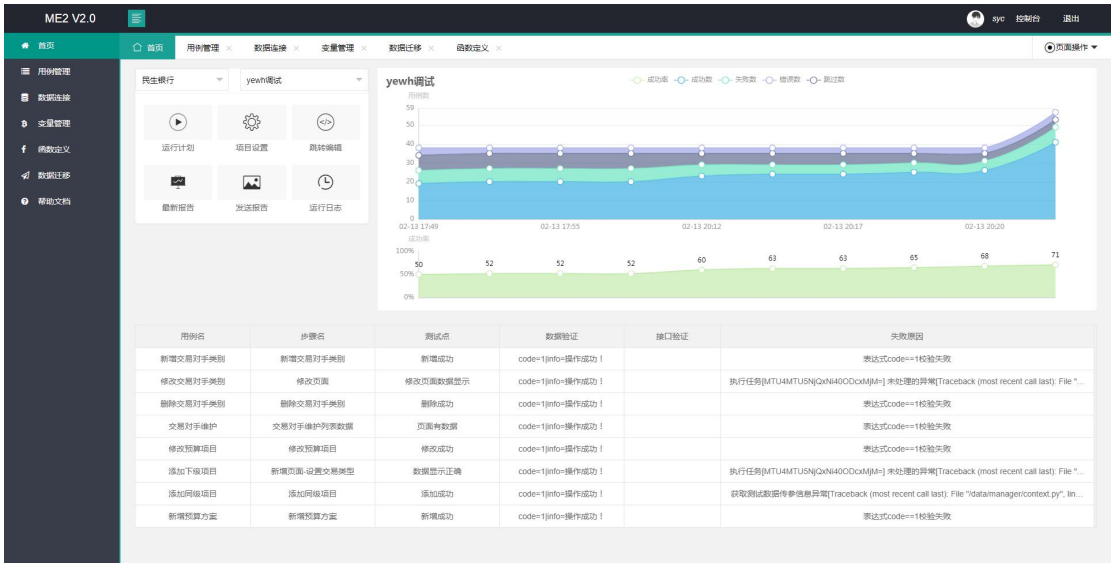
ME2 自动化框架使用说明书

目录

1. 首页

1.1 运行计划

操作：首页中，选择产品项目-对应计划后，点击【运行计划】后开始执行，可点击【运行日志】查看执行过程中产生日志。
注意：运行计划相比树级中的执行存在不同，此处正式点击【运行计划】会记录执行通过率和记录执行产生缺陷，树级中的“调试执行”则不会记录数据，此处【运行计划】会用于度量产品项目的质量问题。因此推荐在树级中完成调试后，再在此处正式执行，需要结合 jenkins 做一键集成部署自动化归回也建议用此处执行触发。



1.2 项目设置

操作：点击首页【项目设置】，可添加邮箱配置，钉钉群执行完成机器人提醒配置，jenkins jacoco 配置。

1.3 自动化测试报告

操作：在使用 1.1 功能运行计划成功后，可以点击【最新报告】查看最新执行结果生成的测试报告，再点击【发送报告】，对 1.2 功能项目设置中的设置好的邮箱发送测试报告。（也可以设定为自动发送）
注意：在自动化测试用例未执行结束之前，不能发送测试报告。

1.4 图形统计

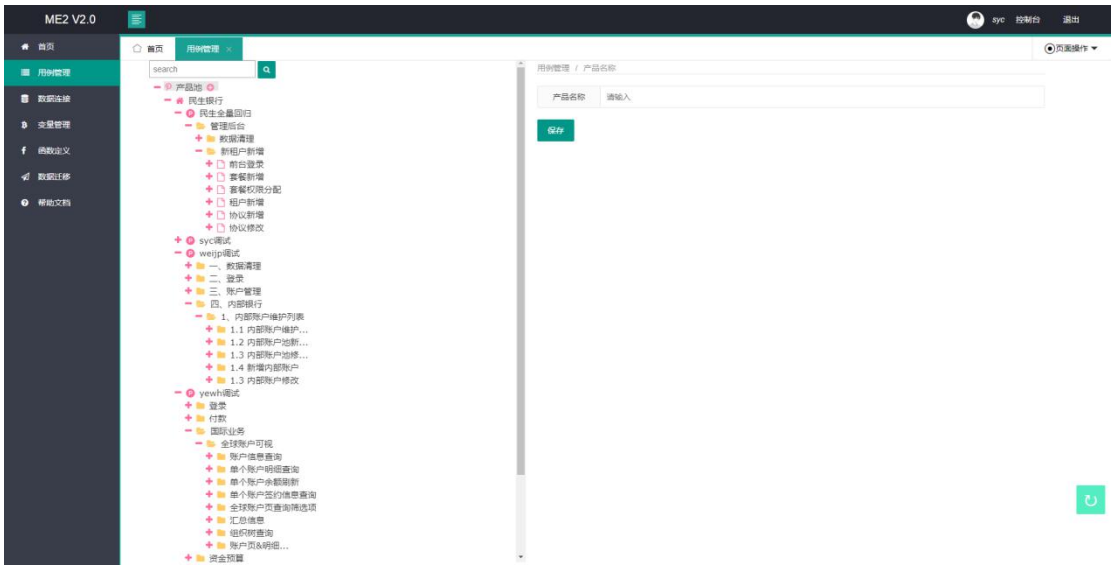
操作：首页中，选择产品项目-对应计划后，点击【运行计划】后开始执行，执行结束后不论成功失败，首页图形统计会记录**历次**自动化测试用例执行的，**用例数、成功率、具体失败用例明细。**

