files

https://files.pythonhosted.org/packages/47/69/5139588686eb40053f8355eba1fe18a8bee94dc3efc4e36720c73e07471a/mod_wsgi-4.6.5.tar.gz

注意: 安装 mod_wsgi 时, 如果 Apache 安装目录不为标准安装目录 (C:/Apache24), 则需要在安装 mod_wsgi 之前设置环境变量(这里 E:/Apache24 为 Apache 实际安装目录)

```
set MOD_WSGI_APACHE_ROOTDIR=E:/Apache24
```

否则运行 python setup.py install 安装 mod_wsgi 时会报错, 如下:

Traceback (most recent call last):

File "setup.py", line 158, in <module>

```
raise RuntimeError('No Apache installation can be found. Set the 'RuntimeError: No Apache installation can be found. Set the MOD_WSGI_APACHE_ROOTDIR environment to its location.
```

参考链接:

https://pypi.org/project/mod_wsgi/#description

Microsoft Visual C++ Build Tools

下载地址 1:

<http://go.microsoft.com/fwlink/?LinkId=691126>

下载地址 2:

<https://www.microsoft.com/en-us/download/details.aspx?id=48159>

解决安装 channels-2.1.7, mod_wsgi-4.6.5 时报错问题:

error: Microsoft Visual C++ 14.0 is required. Get it with "Microsoft Visual C++ Build Tools": <http://landinghub.visualstudio.com/visual-cpp-build-tools>

mysql-connector-python-2.1.8-py3.5-windows-x86-64bit.msi

下载地址:

<https://dev.mysql.com/downloads/connector/python/2.0.html>

<https://dev.mysql.com/downloads/file/?id=480096>

chardet-2.3.0、chardet-3.0.4.tar.gz

下载地址 1: <https://pypi.python.org/pypi/chardet/>

<https://files.pythonhosted.org/packages/fc/bb/a5768c230f9ddb03acc9ef3f0d4a3cf93462473795d18e9535498c8f929d/chardet-3.0.4.tar.gz>

下载地址 2: <http://pan.baidu.com/s/1nu7XzjN>

客户端:

python 3.4.0

mysql-connector-python-2.0.5-py3.4.msi

mysql-connector-python-2.1.7-py3.4-windows-x86-64bit.msi

下载地址: <https://pan.baidu.com/s/1c3XCnG0>

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

或者

Python 3.5.4

mysql-connector-python-2.1.8-py3.5-windows-x86-64bit.msi

下载地址:

<https://dev.mysql.com/downloads/connector/python/2.0.html>

<https://dev.mysql.com/downloads/file/?id=480096>

chardet-2.3.0、chardet-3.0.4.tar.gz

下载地址 1: <https://pypi.org/project/chardet/#files>

<https://files.pythonhosted.org/packages/fc/bb/a5768c230f9ddb03acc9ef3f0d4a3cf93462473795d18e9535498c8f929d/chardet-3.0.4.tar.gz>

下载地址 2: <http://pan.baidu.com/s/1nu7XzjN>

redis-3.2.1.tar.gz

下载地址: <https://pan.baidu.com/s/1gWEu4UFFXt9wggnbjlI1kQ>

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

<https://files.pythonhosted.org/packages/24/d4/06486dee0f66ef8c5080dc576fdfb33131fd2e0be3747f2be4e5634088a2/redis-3.2.1.tar.gz>

PyCharm 4.0.5

2、 框架功能简介

1、采用 WEB 页面, 可灵活管理测试环境, 测试项目, 测试计划, 测试用例(支持模块层级化组织用例, 支持自由拖拽分组), 运行计划

- 2、支持一键复制单个测试用例，也支持一键复制用例步骤，禁用、启用用例步骤等(禁用的步骤将不被运行)
 - 3、通过项目关联计划，计划关联用例的设计方式，支持一次运行单个、多个测试计划
 - 4、支持本地、在线运行\调试模式，支持按测试计划调试，也支持运行计划调试\运行，还支持单个用例、单个测试用例套件的调试运行
 - 5、支持 HTTPS, HTTP, Webservice 协议；支持 POST, GET 方法；支持 JSON, 非 JSON 数据格式的请求，支持多种形式的
数据校验，包含数据库级别的数据校验
 - 6、纯界面化，无须写代码就可以实现如下操作：
 - a) 自定义变量存储 web 服务器、数据库服务器返回请求/查询结果
 - b) 根据自定义模式对 web 服务器返回结果进行自动校验，支持多种模式的校验，包含字符串，不包含字符串，键值提取，包含成员，不包含成员，匹配/不匹配正则表达式，完全匹配列表/元组/集合/字典
 - c) 根据界面输入的 sql 语句，执行 sql 增删改查，存储过程操作，针对只对返回单条记录的 sql 查询，还支持对查询结果进行提取，保存
 - d) 支持 url, 请求头，输入参数的动态参数化，支持全局动态参数，非全局动态参数（如存储某个接口返回结果的自定义变量）
 - e) 支持 Redis 操作(目前仅支持存储 key 值，支持自由扩展)
 - f) 支持插件函数（目前仅支持文件读取，base64 编码，支持自由扩展）
 - 7、针对脚本中已经支持的常见协议及常用数据格式，且不需对接口执行结果进行数据库级别的逻辑校验，支持界面直接增加用例而不需要改动脚本代码，即不会编码的人也可以使用本框架
 - 8、支持不同编码(utf8,ascii,gb2312)的返回结果，且可自由扩展
 - 9、支持文件、控制台的日志打印，可分别控制开关
 - 10、可集成 Jenkins 自动运行脚本
- 参考文章：
[“为 Jenkins 添加 Windows Slave 远程执行 python 项目脚本”](#)

3、服务端部署

a) Apache 配置

项目文件结构

```
管理员: C:\Windows\system32\cmd.exe

C:\Users\laiyu>cd /d D:\AutotestPlatform

D:\AutotestPlatform>dir
驱动器 D 中的卷是 常用软件
卷的序列号是 AA16-52FC

D:\AutotestPlatform 的目录

2019-04-05  14:29    <DIR>          .
2019-04-05  14:29    <DIR>          ..
2019-04-05  14:29    <DIR>          .idea
2019-04-05  17:04    <DIR>          AutotestPlatform
2019-04-05  14:29    <DIR>          log
2019-04-03  22:12             814 manage.py
2019-04-05  16:25    <DIR>          other
2019-04-05  14:29    <DIR>          static
2019-04-06  18:35    <DIR>          website
2019-04-05  14:29    <DIR>          说明
                1 个文件             814 字节
                9 个目录 68,024,614,912 可用字节

D:\AutotestPlatform>cd AutotestPlatform

D:\AutotestPlatform\AutotestPlatform>dir
驱动器 D 中的卷是 常用软件
卷的序列号是 AA16-52FC

D:\AutotestPlatform\AutotestPlatform 的目录

2019-04-05  17:04    <DIR>          .
2019-04-05  17:04    <DIR>          ..
2019-04-03  22:12             337 routing.py
2019-04-05  17:04             5,173 settings.py
2019-04-03  22:12             893 urls.py
2019-04-05  16:45             517 wsgi.py
2019-04-05  15:54                0 __init__.py
2019-04-05  17:23    <DIR>          __pycache__
                5 个文件             6,920 字节
                3 个目录 68,024,614,912 可用字节

D:\AutotestPlatform\AutotestPlatform>
```

```

D:\AutotestPlatform\website 的目录
2019-04-06 18:35 <DIR> .
2019-04-06 18:35 <DIR> ..
2019-04-03 22:12 63 admin.py
2019-04-06 19:48 <DIR> apiRunner
2019-04-03 22:12 89 apps.py
2019-04-03 22:12 5,145 assertion_type_setting_views.py
2019-04-03 22:12 4,130 browser_setting_views.py
2019-04-05 17:33 32,408 common_views.py
2019-04-03 22:12 25,199 customization_interface_views.py
2019-04-03 22:12 8,649 database_setting_views.py
2019-04-03 22:12 5,038 env_setting_views.py
2019-04-03 22:12 5,039 function_setting_views.py
2019-04-03 22:12 8,680 global_var_setting_views.py
2019-04-06 05:26 12,773 logWebsocketConsumers.py
2019-04-05 16:30 <DIR> migrations
2019-04-03 22:12 26,794 models.py
2019-04-03 22:12 4,672 operation_setting_views.py
2019-04-03 22:12 9,107 page_element_manager_views.py
2019-04-03 22:12 16,545 project_setting_views.py
2019-04-03 22:12 7,441 promble_manager_views.py
2019-04-05 09:45 470 routing.py
2019-04-06 18:35 10,505 running_plan_manager_views.py
2019-04-05 14:29 <DIR> static
2019-04-05 14:29 <DIR> templates
2019-04-06 17:20 38,634 test_case_manager_views.py
2019-04-03 22:12 7,579 test_plan_case_views.py
2019-04-05 09:45 18,140 test_plan_manager_views.py
2019-04-06 16:59 2,302 test_report_case_step_views.py
2019-04-03 22:12 2,560 test_report_case_views.py
2019-04-03 22:12 2,010 test_report_views.py
2019-04-03 22:12 17,763 test_task_manager_views.py
2019-04-06 17:50 17,918 urls.py
2019-04-03 22:12 1,415 views.py
2019-04-03 22:12 1 __init__.py
2019-04-06 18:35 <DIR> __pycache__
28 个文件 291,069 字节
7 个目录 68,024,614,912 可用字节

