

[接口测试用例-编写指导手册]

使用 Excel 完成 Http 协议的单或多接口自动化操作

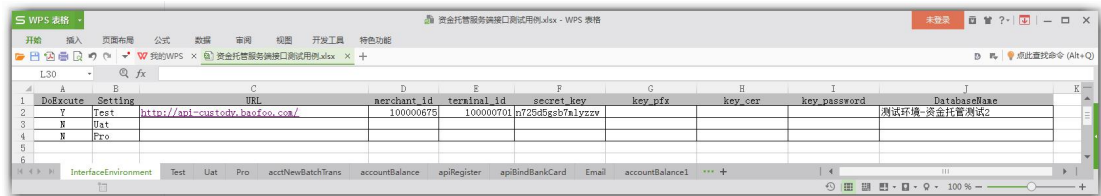
[宝付网络(上海)科技有限公司]
[上海市 浦东新区 居里路 99 号]
[17097159420]
[Early@baofoo.com]
2016 年 8 月 31 日

目 录

第一章：基础入门了解篇.....	3
1.接口测试环境介绍.....	3
2.接口测试集介绍.....	3
3.接口测试用例介绍.....	4
第二章：接口测试用例编写--基本编写.....	5
1.请求参数下层模板的讲解：	5
2.响应参数的讲解：	6
第三章：接口测试用例编写--用例编写进阶.....	7

第一章：基础入门了解篇

1.接口测试环境介绍



表名: **InterfaceEnvironment**, 这个表可以设置接口执行的环境, 当前文档只展示了测试环境的接口。

列名: **DoExcute**: 指定是否执行当前环境, Y: 执行, N 不执行。

列名: **Setring**: 指定了当前环境的测试集的存储位置。

列名: **URL**: 存放的是当前环境的域名。

列名: **merchant_id**: 存放当前商户的商户号。

列名: **terminal_id**: 存放当前商户的终端号。

列名: **secret_key**: 存放当前商户的 MD5 秘钥。

列名: **key_pfx**: 存放当前商户的公钥文件名称。

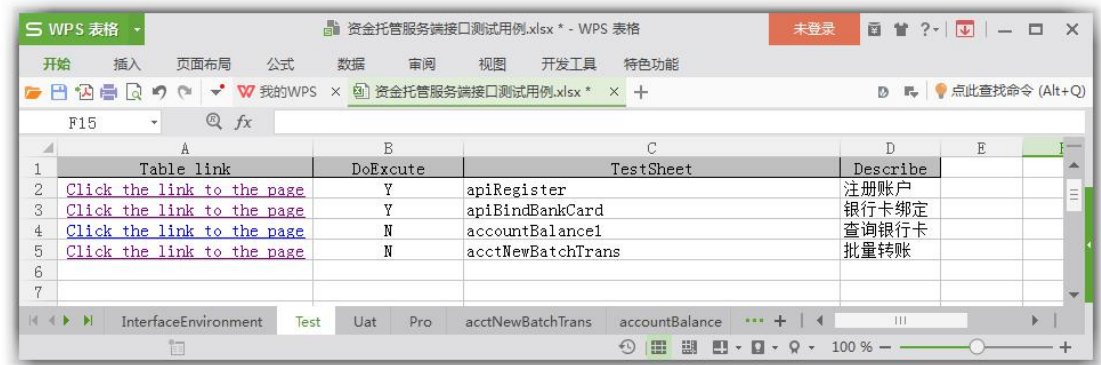
列名: **key_cer**: 存放当前商户的私钥文件名称。

列名: **key_password**: 存放当前商户的私钥文件密码。

列名: **DatabaseName**: 目前无用, 后续扩展数据库使用。

以上就接口测试环境的介绍, 基本的参数, 找开发要来填充即可。

2.接口测试集介绍



表名: **Test**: 测试环境的测试集, 对应之前测试环境的 **Setting** 指定的测试集 (准生产环境: Uat, 生产环境: Pro)

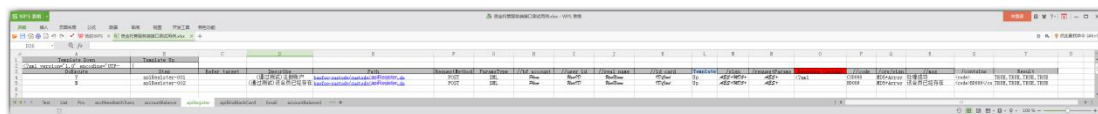
列名: **Table link**: 超链接, 用于接口测试用例过多时快速跳转至对应测试用例。