2. 用例管理

2.1 树级菜单层级

2.1.1 产品层

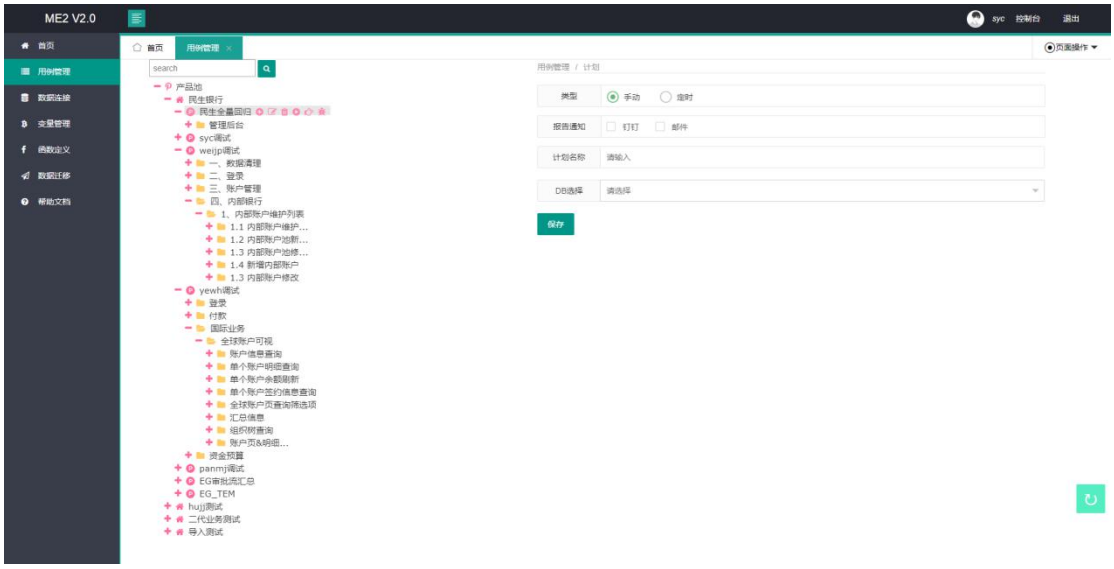
意义：【产品池】根级目录为树级菜单无法编辑层级，产品层级则为根级目录下的第一层。
操作：点击根目录下【产品池】右侧“+”号，添加新的产品目录层级，可以根据需要命名如：“民生银行”，“爱心人寿”，“融汇直联”等等。



2.1.2 计划层

意义：定制自己的执行计划，同个产品也可以按照大家各自的自动化策略建立多个不同的自动化计划，如“XX 产品全量回归”计划，“XX 产品支付结算流程”计划，“XX 产品审批工作流”计划等等。总而言之，一个计划可以是所有用例集的集合，也可以是几个用例集的组合。
操作：点击【产品层】右侧“+”号，可选类型【手动】执行或者【定时】执行，报告通知通过【钉钉】提醒或【邮件】提醒，输入【计划名称】保存。
注意：此处数据库选择【DB 选择】，一般情况可以不选，整个树级菜单中存在多出可以选择 DB 数据库的选项，规则是**从最下级开始判断，下级覆盖上级，下级未选择的情况下按上**