```

解压 httpd-2.4.23-win64.zip, 取出其中的目录(例中 Apache24), 放到目标路径(不能有空格等), 例中 D:/Apache24

检查 Apache 版本是否正确

```
cd /d D:/Apache24/bin
```

```
httpd.exe -V
```

```
Server version: Apache/2.4.23 (Win64)
```

```
.....
```

修改 Apache 配置

打开 `conf/httpd.conf` 文件, 编辑, 修改服务器根目录:

修改 `ServerRoot "c:/Apache24"` 改成 `ServerRoot "d:/Apache24"`

然后查找所有的 `"c:/Apache24"`, 全部改成 `"d:/Apache24"`

修改监听端口(可选, 根据实际需要)

`Listen 80` 改成 `Listen 8000`

修改服务器名称(建议)

`#ServerName www.example.com:80` 改成 `ServerName 10.118.52.35:80`

注: 这里我没有注册域名, 直接改成了 Apache 服务器所在 ip

去掉#注释, 打开访问日志, 当然也可以保持注释状态, 服务器磁盘写操作(推荐)

`CustomLog "logs/access.log" common`

修改`#LoadModule rewrite_module modules/mod_rewrite.so` 为如下:

`LoadModule rewrite_module modules/mod_rewrite.so`

找到如下配置

`<Directory />`

`AllowOverride none`

`Require all denied`

`</Directory>`

修改为

`<Directory />`

`AllowOverride ALL`

`Require all granted`

`</Directory>`

说明: 配置更改, 以防止出现如下情形:

Forbidden

You don't have permission to access / on this server.

启动 Apache

1) httpd.exe -k install -n Apache2.4

Installing the 'Apache2.4' service

The 'Apache2.4' service is successfully installed.

Testing httpd.conf....

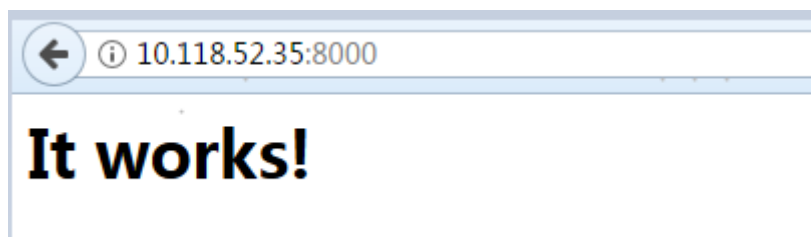
Errors reported here must be corrected before the service can be started.

注: install:把 Apache 注册为 Windows 服务, 反之 uninstall, -n 接服务名称

2) httpd.exe -k start

注: start 启动, stop 停止

浏览器访问



注: 截图为旧图, 未替换为新的, 实践 ip 为 10.118.52.35

再次修改 Apache 配置

cmd 打开控制台, 进入到进入 %PYTHON_HOME%\Scripts(例中为 D:\Program Files (x86)\python35\Scripts) 目录,

执行以下命令 mod_wsgi-express.exe module-config

```
d:\Program Files (x86)\python35\Scripts>mod_wsgi-express.exe module-config
LoadFile "D:/Program Files (x86)/python35/python35.dll"
LoadModule wsgi_module "D:/Program Files (x86)/python35/lib/site-packages/mod_wsgi-4.6.5-py3.5-win-amd64.egg/mod_wsgi/server/mod_wsgi.cp35-win_amd64.pyd"
WSGIPythonHome "D:/Program Files (x86)/python35"
```

(说明: mod_wsgi-express.exe 是在安装完 mod_wsgi 后自动生成的)

复制命令输出内容, 黏贴到 apache httpd.conf 末尾, 如下:

```
LoadFile "D:/Program Files (x86)/python35/python35.dll"
```

```
LoadModule wsgi_module "D:/Program Files
```

```
(x86)/python35/lib/site-packages/mod_wsgi-4.6.5-py3.5-win-amd64.egg/mod_wsgi/server/mod_wsgi.cp35-win_a  
md64.pyd"
```

```
WSGIPythonHome "D:/Program Files (x86)/python35"
```

然后再追加以下内容

```
WSGIScriptAlias / D:/AutotestPlatform/AutotestPlatform/wsgi.py
```

```
WSGIPythonPath D:/AutotestPlatform/AutotestPlatform/website
```

```
Alias /static/ D:/AutotestPlatform/website/static/
```

```
<Directory D:/AutotestPlatform/website/static>
```

```
Require all granted
```

```
</Directory>
```

```
<Directory D:/AutotestPlatform/AutotestPlatform/website/>
```

```
<Files wsgi.py>
```

```
Require all granted
```

```
</Files>
```

```
</Directory>
```

b) Django 配置

Django 访问 IP 配置

修改应用的 settings.py(例中为 D:\AutotestPlatform\AutotestPlatform\settings.py), 编辑, 找到 ALLOWED_HOSTS 修改为如下值, 其中 192.168.52.35 是 Django 所在主机 ip, 也就是客户端浏览器访问用的 IP

```
ALLOWED_HOSTS = ['localhost', '127.0.0.1', '10.118.52.35']
```

创建库配置

修改应用的 settings.py(例中为 D:\AutotestPlatform\AutotestPlatform\settings.py)

```

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'testplatform',
        'USER': 'testacc',
        'PASSWORD': 'test1234',
        'HOST': '10.118.52.33',
        'PORT': '3306',
        'OPTION': {
            'init_command': 'SET default_storage_engine=INNODB'
        }
    }
}

```

如上，设置蓝色部分的 key 值，分别表示数据库用户名，密码，服务器 ip，端口

c) APIRunner 配置

数据库配置

编辑 AutoestPlatform/website/apiRunner/conf/db.conf 文件

```

[TESTPLATFORM]
host = 10.118.52.33
port = 3306
user = testacc
passwd = test1234
db = testplatform
charset = utf8

```

说明：这里配置的数据库，即为 Django 使用的数据库配置，如上，设置蓝色部分的 key 值，分别表示数据库服务器 ip，端口，用户名，密码，数据库名，字符集

https SLL、TSL 版本配置

编辑 AutoestPlatform/website/apiRunner/conf/https.conf 文件

```

[HTTPS]
SSL_OR_TLS_PROTOCOL = V2

