# 接口测试平台V2.0使用文档

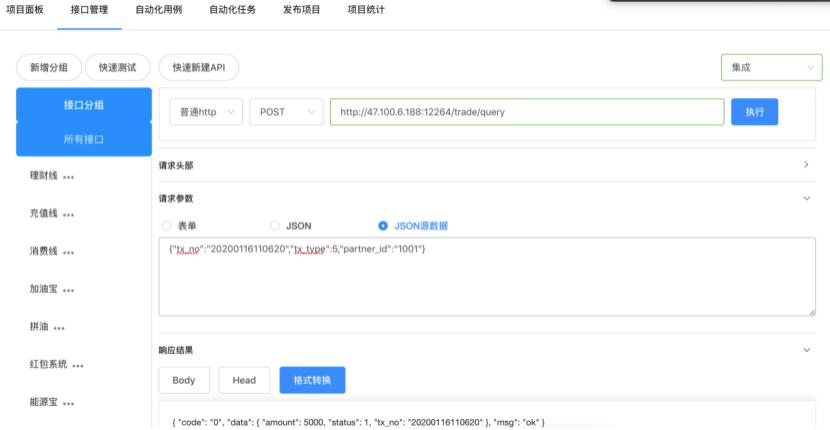
自动化用例的步骤可以添加以下命令：

|  |  |  |
| --- | --- | --- |
| 命令名称 | 功能  文本描述格式 | 参数 |
| 验证结果相等  assertResultEquals | 验证结果相等  验证结果locator是expected | expected：预期值  locator：json路径  如果返回结果是{"id":1,"data":{"name":"test"}}，其中name的locator是data.name，如果 json中含有数组，比如 {"id":1,"datas":[{"name":"test"},{"name":"test1"}]}，则第一个name的locator是 datas[0].name，第二个name是datas[1].name |
| 验证结果不相等  assertResultNotEquals | 验证结果不相等  验证结果locator不是expected | expected：预期不相等的值  locator：json路径 |
| 验证结果包含  assertResultContains | 验证结果包含  验证结果locator包含expected | expected：预期包含的值  locator：json路径 |
| 验证结果不包含  assertResultNotContains | 验证结果不包含  验证结果locator不包含expected | expected：预期不包含的值  locator：json路径 |
| 验证结果为空  assertResultBlank | 验证结果为空（null,空字符串）  验证结果locator为空 | locator：json路径 |
| 验证结果不为空  assertResultNotBlank | 验证结果不为空（null,空字符串）  验证结果locator不为空 | locator：json路径 |
| 验证表达式为真  assertTrue | 验证表达式为真  验证表达式eval为真 | eval：被断言的表达式  例如  ${data}>2 and ${data}<=10  '${data}'=='预期结果' |
| 暂弃用 验证结果满足表达式  assertResultTrue | 验证结果为真  验证结果表达式eval为真 | eval：预期为真的表达式，  如返回结果为{"msg": "", "code": "0", "data":""}，则表达式可以为：  ${code}==0 and '${msg}'=='' |
| 验证表达式不为真  assertNotTrue | 验证表达式不为真  验证表达式eval不为真 | eval：被断言的表达式 |
| 暂弃用 验证结果不满足表达式  assertResultNotTrue | 验证结果不为真  验证结果表达式eval不为真 | eval：预期不为真的表达式 |
| 验证结果符合正则表达式  assertResultMatch | 验证结果匹配正则表达式  验证结果locator匹配regexp | locator：json路径  regexp：正则表达式 |
| 保存数据  storeVariable | 保存数据到变量  保存数据value为参数name | value：数据  name：参数名称 |
| 保存结果  storeResultVariable | 保存结果到变量  保存结果locator为参数name | locator：json路径，如果要保存整个返回数据，不用填  name：参数名称 |
| 保存全局数据  storeGlobalVariable | 保存数据到全局变量，可以在不同的用例中进行访问。  保存数据value为全局参数name | value: 数据 name：全局参数名称 |
| 保存用例参数  storeAutomationVariable | 保存数据到用例参数，可以在下一次执行用例使用。  保存数据value为用例参数name | value: 数据 name：用例参数名  例子：  {"value": "##${id}+1##", "name": "id"}，把用例参数id加1保存到id参数中，下一次执行用例会使用加1后的id值 |
| 保存数组长度  storeArrayLen | 保存结果数组长度  保存结果数组locator长度为参数name | locator：json路径，如结果为{"id":1,"data":[{"name":"test"},{"name":"test1"}]}，locator为data  name：变量名 |
| 保存正则结果  storeResultByRegex | 通过正则表达式从结果提取值并保存  保存结果正则regex为参数name | regex：正则表达式，如server=(.+?)&  name：变量名 |
| 执行sql  runSql | 执行sql语句  使用用户名username和密码password,连接数据库host:port名称为database,执行sql | host：数据库ip  port: 数据库端口，默认3306  database: 数据库名称  username：数据库账号  password：数据库密码  sql：执行的sql语句 |
| 执行表达式  runEval | 执行表达式  执行表达式eval名称为name | eval: python语句，多行用;分隔，如果为空取已保存的名称为name的语句。最后一行语句可以执行赋值语句，保存数据到变量中，后续可以通过${变量名}获取保存的数据  例如：  from datetime import datetime;trace=datetime.now().strftime('%Y%m%d%H%M%S%f')  该语句保存当前时间戳到变量名trace，后续可以使用${trace}获取该时间戳  name: 名称，可以把语句保存到该名称 |
| 执行ssh  runSSH | 执行ssh命令  使用用户名username和密码password,连接服务器host:port,执行command | host：服务器host  port: ssh端口，默认22  username：账号  password：密码  command：执行的命令 |
| 等待  sleep | 等待指定时间（秒）  等待seconds秒 | seconds：等待秒数 |
| 打印数据  print | 输出数据  输出value | value：要输出的数据 |
| 执行Get  httpGet | 发送HTTP get请求  get请求url | url：请求url |
| 执行Post  httpPost | 发送HTTP POST请求  post请求url数据为data | url：请求url  data：请求数据 |
| 执行失败  fail | 强制用例失败  执行失败提示message | message：异常信息 |
| IF,ELSE,END\_IF | 如果条件语句为真，执行IF后面的步骤，否则执行ELSE后面的步骤，或者不执行。 | eval：条件判断语句 |
| FOR,END\_FOR | 循环以下步骤指定次数，循环的index存放在for\_index变量 | count:循环次数 |
| LOOP,END\_LOOP | 根据指定的数组循环 | array:json格式循环数组 |
| WHILE,END\_WHILE | 如果条件为真则循环以下步骤 | eval:条件判断语句 |
| CONTINUE | 跳过当前循环的步骤,进入下一个循环 |  |
| BREAK | 跳出循环 |  |