列名: **DoExcute**: 与测试环境的 **DoExcute** 相同, 表示当前测试用例是否执行。

列名: **TestSheet**: 接口测试用例的存放表, 内容与表名相同, 表示用例在对应表单。 (注: 一定需要是一致的。)

列名: **Describe**: 描述。描述当前接口的作用, 查看日志是可用于参考。

3.接口测试用例介绍



先来个大致介绍：

表名：**apiRegister**：这是一个注册接口。

列名：**Template_Down**：下层接口模板。

列名：**Template_Up**：上层接口模板。

先介绍下这两个接口模板吧。

由于公司的接口参数基本都是：一层键值对中包含一个 XML 格式的参数，但是由于 XML 格式不支持有 2 个头文件，导致程序解析时无法解析，所以就将参数分为 2 层。（注：Json 格式可以包含，但资金托管暂不支持 json）

情况就是这样，下面可以看一下模板信息：

Template_Down	Template_Up
<pre><?xml version='1.0' encoding='UTF-8'?> <custody_req> <bf_account>15000918908</bf_account> <user_id>20140375</user_id> <real_name>李清照</real_name> <id_card>32030019800918181X</id_card> <account_type>1</account_type> </custody_req></pre>	

详细情况，可参看【宝付资金托管商户接口文档(房天下定制版).doc】里面的 apiRegister 接口。

继续...

列名：**DoExcute**：表示是否执行，Y：执行，N 不执行。

列名：**Step**：步骤名称，与测试用例一样，必须唯一。

列名：**Refer target**：目标测试步骤，可不填，进阶多接口测试用例编写可讲。

列名：**Describe**：描述，描述当前测试步骤的作用。

列名：**Path**：路径，存放 Http 接口域名以后的路径，用于请求。

列名：**RequestMethod**：接口请求方式：POST，目前只支持 POST，后续可扩展。

列名：**ParamsType**：下层参数模板请求时的类型，支持 Json、Xml

列名：.... 下层参数的参数操作

列名：**Template**：切换模板，从下层模板的切换至上层模板。

列名：.... 上层模板的参数操作

列名：**Response_Content**：不用填写值，系统会写入响应报文并切换至响应报文

列名：.... 响应报文的断言操作

列名：**Result**：最终断言，判断当前测试用例是否执行成功。

接口测试用例介绍结束，现在进入学习编写接口测试用例吧。

第二章：接口测试用例编写---基本编写

```
{
  "custody_req": {
    "A": [
      "a",
      "a",
      "a",
      "a"
    ],
    "test": "123",
    "num": [
      {
        "A": [
          "a",
          "a",
          "a",
          "a",
          "a",
          "a"
        ],
        "B": [
          "b",
          "b1",
          "b",
          "b",
          "b",
          "b"
        ]
      }
    ],
    "B": "b"
  },
  {
    "C": "c",
    "D": "d"
  },
  {
    "E": "e",
    "F": "f"
  }
],
"tester": {
  "TEST": "123",
  "t0": {
    "TEST": "123"
  }
}
}
```

←Json格式的参数字模板

↓Xml格式的参数字模板

```
<?xml version="1.0" encoding="UTF-8" ?>
<custody_req>
  <A>a</A>
  <A>a</A>
  <A>a</A>
  <A>a</A>
  <num>
    <A>a</A>
    <A>a</A>
    <A>a</A>
    <A>a</A>
    <A>a</A>
    <A>a</A>
    <B>b</B>
    <B>b1</B>
    <B>b</B>
    <B>b</B>
    <B>b</B>
    <B>b</B>
  </num>
  <B>b</B>
</custody_req>
</num>
<C>c</C>
<D>d</D>
</num>
<test>123</test>
<tester>
  <TEST>123</TEST>
  <t0>
    <TEST>123</TEST>
  </t0>
</tester>
<num>
  <E>e</E>
  <F>f</F>
</num>
</custody_req>
```

讲解的问题是：如何寻址？

模板参数寻址要点：
/ : 表示绝对路径
// : 表示相对路径
[] : 此为数值存放时使用，例如[1]