```

[README]

#SSL_OR_TLS_PROTOCOL 可选值 V1、 V2、 V23、 V3，不区分大小写

#V1 - TLSv1

#V2 - SSLv2

#V23 - SSLv23

#V3 - SSLv3

d) 系统环境变量配置

新建 PYTHONPATH 系统环境变量(如果不存在的话), 添加 Django 项目根目录绝对路径(例中为 D:/AutotestPlatform)到 PYTHONPATH 系统环境变量

e) 数据库的创建

```
CREATE DATABASE IF NOT EXISTS `testplatform` DEFAULT CHARACTER SET utf8;
```

f) 创建表

```
cd /d D:\AutotestPlatform
```

```
python manage.py makemigrations website
```

```
python manage.py migrate
```

执行初始化 sql

详情见“初始化 sql.sql”

g) 重启 Apache 并启动 Django 应用

```
D:\Apache24\bin>httpd.exe -k stop
```

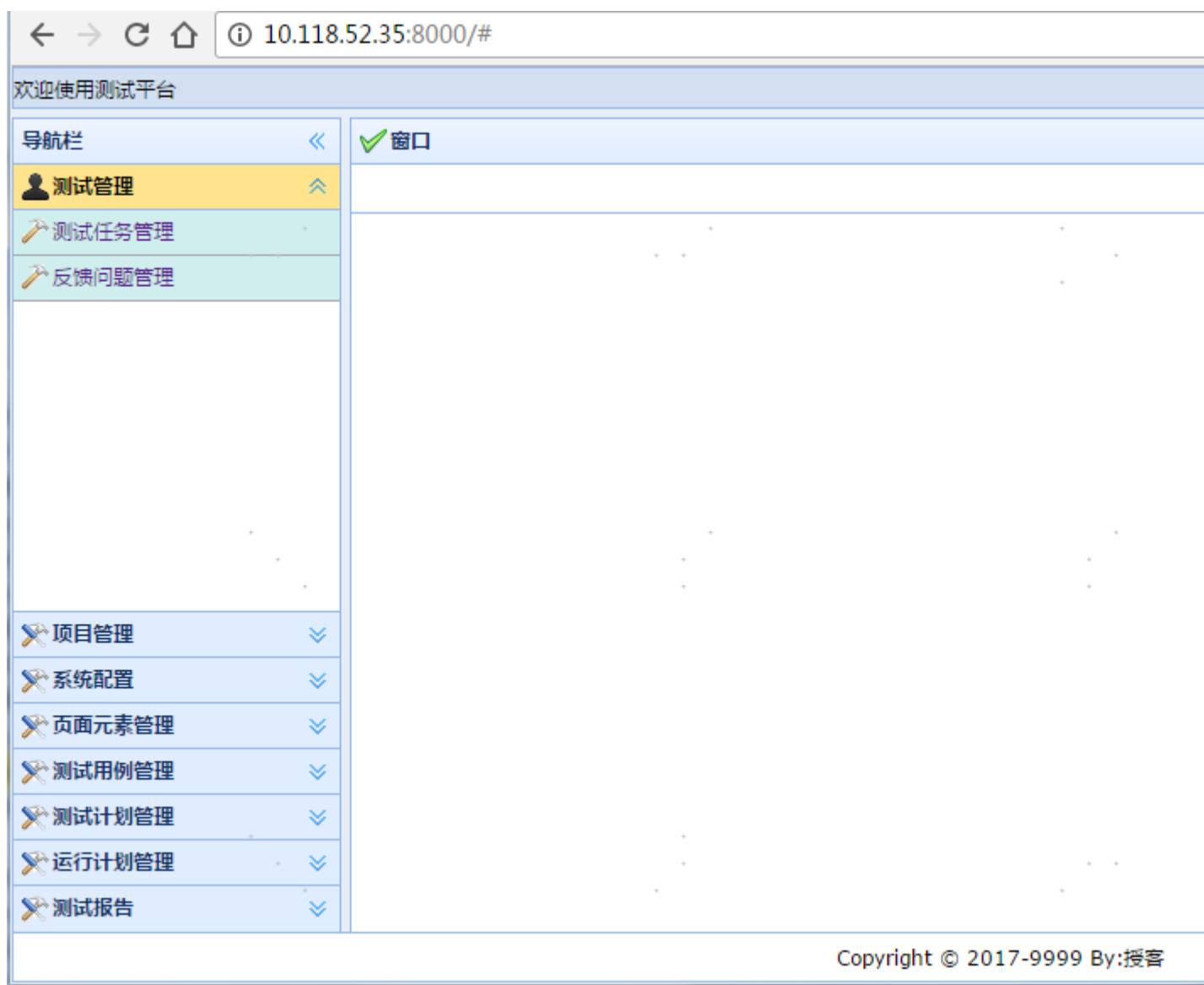
```
The 'Apache2.4' service is stopping.
```

```
The 'Apache2.4' service has stopped.
```

```
D:\Apache24\bin>httpd.exe -k start
```

说明:到这一步, 已经可以浏览器访问了

浏览器验证



h) 验证 daphne 可否正常运行

- 1、添加 \$PYTHON_HOMED\Scripts (例中为 D:\Program Files\Python35\Scripts) 到系统环境变量
- 2、cmd 控制台运行命令, 启动 daphne

```
daphne -b 0.0.0.0 -p 8001 AutotestPlatform.asgi:application
```

注意: 如果启动失败, 程序会自动退出命令, 没有任何提示, 如下, 否则也不会有提示, 但是不会自动退出命令

```
C:\Users\01367599>daphne -b 0.0.0.0 -p 8000 AutotestPlatform.asgi:application
C:\Users\01367599>平.
```

运行成功后, 关闭 cmd 窗口, 以关闭 daphne 进程

i) 修改 apache 配置

打开 conf/httpd.conf 文件，

修改

```
#LoadModule proxy_module modules/mod_proxy.so
```

为

```
LoadModule proxy_module modules/mod_proxy.so
```

修改

```
#LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
```

为

```
LoadModule proxy_wstunnel_module modules/mod_proxy_wstunnel.so
```

在配置文件最末尾添加以下内容

```
ProxyPass /ws/debugAPICaseOrSuit/ ws://10.118.52.35:8001/debugAPICaseOrSuit/
```

```
ProxyPass /ws/debugAPITestPlan/ ws://10.118.52.35:8001/debugAPITestPlan/
```

```
ProxyPass /ws/debugAPIRunningPlan/ ws://10.118.52.35:8001/debugAPIRunningPlan/
```

注意：这里 10.118.52.35 为 daphne 启动时所在服务器 ip（基于当前部署方式，固定为为 Apache 服务器 ip，如果 daphne 和 Apache 两者 ip 不相同，js 文件中基于当前的实现方式是没法动态获取到 ip 地址的），根据实际填写

j) 开启 daphne 监控

Cmd 窗口执行以下命令，开启 daphne 并对其进行监控

```
cd /d D:\AutotestPlatform
```

```
python monitorDaphne.py
```

注意：如果有对代码做涉及 websocket 消息发送的变更，并重启 Django 项目，要重新开启 daphne 监控

4、客户端部署

a) 数据库配置

编辑 conf/db.conf 文件

```
[TESTPLATFORM]
```

```
host = 192.168.1.103
```

```
port = 3306
user = testacc
passwd = test1234
db = testplatform
charset = utf8
```

说明：这里配置的数据库，即为部署服务端时使用的数据库。

b) https SLL、TSL 版本配置

编辑 conf/https.conf 文件

[HTTPS]

SSL_OR_TLS_PROTOCOL = V2

[README]