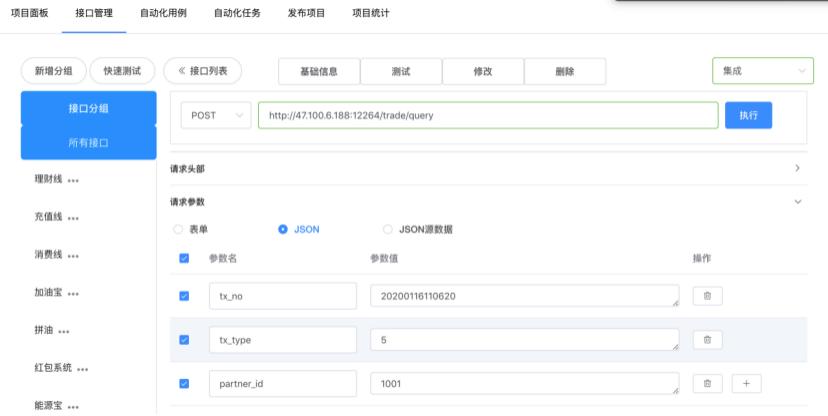
基本功能介绍：

一、接口管理：

1、快速测试

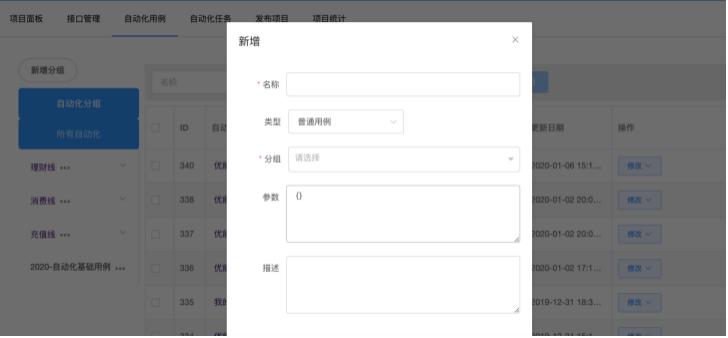


2、接口测试：

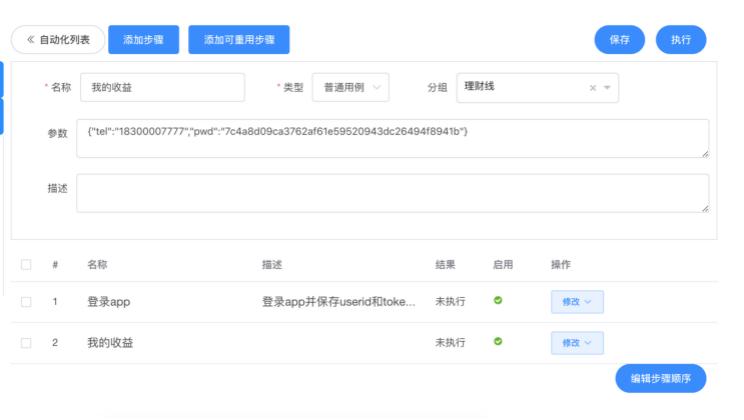


二、自动化用例：

1、新增用例：



2、修改用例：