注：1. 所有下标数值都是从0开始数
2. Xml与Json的寻址方式已统一，便于编写寻址路径

下面是栗子：别客气，多嚼嚼。

1. 编写寻址值为：b1 的绝对路径
路径：/custody_req/num/*[0]/A/*[5]/B/*[1]
结果：b1

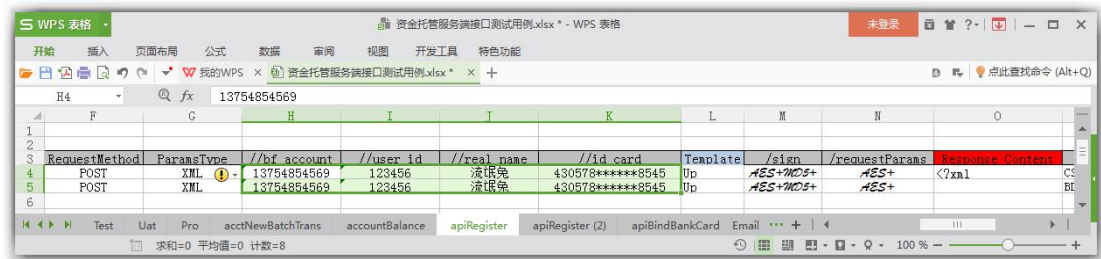
2. 编写寻址值为：b1 的相对路径
1) 路径：//custody_req/num//B/*[1]
2) 路径：//num/*[0]/A/*[5]/B/*[1]
1) 结果：b1
2) 结果：b1

3. 编写寻址值为：c 的相对路径
1) 路径：//custody_req/C
2) 路径：//custody_req//C
3) 路径：//C
1) 结果：c
2) 结果：c
3) 结果：c

4. 编写寻址值为：123 的相对路径
1) 路径：//TEST
2) 路径：//t0/TEST
1) 结果：123, 123
2) 结果：123

根据以上的路径寻址，相信大家都知道如何进行寻址，也知道了 Excel 的列名带有//或/就知道是做什么用的了吧？不知道？好吧，是用于寻址。

1.请求参数下层模板的讲解：



上图，被选中的单元格中的值为要替换的值。例如：

- 列名：//bf_account ：将下层模板中指定的值替换成：13754854569
- 列名：//user_id ：将下层模板中指定的值替换成：123456

列名: **//real_name** : 将下层模板中指定的值替换成: 流氓兔

列名: **//id_card** : 将下层模板中指定的值替换成: 430578*****8545

一、请求参数上层模板的讲解:

列名: **Template** 之后就是切换成上层模板了。



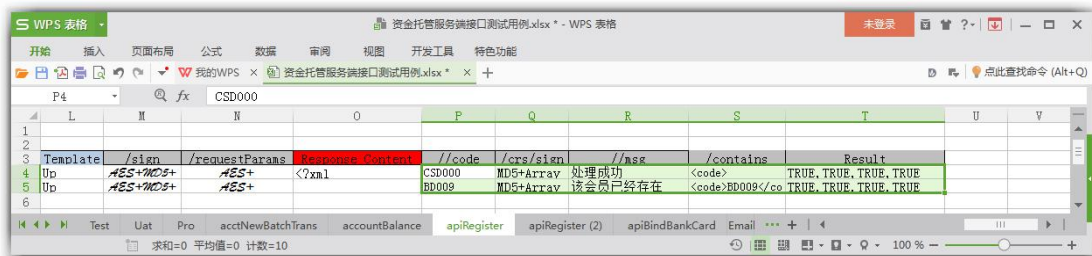
列名: **/sign** : 上层模板对下层模板的操作, 如 AES+MD5+

列名: **/requestParams** : 上层模板对下层模板的操作, 如 AES+
其实将下层模板参数加密后, 再替换给指定的值。

详细加密关键字, 可以了解作者 blog:[接口测试用例编写指南](#)

2. 响应参数的讲解:

列名: **Response_Content** 之后就是切换成响应报文模板了。



列名: **//code**: 断言报文中 code 值为 “CSD000”。

列名: **/crs/sign**: 断言报文中 sign 值为 MD5+Array 方式加密后的内容。

列名: **//msg**: 断言报文中 msg 值为 “处理成功”。

列名: **/contains**: 断言报文中包含值为 “<msg>处理成功</msg>”