#SSL_OR_TLS_PROTOCOL 可选值 V1、 V2、 V23、 V3，不区分大小写

#V1 - TLSv1

#V2 - SSLv2

#V23 - SSLv23

#V3 - SSLv3

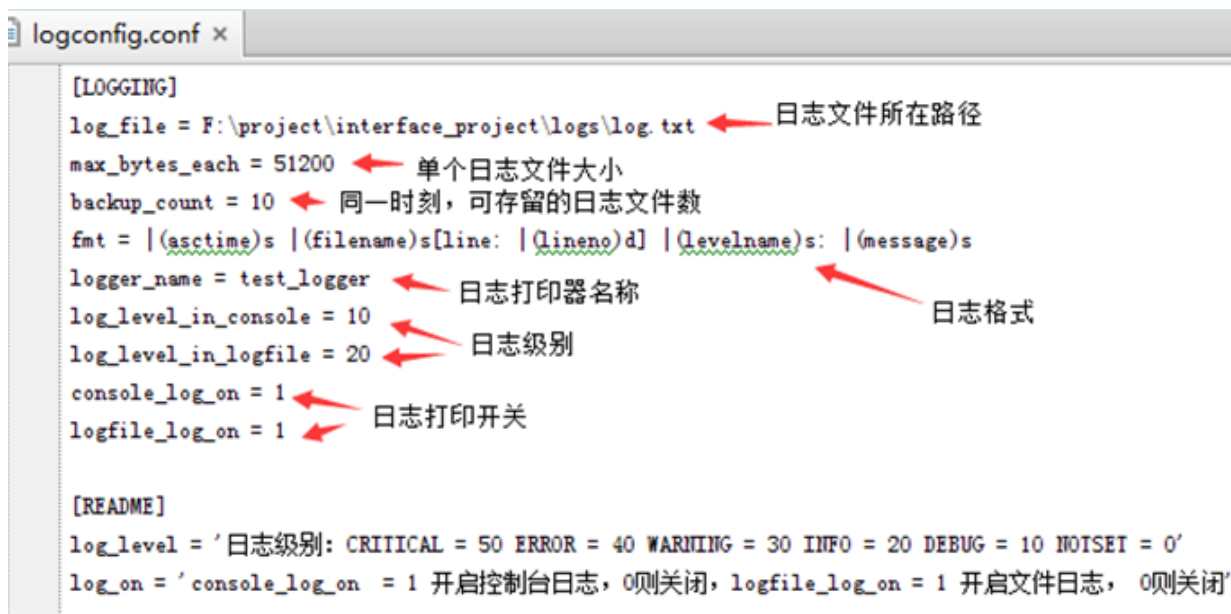
c) 服务器端日志配置



说明：WEB_DEBUG_LOG_LEVEL 用于 web 页面运行(调试)实时打印日志级别控制，可选值 debug, info, warn, error, critical，从左到右，级别升序，即后者的不包含前者，比如设置为 info，那么不显示 debug 日志

d) 客户端日志配置

编辑 conf/log.conf 文件



5、大致操作流程

a) 环境配置

系统配置-环境配置

b) 新增项目

项目管理-API 项目配置

c) 系统配置

数据库配置

操作配置(如果没有初始化配置的话)

断言配置(如果没有初始化配置的话)

全局变量配置(有必要的話)

d) 新增用例

测试用例管理-API 测试用例管理, 支持嵌套管理

e) 新增 API 计划

测试计划管理-API 测试计划

f) 关联 API 测试用例

可先关联要最先运行的用例, 关联后也可做适当的调整。

g) 新增并执行“运行计划”

新增运行计划后, 可直接点击对应运行计划的“在线运行”, 然后点击“在线运行”、“在线调试”

也可以直接客户端传参运行

cd /d E:\interface_project_for_dev

```
python main.py rop 1517487657
```

说明:

1、这里的数字 1517487657 即为运行计划编码

2、rop, 大小写不敏感 -> run one project 运行单个项目

h) 查看测试报告

所在模块: 测试报告-API 测试报告

6、测试任务可视化管理功能介绍

(附加功能)

除了接口自动化, 平台还提供了一个简单的测试任务管理功能, 测试任务等可视化管理

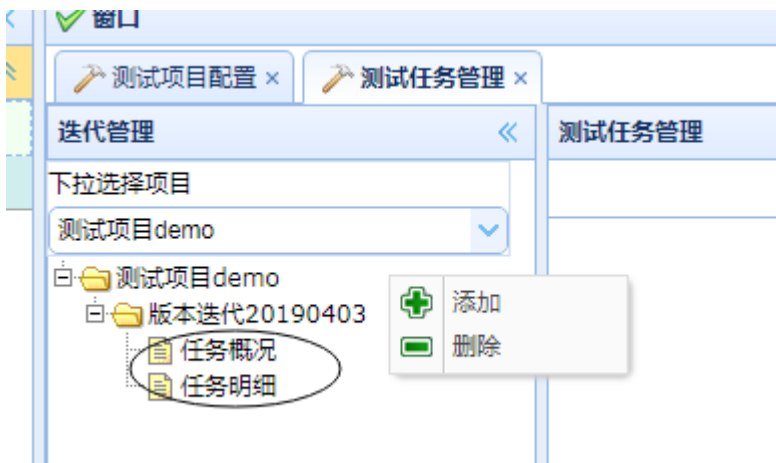
① 项目配置-测试项目配置, 新增 项目配置



② 测试管理-测试任务管理

如下, 可以右键根目录, 新增迭代子目录, 然后右键新增子节点, 分别命令为“任务概况”, “任务明细”

注意: 这里“任务概况”, “任务明细”名称固定, 不是这个名称, 点击节点无反应



新增-任务概况

新增后, 会往任务明细插入一条记录

» 窗口

测试任务管理 ×

» 测试任务管理

任务概况 × 任务明细 ×

新增 修改 删除 保存 取消

<input type="checkbox"/>	任务ID	功能模块	是否完成	需求任务	开发任务细分	预估转测时间	实际转测时间	是否延迟	开发负责人	测试负责人	产品负责人	备注	顺序
1	4	根模块-子模块-子模块	未完成	新任务需求名称	xxx任务开发1			否					4
2	3				xxx任务开发3	2019-04-09	2019-04-12		赖某人	林某人	坏某人		3
3	1	根模块-子模块-子模块	未完成	任务需求名称	xxx任务开发2	2019-04-09	2019-04-12	是	赖某人	林某人	坏某人		2
4	2				xxx任务开发	2019-04-09	2019-04-08	否	赖某人	林某人	坏某人		1

编辑-任务明细

» 窗口

测试任务管理 ×

» 测试任务管理

任务概况 × 任务明细 ×

新增 修改 删除 保存 取消

<input type="checkbox"/>	任务ID	功能模块	需求任务	测试负责人	子任务	进度	预估耗时(H)	预计截止时间	实际完成时间	是否延迟	进度更新轨迹	备注
1	2	根模块-子模块-子模块	新任务需求名称	小某人	xx功能测试	100%	3	2019-04-06	2019-04-06	否	0%>周六(04.06):100%	
2	3	根模块-子模块-子模块			xxx功能测试	15%	3	2019-04-06			0%>周六(04.06):15%	
3	1	根模块-子模块-子模块	任务需求名称	林某人	功能测试	0%					0%	

7、用例步骤填写规范

a) 接口请求

针对 http,https 请求

1. 执行操作

test_api_for_urlencode

针对请求参数为以下形式的:

arg1=value1&arg2=value2&...&argN

test_api_for_json

针对请求参数为以下形式(json 格式)的:

```
{
  "key1": "value1",
  "key2": "value2",
  .....,
  "keyN": "valueN"
}
```

test_api_for_xml

针对请求参数为以下形式(xml 形式)的:

```
<ArrayOfString xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://WebXml.com.cn/">
  <string>阿尔及利亚,3320</string>
  <string>阿根廷,3522</string>
  <string>阿曼,3170</string>
  <string>阿塞拜疆,3176</string>
  <string>埃及,3317</string>
  <string>埃塞俄比亚,3314</string>
  <string>爱尔兰,3246</string>
  <string>奥地利,3237</string>
  <string>澳大利亚,368</string>
  <string>巴基斯坦,3169</string>
  <string>巴西,3580</string>
  <string>保加利亚,3232</string>
  <string>比利时,3243</string>
</ArrayOfString>
```