级已选 DB 选择为准。



2.1.3 用例层

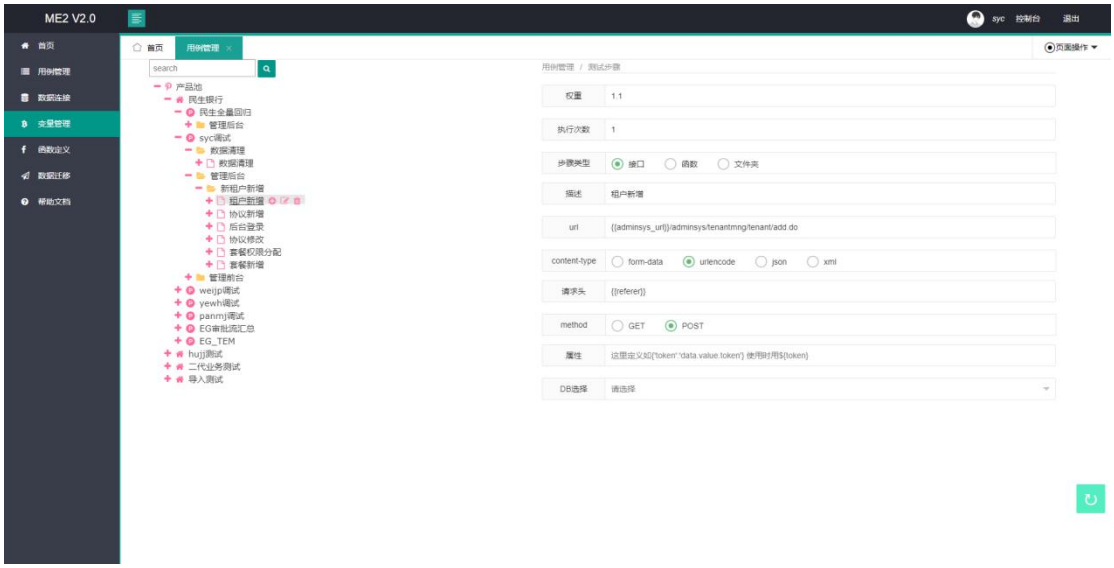
意义：文件夹形式自动化测试用例管理，用例层可以建多层，便于归类整理用例。
操作：点击【计划层】右侧“+”号，输入用例文件夹名称，【DB 选择】填写可选。
注意：与文件夹同一层级只能存在文件夹，暂时不支持用例文件夹与计划、接口、数据存在同一层，否则执行会出错。

2.1.4 接口层

意义：接口相关配置参数填写。
【权重】表示接口编号顺序，同一个文件下下的接口按此顺序执行。
【执行次数】表示需要此接口执行的次数，不需要执行时可以填“0”。
【步骤类型】改变类型，选择接口、函数、文件夹分别表示添加不同类型的层级，函数可以与接口统一层级，用例文件夹则不允许，接口和函数必须在用例文件夹下级。
【url】接口请求地址。
【content-type】选择接口请求数据格式，目前支持 form-data，urlencode，json，xml。
【请求头】目前系统存在默认请求头，此处可以根据各条线不同请求头格式直接添加。
如已有默认请求头：
{'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.109 Safari/537.36', 'Content-Type': 'application/x-www-form-urlencoded;charset=UTF-8'}
想添加部分，直接在【请求头】处填写：
{'Referer':'http://adminsysnew.tfp-cmbc-mysql-uat.k8s.fingard.cn/adminsys/systemindex/passwordloginpage.do'}
即可以生成：
{'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/59.0.3071.109 Safari/537.36', 'Content-Type':

'application/x-www-form-urlencoded;charset=UTF-8', 'Referer': 'http://adminsysnew.tfp-cmbc-mysql-uat.k8s.fingard.cn/adminsys/systemindex/passwordloginpage.do'}