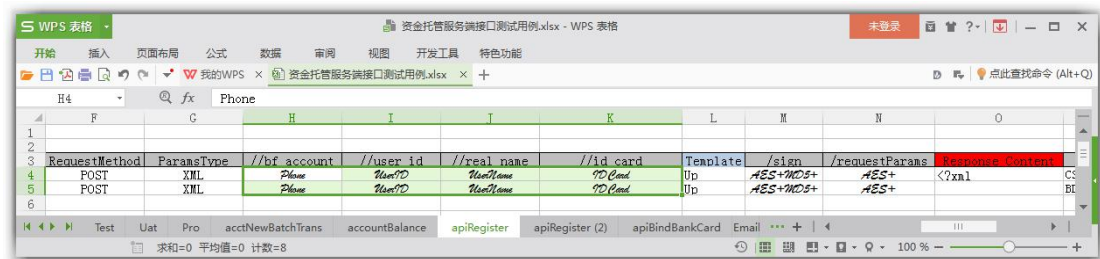
以上响应报文的断言, 正确返回: **TURE**, 错误返回: **FALSE**

列名: **Result**: 最终断言! 校验以上报文的实际执行结果与预期结果是否一致, 一致则测试通过, 否则测试失败。

注: 请求报文、响应报文的单元格数量并不限制, 可以根据测试用例的想法, 字段的长短以及不填的效果来组装测试用例。

以上便是接口测试用例的编写入门! 你会了吗? 会了就感觉编写一个, 毕竟书读百遍, 不如手写一篇!

第三章：接口测试用例编写---用例编写进阶



RequestMethod	ParamsType	//bf_account	//user_id	//real_name	//id_card	Template	/sign	/requestParams	ResponseContent
POST	XML	Phone	UserID	UserName	IDCard	Up	AES+MD5+	AES+	<?xml
POST	XML	Phone	UserID	UserName	IDCard	Up	AES+MD5+	AES+	

列名：**//bf_account**：将下层模板中指定的值替换成：Phone 关键字随机生成的手机号码。

列名：**//user_id**：将下层模板中指定的值替换成：UserID 关键字随机生成的 user_id。

列名：**//real_name**：将下层模板中指定的值替换成：UserName 关键字随机生成的用户姓名。

列名：**//id_card**：将下层模板中指定的值替换成：IDCard 关键字随机生成的身份证号码。

以上为随机生成对应的字段快速执行测试，但是有些测试场景并不适合，例如，单一注册，此方式即可满足，但是再加上绑卡操作。那么就需要获取以上的字段，例如：**user_id**、**bf_account**，但上面是随机生成的字段，在未生成前，程序都不知道是什么，怎么办呢？

看下面，参数填充的进阶写法：↓



Step	Refer target	Describe	Path	RequestMethod	ParamsType	Params	Template	ResponseContent
1	apiBindBankCard-001	apiRegister-001	/api-bind-bank-card	POST	XML	\$(Down.//bf_account)	BankCard	AES+MD5+

列名：**Refer target**：指定参数获取的目标，填写需要获取接口的唯一值：Step

列名：**//user_id**：此处填写\${Down.//user_id}

列名：**//real_name**：此处填写\${Down.//user_id}

列名：**//id_card**：将下层模板中指定的值替换成：BankCard 关键字随机生成的身份证号码。

☆☆☆☆☆进阶的写法要点：

0.在 **Refer target** 指定上一接口步骤。

获取值的方式为：

1.使用 **\${}** 将参数包起来。

- 2.内部的值可以获取指定模板的值：Up、Down、Response
- 3.使用.将分隔模板和参数路径
- 4.指定对应值的路径：//user_id
- 5.可以只写路径,不写模板，但不建议这样做。