2. 请求头

选填, 必须符合 json 格式, 形如以下

```
{
  "User-Agent": "Mozilla / 5.0(Windows NT 6.1; WOW64) AppleWebKit / 537.36(KHTML, like Gecko) Chrome
/ 49.0 .2623 .112 Safari / 537.36",
  "Cache-Control": "no-cache"
}

{"Content-Type":"application/x-www-form-urlencoded","charset":"utf-8"}
```

注: 这里添加的请求头, 不会覆盖“全局请求头”, 而是在全局请求头的基础上新增请求头域及对应值

3. URL/SQL

绝对 URL, 形如以下

```
/pages/nodeTree
/pages/nodeTree?
/pages/nodeTree?page=1&size=10
```

注:

- 1、url 打头可以携带/, 也可以不带/, 如 pages/nodeTree
- 2、如果参数携带中文, 要放到 url 中, 则必先 urlencode, 形如:
/sug?code=utf-8&q=%E7%94%B5%E5%8A%A8%E8%BD%A6&callback=cb

例中, E7%94%B5%E5%8A%A8%E8%BD%A6 decode 后为“电动车”

4. 输入参数

选填

如果执行操作为 `test_api_for_json`, 输入参数的必须符合 json 格式, 形如以下

```
{
  "arg1": "value1",
  "arg2": "value2",
  .....,
  "argN": "valueN"
}
```

如果执行操作为 `test_api_for_urlencode`, 输入参数可以写成 json 格式的(包含两个及以上的相同参数名时不支持), 也可以如下形式的(推荐):

```
arg1=value1&arg2=value2&arg3=value3&.....&argN=valueN
```

```
arg1=value1&arg2=value2&arg3=value3&.....&argN=valueN 安全模式 xxx
```

说明:

安全模式 **xxx**: 指定不需要进行 `url` 编码的字符, 默认字符为 `&` 和 `=`

如果不想指定“安全模式字符”则附加值写 “安全模式无”

例:

json:

```
{
  "userName": "dataviewer",
  "userPwd": "f0NfsZjEy65NSc0qBOB4Vzef-NZ0B6n2cXfGvHmXESa-0uzwD3-Q4T0kq0kKdoB9tuCr3N1",
  "captcha": "5555",
  "channelType": "dt",
  "remark": ""
}
```

urlencode:

```
custNo=全部&describe=全部&waybillNo[]=198130045411&waybillNo[]=198129993774
```

执行操作为 `test_api_for_xml`, 输入参数的必须符合 XML 规范, 形如以下

```
<ArrayOfString xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://WebXml.com.cn">
```



```
<string>阿尔及利亚,3320</string>
<string>阿根廷,3522</string>
<string>阿曼,3170</string>
<string>巴西,3580</string>
<string>保加利亚,3232</string>
<string>比利时,3243</string>
</ArrayOfString>
```

5. 检查响应

支持 body, header, code, 含义如下:

body - 针对响应体执行断言、从响应体中提取目标值

header - 针对响应头执行断言、从响应头中提取目标值

code - 针对响应代码执行断言、从响应代码中提取目标值

6. 校验规则及校验模式

选填, 参考文档“结果断言和提取服务器返回结果.txt”

7. 输出

选填, 参考文档“结果断言和提取服务器返回结果.txt”

8. 协议

选填, 如果步骤请求使用的协议和项目统一配置不一样, 则可在此处做变更, 仅支持 https, http

9. 主机地址

选填, 如果步骤请求使用的主机地址和项目统一配置不一样, 则可在此处填写

10. 端口

选填, 如果步骤请求使用的主机地址和项目统一配置不一样, 则可在此处填写

11. 运行次数

必填, 单个步骤的运行次数, 大于等于 1, 该参数为通用参数, 和步骤类型无关, 往下不再赘述

12. 失败重试次数

必填, 步骤执行失败时的重试次数, 大于等于 1, 该参数为通用参数, 和步骤类型无关, 往下不再赘述

13. 重试频率

必填, 失败后重试间隔, 大于等于 1, 单位 秒, 该参数为通用参数, 和步骤类型无关, 往下不再赘述

比如设置失败重试次数为 3, 重试频率为 2, 则意为该步骤运行失败后, 等待 2 秒后会再次执行, 如此循环, 直到执行成功、超过最大重试次数后停止。

b) 操作数据库

① 执行操作

目前支持以下几种操作

`select_one_record`

针对查询结果返回单条记录的查询,如果返回了多条,只会取第一条记录

`update_record`

执行数据库 `update` 操作

`delete_record`

执行数据库删除记录操作

`truncate_table`

执行数据库清空表操作

`call_proc`

执行数据库存储过程

`insert_record`

往数据库插入记录

② URL/SQL

必填, 形如

```
SELECT phone FROM user WHERE user_name = 'dataviewer';
```

```
UPDATE user SET qq=1033553122 WHERE user_name='shouke';
```

注意: `where` 后面跟的“参数值”(例中的 `'dataviewer'`)如果包含单引号', 程序会自动转换为双引号 "

③ 输入参数

选填,如果有多个参数,则用逗号分隔, 其中字符串类型的值, 必须用双引号, 否则会报错。

例子: 携带输入参数的 SQL

URL/SQL:

```
SELECT phone FROM ddt_user WHERE qq= %s AND user_name="%s";
```

说明: 这里的 `%s` 占位符和以下输入参数中值, 从左到右, 一一对应。

注意: 如果 `%s` 占位符对应的是一个字符串类型的值, 一定要加双引号, 形如: `"%s"`

输入参数:

1033553122, "dataviewer"

④ 校验规则及校验模式

选填, 参考文档“结果断言和提取服务器返回结果.txt”

⑤ 输出

选填，参考文档“结果断言和提取服务器返回结果.txt”

c) 操作 Redis

① 执行操作

目前支持以下几种操作

set_key_value # 存储键值

输入参数格式: key, value

② 输入参数

选填,如果有多个参数,则用逗号分隔

③ 校验规则及校验模式

选填,暂时未实现

④ 输出

选填,暂时未实现

d) 执行函数

① 操作对象

目前支持函数:

死等待

② 输入参数

根据参数样例填写

③ 校验规则及校验模式

选填,参考文档“结果断言和提取服务器返回结果.txt”

④ 输出

选填,参考文档“结果断言和提取服务器返回结果.txt”

e) 关于用例及步骤运行顺序说明

1、用例执行顺序: 按计划用例关联列表中的用例顺序升序, 从上往下执行

2、用例步骤执行顺序: 按用例步骤列表中的步骤顺序降序, 从下往上执行。

8、参数化

① 全局变量

全局变量配置 ×						
新增 修改 删除 保存 取消						
	ID	变量名	变量值	备注	项目类型	所属项目
26	14	global_emailAddr3	unp-01@sfuat.com		API项目	UNP统一通知平台API测试,UNP统一通知平台API测试
27	13	global_emailAddr2	sfuat777@sfuat.com		API项目	UNP统一通知平台API测试
28	10	global_emailAddr1	sfuat888@sfuat.com		API项目	UNP统一通知平台API测试
29	9	global_smsBusinessTemplateChannelUnau	TESTCODE-SMS-BUSINESS-TEMPLATE2		API项目	UNP统一通知平台API测试
30	20	global_sfJobNum2	259631		API项目	UNP统一通知平台API测试
31	8	global_sfJobNum	01367599		API项目	UNP统一通知平台API测试
32	7	global_gangAoTaiMobile	852-31235261		API项目	UNP统一通知平台API测试
33	4	global_smsChannelTemplateParam	{"smsVerifyCode":654321}		API项目	UNP统一通知平台API测试
34	3	global_smsBusinessTemplate	TESTCODE-SMS-BUSINESS-TEMPLATE		API项目	UNP统一通知平台API测试
35	5	global_accessID	o17O12xf		API项目	UNP统一通知平台API测试
36	6	global_accessToken	a950ca7a32104ebfba5ba214101fe66e		API项目	UNP统一通知平台API测试
37	2	global_mobile	17817738178		API项目	UNP统一通知平台API测试
38	1	global_headers	{ "public-int-gw.intsit.sfdc.com.cn":{"Content-Type":"application/json","charset":"utf-8"} }		API项目	UNP统一通知平台API测试