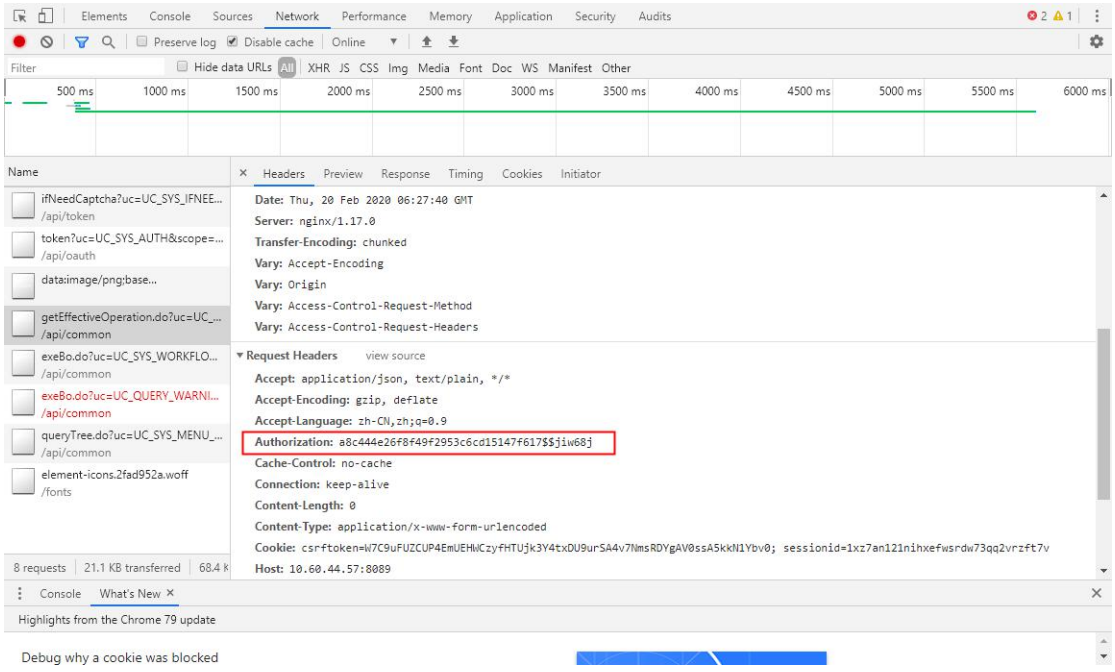
【method】GET or　POST，get 请求参数字段支持 json 格式。



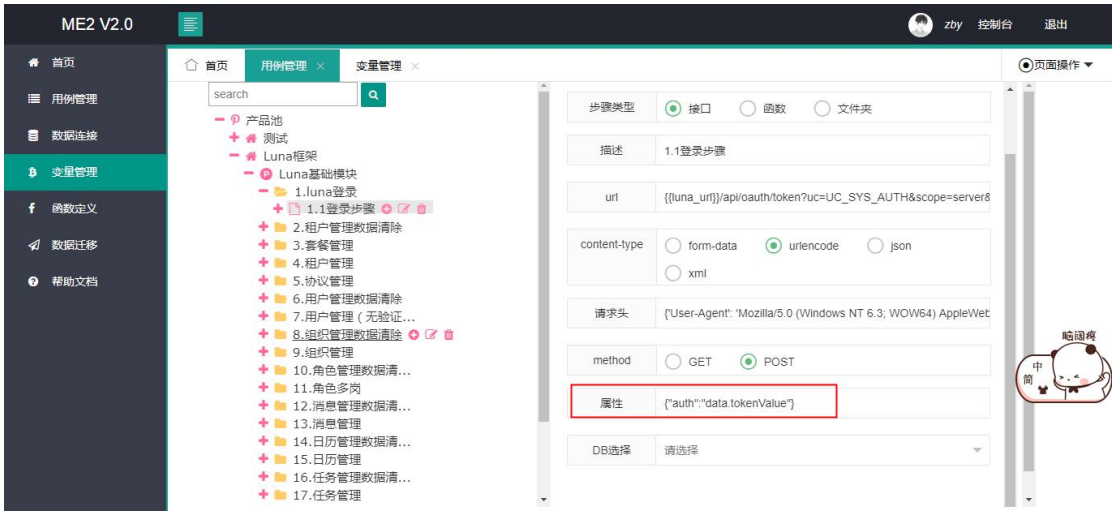
【属性】：定义此接口返回参数的某个数据为一个属性进行使用，属性只在计划运行期间有效，不能跨计划使用。