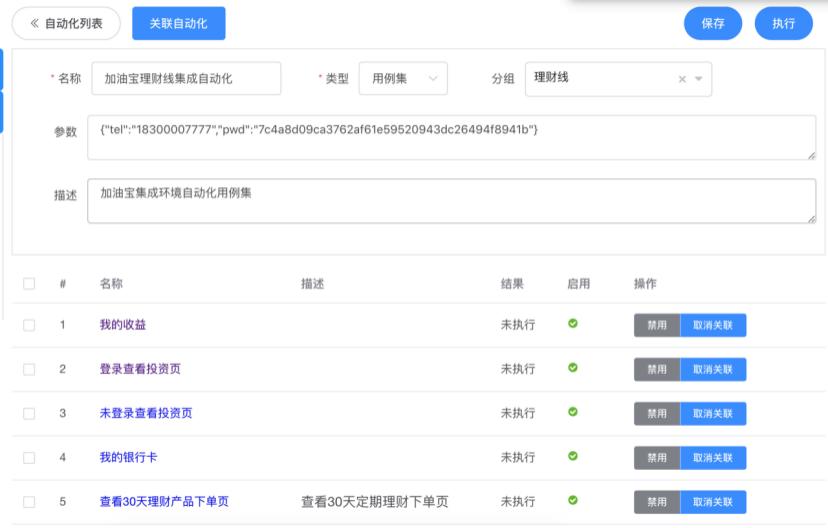


3、执行用例：

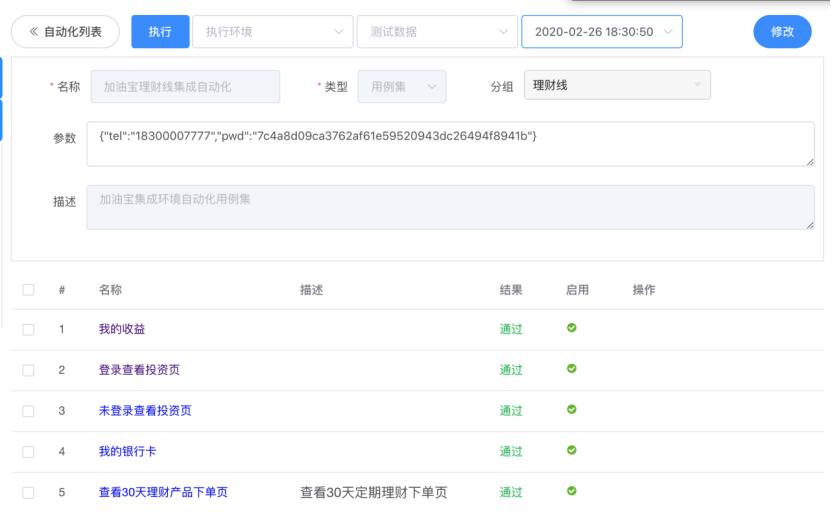


三、用例集：

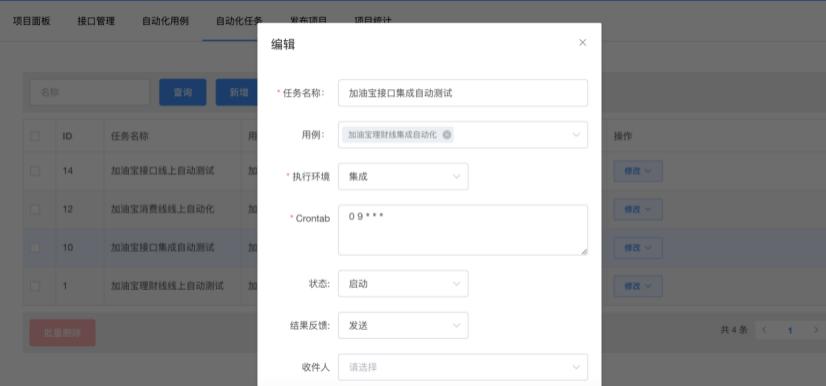
1、修改用例集：



2、执行用例集：



四、自动化任务：





五、发布项目：