注意：

- 1、这里的全局变量，和环境及项目两个因素紧密关联
- 2、全局请求头，名称固定，必须为小写的 `global_headers`，注意，这里的请求头和用例步骤定义的请求头是“并集”关系。格式,形如以下，必须符合 json 格式：

```
{
  "host1":{"header_field1":"value1", "header_field2":"value2", ..., "header_fieldN":"valueN"},
  "host2":{"header_field1":"value1", "header_field2":"value2", ..., "header_fieldN":"valueN"},
  ...
}
```

- 3、为防止变量值被覆盖，全局变量的名称，必须以 `global_` 打头(大小写不限)，`global_` 之后的部分大小写敏感
- 4、修改全局变量，会通过正则匹配的方式，替换其关联的老项目中对应用例，对应步骤中的全局变量

② 局部变量

在用例步骤自定义的变量。

注意：

- 1、局部变量，名称不能包含“空格，-，tab 符”，且不能以 `global_`、`GLOBAL` 开头
- 2、局部变量，大小写敏感

③ 变量的引用

`${变量名}`，比如 `${global_headers}`

④ 变量使用范围

支持字段“请求头，URL/SQL，输入参数，校验模式， 主机地址”

注意：当前步骤，输出中的定义的“变量”可用只可用于当前步骤的“校验模式”及其它步骤中上述字段，但是不可用于当前步骤中的“请求头，URL/SQL，输入参数，主机地址”

样例：

请求头	请求方法	URL/SQL
{ "Token": "\$global_token\$"	GET	/res/rpatchca.png? \$global_url_arg\$

校验规则	校验模式	输出
键值相等	[{"模式": {"userName": "\$userName\$"}, "消息": "FAILURE#用户名不为dataviewer"}]	{ "dic": { "userName": { "userName": "value", "atten": { "atten": "value" } } } }
匹配正则表达式	[{"模式": "\\qq\\ "(+?)\\", "消息": "FAILURE#不符合断言时返回的消息"}]	{ "re": { "userName2": "userName": "\\ "(+?)\\ " } }

注意：截图为旧版本的截图，新版本中变量引用已经改成 `#{varName}`，如下，

ID50-N_消息过期预警(电话号码外呼)

	输入参数	检查响应	校验规则	校验模式	输出
1	<pre>{ "token": "unp", "requestID": "\${requestId}", "callbackRecord": "false" }</pre>	body	包含字符串	<pre>[{ "模式": "\\success\\": true, "消息": "fail#success不为true" }, { "模式": "\\state\\": 2, "消息": "fail#邮件未被过滤" }, { "模式": "\\receiver\\": "\${global_mobile}", "消息": "fail#接收人不为\${global_mobile}" }, { "模式": "\\failureDec\\": 2, "消息": "fail#失败原因描述不对，正确描述为：预期发送时间已过" }]</pre>	
2	5	body			
3	<pre>{ "userId": "\${global_mobile}", "templateCode": "\${global_teleCallBusinessTemplate}", "templateParam": "\${global_teleCallChannelTemplateParam}", "expectedTime": "\${curTime}", "expiredTime": "2019-3-18 16:35:00", "accessId": "\${global_accessID}", "accessToken": "\${global_accessToken}" }</pre>	body	包含字符串	<pre>[{ "模式": "\\success\\": true, "消息": "fail#success不为True" }]</pre>	{ "dic": { "requestId": { "requestId": "value" } } }
4					

以下不再赘述

校验规则	校验模式	输出
db列值相等	<pre>[{"模式":["\$user_name\$","dataviewer"],"消息":"FAILURE#姓名不为dataviewer"}, {"模式":["\$phone\$","18159001414"],"消息":"FAILURE#手机号不为18189001424"}]</pre>	
db列值相等	<pre>[{"模式":["\$user_name\$","dataviewer"],"消息":"FAILURE#姓名不为dataviewer"}, {"模式":["\$phone\$","18159001414"],"消息":"FAILURE#手机号不为18189001424"}]</pre>	<pre>{"db":{"user_name":2,"phone":1}}</pre>
xpath断言	<pre>[{ "模式": { "//return": "dataviewer" }, "消息": "FAILURE#获取的实体集团卡号错误" }, { "模式": { "//return" "\$groupCode_1\$" }, "消息": "FAILURE#获取的实体集团卡号错误" }]</pre>	<pre>{"xpath":{"groupCode": "//return"}}</pre>

校验规则	校验模式	输出
键值相等	<pre>[{"模式": {"userName": "\$userName\$"}, "消息": "FAILURE#用户名不为dataviewer"}]</pre>	<pre>{"dic":{"userName":{"userName":"value"},"atten":{"atten":"value"}}</pre>
POST	/ws/ddtDimCustGroupMapping	<pre><soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ser="http://service.rpt.ddtm.platform.ddt.sf.com/"> <soapenv:Header/> <soapenv:Body> <ser:findByAccountCompany> <!--Optional:--> <!--Optional:--> <!--Optional:--> <userName>\$userName\$</userName> <!--Optional:--> <accountCompany>2017112815</accountCompany><vaildFlag>1</vaildFlag><isReplace>0</isReplace> </ser:findByAccountCompany> </soapenv:Body> </soapenv:Envelope></pre>

注意:

截图中的 FAILURE# 已经改成了 fail#, 不区分大小写, 比如也可以 FAIL#, 这个是给旧版程序用的, 新版本填不填写都无所

⑤ 插件函数

语法: `${__functionName()}、${_functionName()}`

说明:

- 1、函数名称不包含空格部分(函数名称同`${ }`之间可以存在空白字符, 形如 `${ __myfunction() }`), 以双、单下划线打头,
- 2、支持函数嵌套
- 3、所有字符串参数, 均用“英文双引号”引起来

样例:

```
{
  "userId": "${global_emailAddr1}",
  "templateCode": "${global_emailBusinessTemplate}",
  "templateParam": "${global_emailChannelTemplateParam}",
  "subject": "邮件内嵌图片附件",
  "attachments": {"emailPic.png": "${__base64encode(${_read_file("emailPic.png", "rb")})"}},
  "nestedImgs": ["emailPic.png"],
  "accessId": "${global_accessID}",
  "accessToken": "${global_accessToken}"
}
```

body 包含字符串

支持的函数列表:**1) 读取文件**

`${__read_file(file_path, mode, encoding=None)}`

参数说明:

file_path: 为文件路径,

针对服务端: 文件存放地址为 `AutotestPlatform/website/apiRunner/testdata`

针对客户端: 文件存放地址为 `interface_project_for_dev/testdata`

编写规范: 相对于文件存放地址的相对路径, 比如 `emailPic.png` (绝对路径

`AutotestPlatform/website/apiRunner/testdata/emailPic.png`)、`mycustomdir/emailPic.png`(绝对路径