举例：

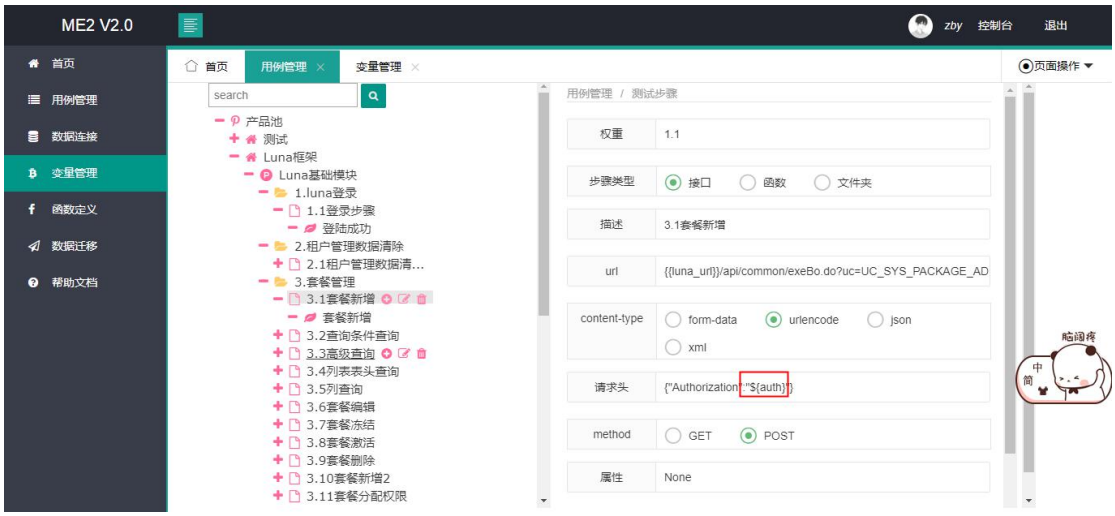
某系统请求头部需要增加如下字段：



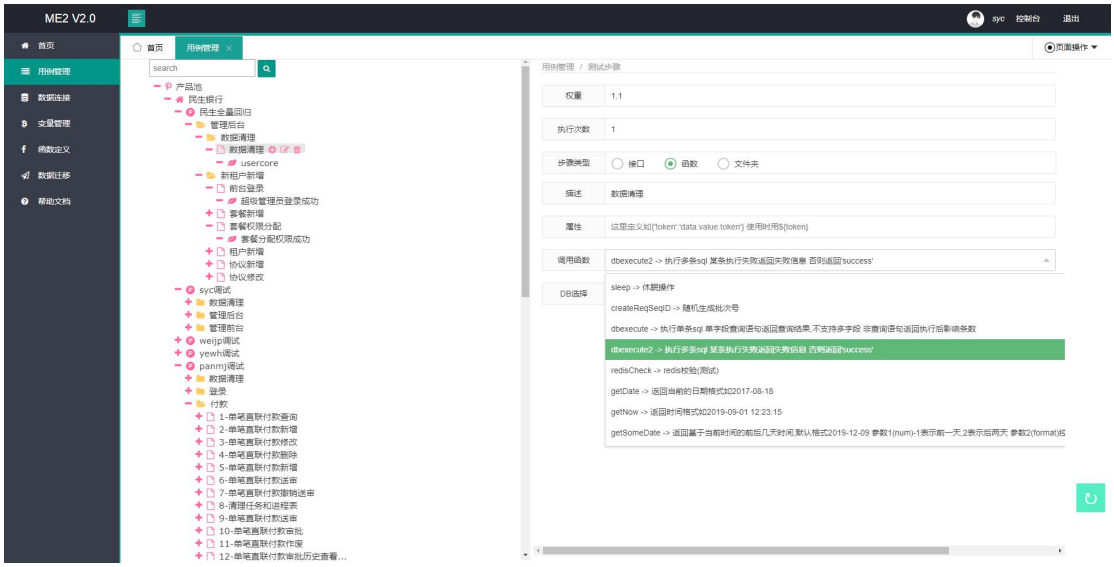
需要在登录接口【接口层】中如下定义属性：



这里表示登录接口成功返回时，取字段名称为 auth 的值定义为一个属性，属性在每次执行登录步骤是动态获取，并可以给在同个计划内的后续步骤使用。
其他步骤使用时如下，即达到例子中第一张图中新增头部字段要求：



