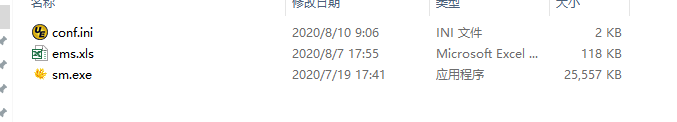
1. 项目包含3个部分，客户端、用例文件、配置文件，用例文件和配置文件需要在同一目录下，配置文件名称不可更改，如果使用客户端加载用例文件时报初始化conf.ini错误，且之前打开或修改过配置文件，可能是由于文本编辑器的编码格式引起的，建议使用UE编辑配置文件。



1. 用例编写

假设有一个接口，请求参数是：  
{“username”:”name”} #此处要用双引号

接口响应是：

[

{

"name": "菜鸟教程",

"url": "www.runoob.com"

},

{

"name": "google",

"url": "www.google.com"

},

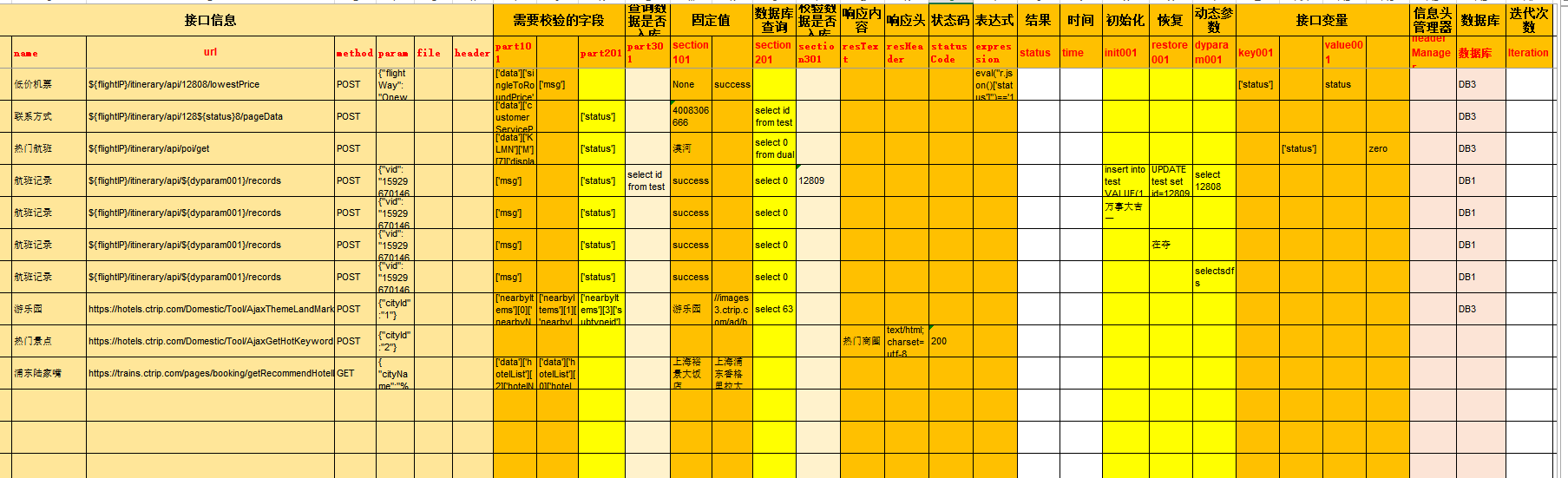
{

"name": "微博",

"url": "www.weibo.com"

}

]



首先需要填写接口信息，

然后填写需要校验的数据及相应的预期结果，part101~part301是要校验的数据，section101~section301是预期结果

这部分分为3种不同的校验方式，详情如下：

Part101~section101：固定值校验

Part101填写['sites'][0]['name']

Section101填写”菜鸟教程”

Part201~section201:数据库校验

Part101依然填写['sites'][0]['name']

这时候Section201填写的不再是一个固定的值，而是从数据库中查询出的数据，可以这么写：select菜鸟教程

需要注意的时候，如果用例中需要写sql语句（不仅此处），那么首先需要选择数据库

Part301~section301:校验数据是否入库

有时候在接口请求之后要校验数据是不是真的入库了就可以使用这部分校验

Part301填写sql语句

Section301填写预期结果

如果接口响应层次比较深，可以使用“JSON解析”功能把接口响应解析为key-value形式。

以上part101~section301部分为精确校验

如果不想进行精确校验，只想看某些值是不是包含在接口响应中，则可以使用响应断言resText,同理，响应头断言resHeader也是模糊匹配

以上校验通常可以满足大多数场景，但有些场景不能很好的满足，如，我可能想校验响应中的列表的长度，或者某个字段是不是手机号，如此种种，这时候需要使用表达式进行校验，这里的表达式其实就是一句python代码，如我想校验上面给出的接口响应中的列表长度则可以这么写：eval("r.json()['sites']")==3，此处需注意：程序中的接口响应对象是“r”

数据库初始化、数据恢复、动态参数：

这三部分都是sql语句

顾名思义，数据初始化是在接口请求之前对数据库进行的初始化操作，数据恢复是在接口请求之后对数据库进行的恢复操作

动态参数最初则是基于这样的考虑，有些接口的请求参数不是固定的，那么我们就需要从数据库中实时获取，假如参数ID是自增的，我们实时从数据库中查询出最新的ID赋给参数列表，可以这么写：select ID from XX where XX,参数列表中使用这个查询出的ID需要这么写：{“ID”:”${ dyparam001 }”}，当然动态参数不仅仅用于自增参数，任何从数据库中取出的数据都可以用作动态参数，如使用sql随机一个手机号或者调用md5函数对某个值进行加密。

${变量名}，这个是本项目中变量的引用格式，项目中存在3种变量，用户变量，动态参数和接口变量，都使用相同的引用格式。

接口变量：

很多情况下接口之间会存在依赖关系，如其他接口依赖于登录接口，这时候就要设置接口变量。如B接口依赖A接口中的某个[‘token’]字段，则可以在A接口的key001填写[‘token’]在value001中填写token，key001是依赖的字段，value001是给这个字段设置的变量名，在B接口中就可以用${token}的形式来调用这个变量

信息头管理器：

信息头管理器可供后续接口隐式调用，还是以AB两个接口为例，如果B接口请求时需要带请求头，当然可以在B的header中直接填写，但如果B之后的接口CDEF……都需要使用相同的请求头，那么这时候就可以用到信息头管理器了，可以在A的信息头管理器中填写{“token”:”${token}”}，如果B中填写了请求头，那么信息头管理器在B接口中就不起作用了。

数据库：

数据库信息是在配置文件中配置的，支持3种数据库，oracle,sqlserver.mysql

项目中预期了3个数据库：DB1,DB2,DB3，也可以在配置文件中自定义数据库，使用${}形式调用即可

迭代次数：

需要注意的是迭代次数为0即该条用例不执行