`AutotestPlatform/website/apiRunner/testdata/mycustomdir/emailPic.png`

mode: 文件打开方式, 支持 `r`, `r+`, `rw`, `rb`, `rb+`,

encoding: 如果 `mode` 为 `rb`, `rb+`, 则不填写, 或者填写 `None`

函数返回值

如果读取文件成功, 如果 `mode` 不为 `rb`, `rb+` 则返回字符串数据, 否则返回字节数据, 如果读取文件失败, 则返回 `None`

2) base64 编码

`${__base64encode(byteest, altchars=None)}`

参数说明：

byteest：字节数据

函数返回值

如果编码成功，返回 base64 编码后的数据，如果读取编码失败，则返回 None

函数使用范围说明

支持字段“请求头，URL/SQL，输入参数，校验模式， 主机地址”

9、 运行与调试

a) 服务器在线(调试)运行

① 项目级别的(调试)运行

窗口

运行计划 ×

新增

修改

删除

保存

取消

	项目名称	计划名称	自动化脚本所在父级	Python.exe程序绝对路	是否启用	运行状态	失败原因备注	
1	DEMO项目	测试计划2,demo测试计划	.	.	启用	暂未运行		<div>重置状态</div> <div>在线运行</div> <div>插入</div>
2	UI项目demo	ui计划demo			启用	运行失败		<div>重置状态</div> <div>在线运行</div> <div>插入</div>
3	DEMO项目	demo测试计划	.	.	启用	暂未运行		<div>重置状态</div> <div>在线运行</div> <div>插入</div>
4	DEMO项目	demo测试计划			启用	运行失败	项目运行失败：计划ID：1，失败原因：存在运行失败、被阻塞的用例	<div>重置状态</div> <div>在线运行</div> <div>插入</div>

欢迎使用测试平台

窗口

运行计划 ×

run-demo运行计划 ×

在线运行

调试运行

清除日志

2019-04-06 23:51:31 logWebsocketConsumers.py 275 INFO 运行当前程序的Python版本：3.5.4

2019-04-06 23:51:31 logWebsocketConsumers.py 279 INFO 当前运行计划编码为：1554493626460440

2019-04-06 23:51:31 logWebsocketConsumers.py 117 INFO 开始调试运行

2019-04-06 23:51:31 APIRunner.py 330 INFO 当前运行计划编码为：1554493626460440，正在查询该运行计划相关信息

2019-04-06 23:51:31 mydb.py 151 INFO 执行的查询语句为：SELECT running_plan_name,project_id, project_name, plan_name, plan_id, valid_flag FROM `website_running_plan` WHERE running_plan_num =1554493626460440

2019-04-06 23:51:31 APIRunner.py 339 INFO 待运行项目：名称：DEMO项目，ID：1，关联的测试计划有：demo测试计划

2019-04-06 23:51:31 APIRunner.py 344 INFO =====开始执行运行计划[名称：demo运行计划]=====

2019-04-06 23:51:31 running_plan.py 34 INFO 正在查询项目[ID：1,名称：DEMO项目]相关信息

2019-04-06 23:51:31 mydb.py 151 INFO 执行的查询语句为：SELECT protocol, host, port, environment_id, valid_flag FROM `website_api_project_setting` WHERE id = 1

2019-04-06 23:51:31 running_plan.py 42 INFO 正在查询与项目关联的数据库信息

2019-04-06 23:51:31 running_plan.py 67 INFO 正在查询与项目关联的全局变量

2019-04-06 23:51:31 running_plan.py 97 INFO =====开始运行测试项目[名称：DEMO项目，ID：1]=====

2019-04-06 23:51:31 test_project.py 44 INFO 待运行计划ID列表：['1']

2019-04-06 23:51:31 test_project.py 47 INFO 正在查询与测试计划关联的用例树节点

2019-04-06 23:51:31 test_project.py 63 INFO 正在查找用例树节点信息

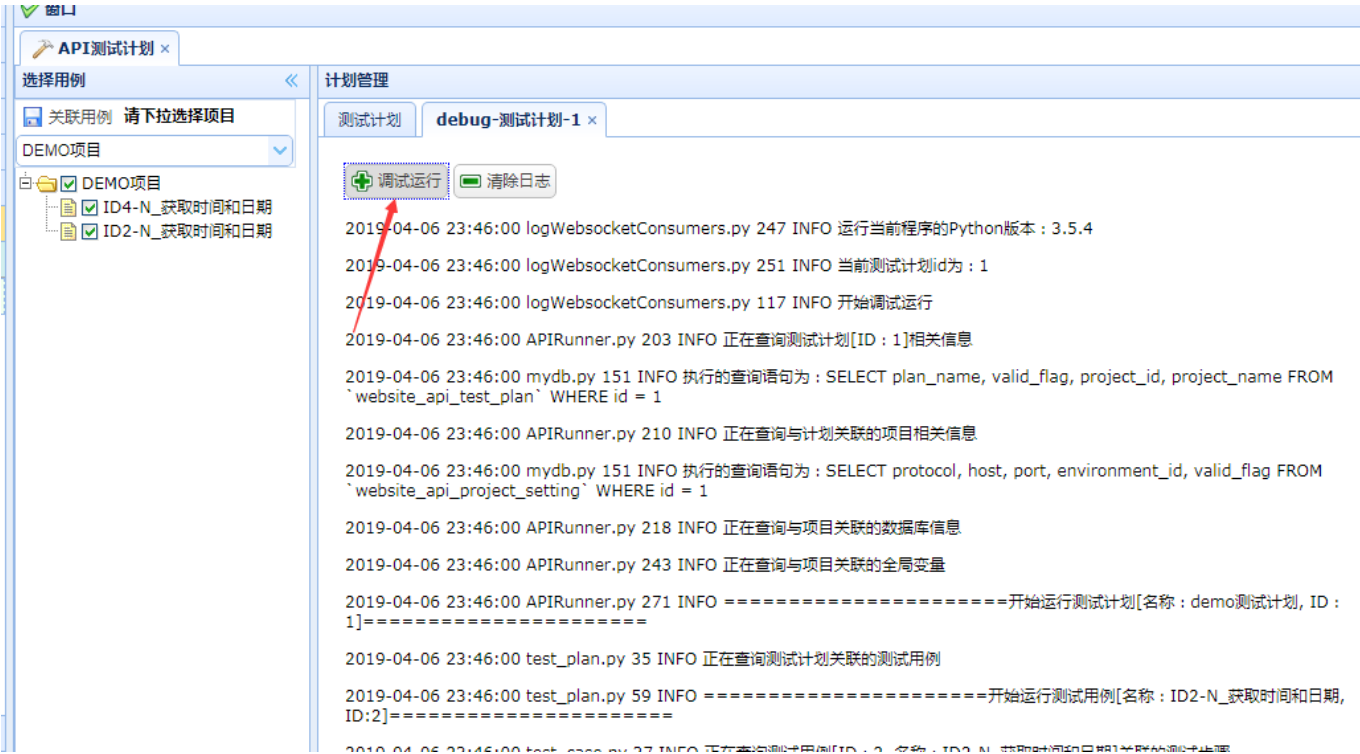
2019-04-06 23:51:31 mydb.py 151 INFO 执行的查询语句为：SELECT t1.text,t1.parent_id, COUNT(t2.id) FROM `website_api_case_tree` AS t1 LEFT JOIN `website_api_case_tree` AS t2 ON t2.parent_id = t1.id WHERE t1.id = 1