当【步骤类型】选择为【函数】时：
需要选择【调用函数】：【调用函数】为已经维护存在与函数定义模块中的可用函数。



2.1.5 数据层

意义：此层级为接口入参数据，接口的和数据的关系**支持一对多**，不同的数据即代表不同的业务场景，因此推荐**以业务场景名称命名**此层级。

【权重】表示数据编号顺序，同一个文件下下的数据按此顺序执行。

【执行次数】表示需要此数据执行的次数，不需要执行时可以填“0”。

【前置操作】在接口执行之前，执行函数。

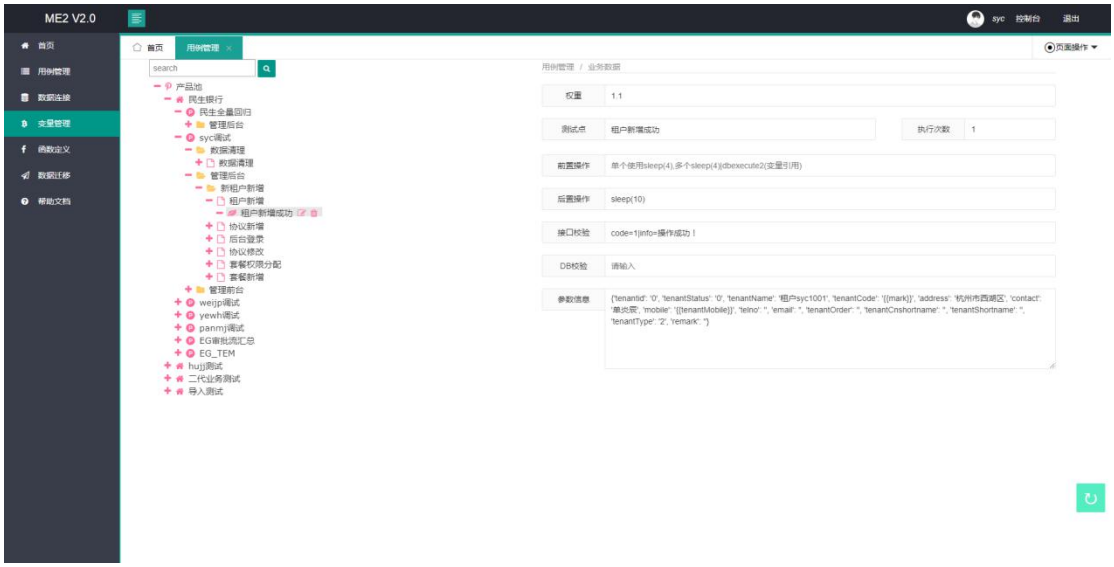
【后置操作】在接口执行之后下个接口执行之前，执行函数。

【接口校验】设定“期望值”与接口返回参数进行匹配校验。

注意：一些字符冲突，比如“|”字符。在接口和数据校验时，如果返回信息包含“|”则不能直接使用等式校验，避免和批量校验分隔符冲突，可用“$”代替。

【DB 校验】设定“期望值”与数据库结果进行匹配校验

【参数信息】填写传参的地方，支持 json，xml 等多种格式，也有根据业务线需求进行格式的简化如：a=1&b=2 这样的格式，简化了大家参数信息填写的步骤。（比较多的抓包工具中都是类似这种可以，可以直接抓包以后复制黏贴）



2.2 常见符号和使用方法（可扩展）

2.2.1 分隔符“|”

用于分隔多个数据时使用。
例如**多个函数之间**：sleep（5）|dbexecute2
多个**接口校验之间**：code=1|info=操作成功！

2.2.2 模糊查询“$”

用于**接口返回数据模糊查询**：“info$共生成了上划单 1 笔”，表示返回字段“info”包含“共生成了上划单 1 笔”内容。
response.text$XXX 用法：用于一些非 json 格式中返回数据的模糊查询校验。
举例：返回 html 页面中含有登录成功字样
则在接口校验处用“response.text$登录成功”，校验 html 页面返回证明登录结果已经成功。

2.2.3 响应头校验

格式用法：response.heard.XXX
如我们想校验响应头中的 Content-Length 字段，可以描述成：
response.heard.content_Length=354（注意此处表达式中需要全部小写和“-”字符的替换为“_”）