下面可以看下进阶用例的执行日志：

```
08-31 17:27:17 [3437] [com.Api.Utils.ExcelUtils] [151] - 开始执行测试用例
08-31 17:27:19 [5505] [com.Api.Utils.ExcelUtils] [295] - 执行步骤：apiRegister-001
08-31 17:27:19 [5505] [com.Api.Utils.ExcelUtils] [296] - 执行步骤的描述：(通过测试)注册账户
08-31 17:27:19 [5515] [com.Api.Utils.ExcelUtils] [189] - 替换参数：kay=//bf_account; value=18787773164
08-31 17:27:19 [5522] [com.Api.Utils.ExcelUtils] [189] - 替换参数：kay=//user_id; value=6637640
08-31 17:27:19 [5523] [com.Api.Utils.ExcelUtils] [189] - 替换参数：kay=//real_name; value=师伯融
08-31 17:27:19 [5528] [com.Api.Utils.ExcelUtils] [189] - 替换参数：kay=//id_card; value=620102196009118867
08-31 17:27:19 [5955] [com.Api.Utils.ExcelUtils] [189] - 替换参数：kay=//sign; value=5eb03246324f3211e5e170fd0a3b2
08-31 17:27:19 [5957] [com.Api.Utils.ExcelUtils] [189] - 替换参数：kay=//requestParams; value=412284fac44951266706558412a5511c689c5538754005b43c3dab06f05bb0cf7c
08-31 17:27:19 [5957] [com.Api.Utils.ExcelUtils] [227] - 输出请求参数：sign=5eb03246324f3211e5e170fd0a3b2&requestParams=412284fac44951266706558412a5511c689c
08-31 17:27:19 [5959] [com.Api.Utils.ExcelUtils] [228] - 请求地址：http://api-custody.baofoo.com/fund-custody/custody/apiRegister.do
08-31 17:27:20 [7103] [com.baofoo.util.HttpRequestUtil] [150] - 状态码：200
08-31 17:27:20 [7104] [com.Api.Utils.ExcelUtils] [237] - 输出响应参数：<?xml version="1.0" encoding="UTF-8" ?><rs><code>CSD000</code><msg>处理成功</msg><sign>5
08-31 17:27:20 [7107] [com.Api.Utils.ExcelUtils] [461] - 获取实际值：CSD000
08-31 17:27:20 [7107] [com.Api.Utils.ExcelUtils] [462] - 获取预期值：CSD000
08-31 17:27:20 [7107] [com.Api.Utils.ExcelUtils] [201] - 获取执行结果：TRUE,
08-31 17:27:20 [7107] [com.Api.Utils.ExcelUtils] [482] - 匹配参数，加密参数，动态匹配：MD5+Array
08-31 17:27:20 [7109] [com.Api.Config.ResponseMethod] [93] - 获取实际值：e89e3be8e509b8a220c1f3ca3490b089
08-31 17:27:20 [7109] [com.Api.Config.ResponseMethod] [94] - 获取预期值：e89e3be8e509b8a220c1f3ca3490b089
08-31 17:27:20 [7109] [com.Api.Utils.ExcelUtils] [201] - 获取执行结果：TRUE, TRUE,
08-31 17:27:20 [7110] [com.Api.Utils.ExcelUtils] [461] - 获取实际值：处理成功
08-31 17:27:20 [7111] [com.Api.Utils.ExcelUtils] [462] - 获取预期值：处理成功
08-31 17:27:20 [7111] [com.Api.Utils.ExcelUtils] [201] - 获取执行结果：TRUE, TRUE, TRUE,
08-31 17:27:20 [7111] [com.Api.Utils.ExcelUtils] [482] - 匹配参数，加密参数，动态匹配：/Contains
08-31 17:27:20 [7112] [com.Api.Utils.ExcelUtils] [201] - 获取执行结果：TRUE, TRUE, TRUE, TRUE,
08-31 17:27:20 [7112] [com.Api.Utils.ExcelUtils] [314] - 执行步骤：apiRegister-001--->执行结果值：TRUE, TRUE, TRUE, TRUE
08-31 17:27:20 [7112] [com.Api.Utils.ExcelUtils] [315] - 执行步骤：apiRegister-001--->预期结果值：TRUE, TRUE, TRUE, TRUE
08-31 17:27:20 [7112] [com.Api.Utils.ExcelUtils] [320] - 最终执行结果：通过
08-31 17:27:27 [14203] [com.Api.Utils.ExcelUtils] [151] - 开始执行测试用例
08-31 17:27:31 [17448] [com.Api.Utils.ExcelUtils] [295] - 执行步骤：apiBindBankCard-001
08-31 17:27:31 [17448] [com.Api.Utils.ExcelUtils] [296] - 执行步骤的描述：(通过)绑定银行卡
08-31 17:27:31 [17448] [com.Api.notes.FindAndSaveApiValue] [47] - 获取上一接口执行结果：==>【Pass】
08-31 17:27:31 [17449] [com.Api.notes.FindAndSaveApiValue] [179] - 获取上一接口参数信息：<?xml version="1.0" encoding="UTF-8"?>