说明：

调试运行：不生成测试报告

在线运行：生成测试报告

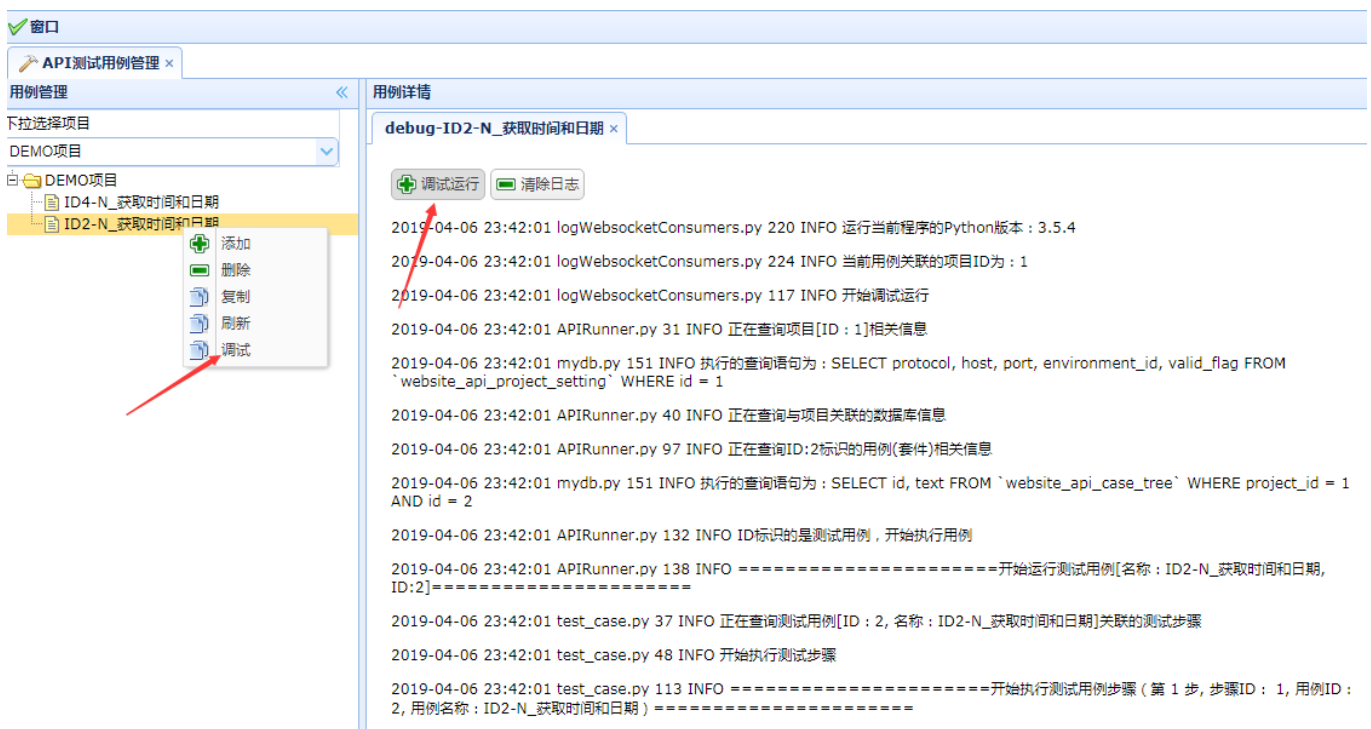
② 测试计划级别的调试



说明：点击 清除日志 按钮，可以清空页面日志

③ 用例(套件)级别的调试





b) 客户端本地调试

① 项目级别的调试

python main.py rop 1517487657 debug

说明：

1、这里 1517487657 为运行计划编码

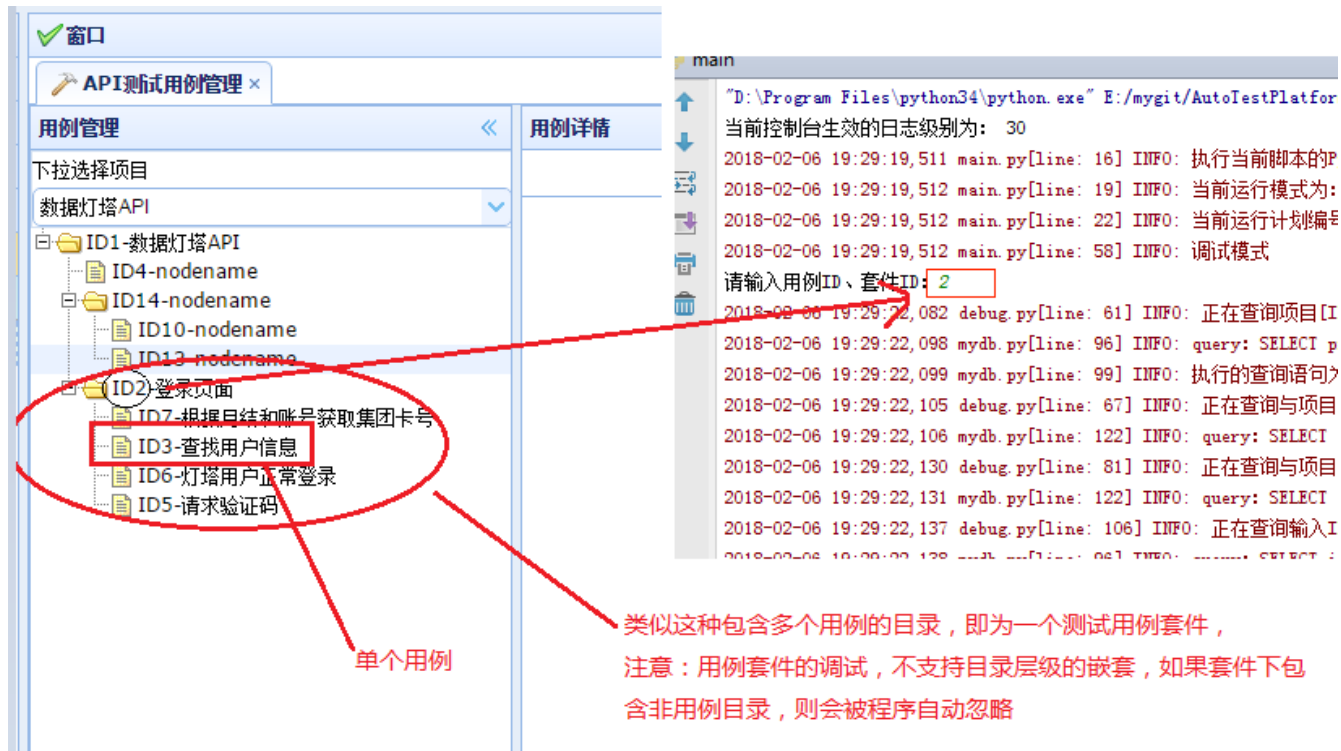
2、开启调试模式，不会往数据库记录相关数据

② 用例(套件)级别的调试

python main.py debug 项目 ID, 形如: python main.py debug 1

说明: debug 大小写不敏感

程序运行后, 按要求输入会要求测试用例、用例套件的 ID, 回车即可查看运行结果



注意:

1、用例套件的调试，不支持目录层级的嵌套，如果套件下包含非用例目录，会被自动跳过。

2、套件下用例的运行顺序：从下往上，顺序执行。

3、调试之前，需要开启并配置好项目相关信息

10、数据备份

冷备份:

```
net stop mysql
```

```
xcopy /E /Y D:\MySQL57\data\testplatform\* D:\MySQL57\backup\%date:~0,4%%date:~5,2%%date:~8,2%\
```

```
net start mysql
```

命令说明:

```
net stop mysql
```

停止 mysql 服务

```
xcopy /E /Y D:\MySQL57\data\testplatform\* D:\MySQL57\backup\%date:~0,4%%date:~5,2%%date:~8,2%\
```

`xcopy /E /Y` 源文件路径 目标目录路径

`D:\MySQL57\data\testplatform*` 表示以 `testplatform` 命名数据库的所有数据文件

`D:\MySQL57\backup\%date:~0,4%%date:~5,2%%date:~8,2%\` 表示要复制到的目标目录, `%date:~0,4%%date:~5,2%%date:~8,2%` 表示当前日期,格式形如 `20180315`

`/E` 表示按目录结构循环复制, 包括空目录

`/Y` 表示如果文件存在, 则直接静默覆盖, 不需要询问

```
net start mysql
```

重启 `mysql` 服务