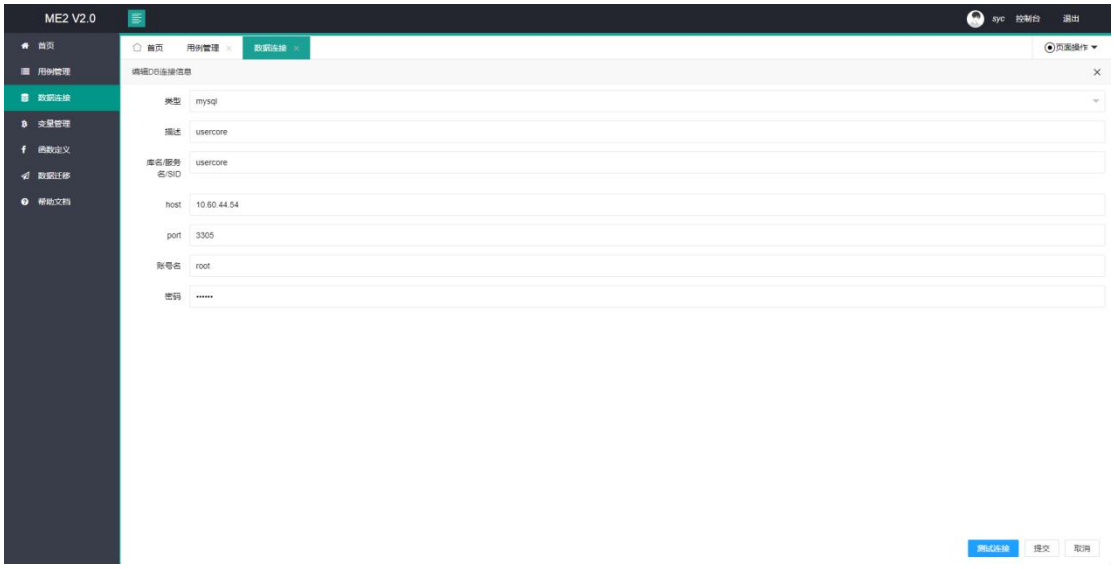
2.2.4 多 sql 时的跨库查询

在执行一个参数框下的 sql 时，如果也需要几个 sql 分别查询不同的数据库，可以如下填写：Sql1@库 1;sql2@库 2

3. 数据连接

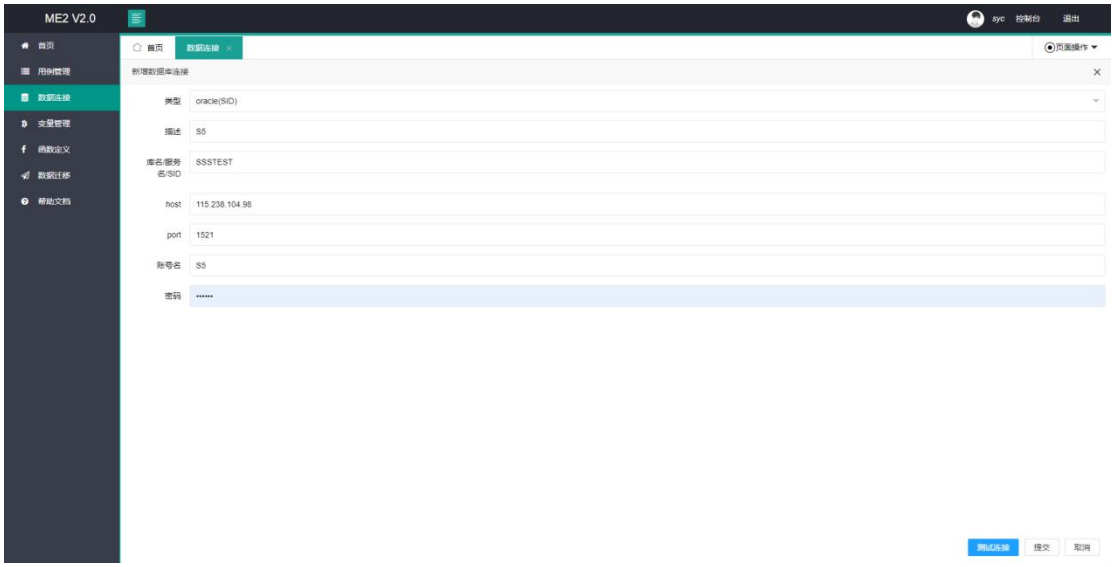
3.1Mysql 数据库连接配置

类似 mysql 连接工具，如下图：



3.2Oracle 数据库连接配置

类似 oracle 连接工具，如下图：



4. 变量管理

4.1 变量的引用

假设我们在变量管理中维护一个键名为 mark，mark 即为一个变量名称。当我们在系统中想要引用这个变量的时候，我们统一引用格式为：**{{mark}}**。

举例 1：

我们维护 mark 的值为 sycauto_1001

那么我们想在数据层接口参数中引用这个 sycauto_1001 的值作为'tenantCode'的值时，我们就可以按下面这样填写：

{'tenantid': '0', 'tenantStatus': '0', 'tenantName': '租户 syc1001', 'tenantCode': '{{mark}}', 'address': '杭州市西湖区', 'contact': '单炎辰', 'mobile': '15800000000', 'telno': '', 'email': '', 'tenantOrder': '', 'tenantCnshortname': '', 'tenantShortname': '', 'tenantType': '2', 'remark': ''}