<custody_req><bf_account>18787773164</bf_account><user_id>6637640</user_id><real_name>师伯融</real_name><id_card>620102196009118867</id_card><account_type>1</acc
08-31 17:27:31 [17451] [com.Api.Utils.ExcelUtils] [189] - 替换参数：kay=//mobile; value=18787773164
08-31 17:27:31 [17452] [com.Api.notes.FindAndSaveApiValue] [47] - 获取上一接口执行结果：==>【Pass】
08-31 17:27:31 [17452] [com.Api.notes.FindAndSaveApiValue] [179] - 获取上一接口参数信息：<?xml version="1.0" encoding="UTF-8"?>
<custody_req><bf_account>18787773164</bf_account><user_id>6637640</user_id><real_name>师伯融</real_name><id_card>620102196009118867</id_card><account_type>1</acc
08-31 17:27:31 [17454] [com.Api.Utils.ExcelUtils] [189] - 替换参数：kay=//user_id; value=6637640
08-31 17:27:31 [17458] [com.Api.Utils.ExcelUtils] [189] - 替换参数：kay=//bank_no; value=5264108397821514
08-31 17:27:31 [17461] [com.Api.Utils.ExcelUtils] [189] - 替换参数：kay=//sign; value=b6e929c368f9ee38a27eec32a7fb2088
08-31 17:27:31 [17462] [com.Api.Utils.ExcelUtils] [189] - 替换参数：kay=//requestParams; value=412284fac44951266706558412a5511c689c5538754005b43c3dab06f05bb0cf7c
08-31 17:27:31 [17462] [com.Api.Utils.ExcelUtils] [227] - 输出请求参数：sign=b6e929c368f9ee38a27eec32a7fb2088&requestParams=412284fac44951266706558412a5511c689c
08-31 17:27:31 [17462] [com.Api.Utils.ExcelUtils] [228] - 请求地址：http://api-custody.baofoo.com/fund-custody/custody/apiBindBankCard.do
08-31 17:27:31 [18010] [com.baofoo.util.HttpRequestUtil] [150] - 状态码：200
08-31 17:27:31 [18010] [com.Api.Utils.ExcelUtils] [237] - 输出响应参数：<?xml version="1.0" encoding="UTF-8" ?><rs><code>CSD000</code><msg>处理成功</msg><card_
08-31 17:27:31 [18011] [com.Api.Utils.ExcelUtils] [461] - 获取实际值：CSD000
08-31 17:27:31 [18012] [com.Api.Utils.ExcelUtils] [462] - 获取预期值：CSD000
08-31 17:27:31 [18012] [com.Api.Utils.ExcelUtils] [201] - 获取执行结果：TRUE,
08-31 17:27:31 [18012] [com.Api.Utils.ExcelUtils] [482] - 匹配参数，加密参数，动态匹配：MD5+Appoint
08-31 17:27:31 [18012] [com.Api.Config.ResponseMethod] [111] - 获取实际值：e89e3be8e509b8a220c1f3ca3490b089
08-31 17:27:31 [18012] [com.Api.Config.ResponseMethod] [112] - 获取预期值：e89e3be8e509b8a220c1f3ca3490b089
08-31 17:27:31 [18013] [com.Api.Utils.ExcelUtils] [201] - 获取执行结果：TRUE, TRUE,
08-31 17:27:31 [18013] [com.Api.Utils.ExcelUtils] [461] - 获取实际值：处理成功
08-31 17:27:31 [18013] [com.Api.Utils.ExcelUtils] [462] - 获取预期值：处理成功
08-31 17:27:31 [18013] [com.Api.Utils.ExcelUtils] [201] - 获取执行结果：TRUE, TRUE, TRUE,
08-31 17:27:31 [18013] [com.Api.Utils.ExcelUtils] [482] - 匹配参数，加密参数，动态匹配：/Contains
08-31 17:27:31 [18013] [com.Api.Utils.ExcelUtils] [201] - 获取执行结果：TRUE, TRUE, TRUE, TRUE,
08-31 17:27:31 [18014] [com.Api.Utils.ExcelUtils] [314] - 执行步骤：apiBindBankCard-001--->执行结果值：TRUE, TRUE, TRUE, TRUE
08-31 17:27:31 [18014] [com.Api.Utils.ExcelUtils] [315] - 执行步骤：apiBindBankCard-001--->预期结果值：TRUE, TRUE, TRUE, TRUE
08-31 17:27:31 [18014] [com.Api.Utils.ExcelUtils] [320] - 最终执行结果：通过
08-31 17:27:31 [18168] [com.Api.Email.SendEmail] [57] - Report Link：暂时还没有web端
08-31 17:27:32 [18681] [com.Api.Email.SendEmail] [260] - mail send success!
```