那么在参数实际发送过程中，实际数据就会替换为：

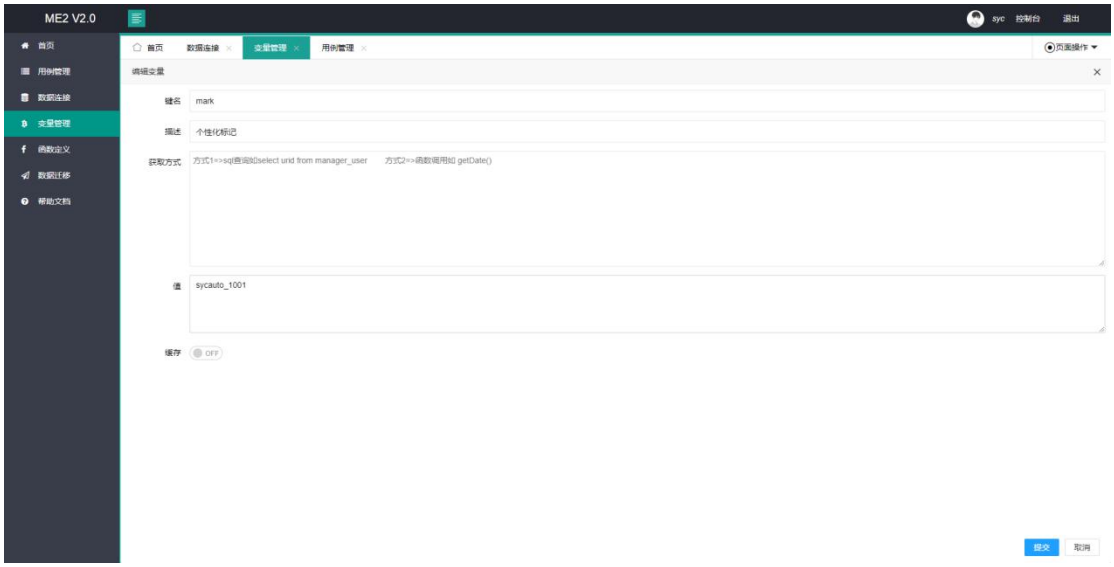
{'tenantid': '0', 'tenantStatus': '0', 'tenantName': '租户 syc1001', 'tenantCode': 'sycauto_1001', 'address': '杭州市西湖区', 'contact': '单炎辰', 'mobile': '15800000000', 'telno': '', 'email': '', 'tenantOrder': '', 'tenantCnshortname': '', 'tenantShortname': '', 'tenantType': '2', 'remark': ''}

此为变量最基本的引用，用于全局替换一些参数，在实际运用中避免想做数据的替换需要每一个数据一条条修改，只需要在全局变量处修改即可全局生效。

4.2 变量的类型

4.2.1 固定变量

意义：即变量值直接为一个固定值，一般为了方便全局修改而把一个固定值作为变量值。
如图：



4.2.2 动态变量

意义：一般需要获取方式间接获取的变量值，变量本身没有固定值是根据条件动态生成的。
获取方式主要为**两种，sql 获取和函数获取：**
Sql 获取方式：在获取方式中填写 sql，sql 末尾需要加上关联数据库。
举例：
SELECT CONTRACT_ID FROM tsys_tenant_contract WHERE TENANT_ID IN(SELECT TENANT_ID FROM tsys_tenant WHERE TENANT_CODE='sycauto_1001')@usercore
上面举例 sql 获取方式是为了获取变量 CONTRACT_ID
函数获取方式：函数方式比较简单，直接在获取方式中输入函数定义中已经存在的函数。
举例：
getDate
getDate 是函数定义中已经维护好的一个随机生成日期的函数

5. 函数定义

5.1 函数

意义：可以通过 Python 代码，简单定义一些功能或者数据的实现。
举例 1**sleep** 函数：
代码定义：

```
def sleep(sec):
   import time
   time.sleep(sec)
```

举例 2**creatReqSeqID** 函数：
代码定义：

```
def createReqSeqID(name=''):
      import time,random
      n=20-len(name)
      if n>=14:
            nowtime=time.strftime("%Y%m%d%H%M%S")
      else:
            nowtime=time.strftime("%m%d%H%M%S")
            if n <10:
                  nowtime=nowtime[:n]
      n=n-len(nowtime)
      return name+nowtime+''.join(random.choice("0123456789") for i in range(n))
```

注意：定义的函数目前最好仅限于使用 Python 自带库，如定义休眠函数时 import time 可能使用到 time 库，它是 Python 自带的。

6. 数据迁移

6.1 数据导入导出

支持新老版本数据通用格式的导入导出

7. 用户管理

7.1 用户新增

通过管理员用户 admin，可以新建系统用户和设置密码。
推荐每个人使用都自建一个账户，权限分离功能会根据用户创建人去区分权限。

7.2 用户权限

目前系统暂时没有用户权限隔离，但是一些变量、函数、用例都存在创建人概念，后续会根据创建用户进行权限隔离，实现本用户创建数据其他用户在没有权限的情况下不能删除、修改、移动等一系列权限限制。