**测试用例可能存在问题，比如参数值多了空格或者参数名多了空格，使用时请注意**

**对于红色字体请仔细阅读**

类型名称全部去掉的优化版本

一定要运行mustRun.sql同步数据库 其他的运行过就不要运行了

功能点：

1. 用户登录需要在数据库中添加记录

功能：用户登录

请求方式：POST

请求路径：http://localhost:9090/user/login

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| userName | String | 是 | 用户名 |
| wxId | String | 是 | 微信ID |

返回值：{

    "uid": 3,

    "userName": "小王",

    "wxId": "3322",

    "registerDate": "2023-03-08T07:19:04.000+00:00"

}

类型：实体类

说明：用户一定可以登录成功，数据库中实事先没有记录就添加一条，存在就成功登录。需要把用户的登录状态存在session中

2.在”记一笔”支出页面中可以根据账本的类型渲染出对应的款项类型

账本类型名称用11号接口查

请求方式：GET

请求路径：http://localhost:9090/book/out

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |

返回值：{

    "BO2": "购物",

    "BO1": "餐饮",

    "BO4": "交通",

    "BO3": "日用",

    "BO6": "水果",

    "BO5": "蔬菜",

    "BO8": "运动",

    "BO7": "零食",

    "BO9": "娱乐",

    "BO13": "家庭",

    "BO12": "快递",

    "BO11": "服饰",

    "CO1a0bebddc08111ed8be0002b67dd5ef6": "你好",

    "BO10": "通讯",

    "CO8711ad93bfc811ed8be0002b67dd5ef6": "测试"

}

类型：HashMap<String,String>

说明：通过uid找到拥有得账本，再通过用户选择的账本名称从账本表中查询到账本类型id，再在账本类型表中利用账本类型id查找到拥有的id款项，通过对字符串的分解从款项类型表，在用户自定义款项类型表通过uid的账本类型查找对应的自定义款项添加进去。

注意：同一个用户的一个账簿对应的bookkeeping\_type表中的类型名称字段要互不相同，不同用户间可以存在相同，所以在用户自定义添加一个账本模板的时候需要检查该账本所关联的所有模板中是否重名。

**测试用例：**

http://localhost:9090/book/out?uid=1&bookKeepingName=我的账簿1

3.在”记一笔”收入页面中可以根据账本的类型渲染出对应的款项类型

账本类型名称用11号接口查

请求方式：GET

请求路径：http://localhost:9090/book/in

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |

返回值：{

    "BI19": "礼金",

    "BI18": "红包",

    "BI17": "工资",

    "CI1": "遗产"

}

类型：HashMap<String,String>

说明：通过uid找到拥有得账本，再通过用户选择的账本名称从账本表中查询到账本类型id，再在账本类型表中利用账本类型id查找到拥有的id款项，通过对字符串的分解从款项类型表中找到相应的类型，在用户自定义款项类型表通过uid的账本类型查找对应的自定义款项添加进去。

**测试用例：**

http://localhost:9090/book/in?uid=1&bookKeepingName=我的账簿1

4. 在”记一笔”支出的设置页面对选择的账本修改(可以增加或删除)款项类型，

**对于基本款项可以做到增加和删除、但对于自定义款项只能做的增加，删除功能见22号接口。原因是对于基本款项是对该字段进行一个覆盖操作，自定义款项是插入操作，所以删除就只能单拿出来**

请求方式：POST

请求路径：http://localhost:9090/book/setting\_add\_out

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |
| json | JSON | 是 | 款项类型名称的json，不能为空 |

**Json要从body里传过来 数据类型为json**

Json参数示例

{

    "json":["BO8","BO9","BO10","BO11","BO12","BO13","BO14","BO15","BO16","CO旅游","CO游戏"]

}

注意：json表示更改后还有的那些项

返回值：success 类型：String

说明：页面渲染出来的默认可添加项是basic\_funds表中存在的项，从前端传入款项类型的名称和账本类型和用户uid后，把款项类型的名称和uid添加到customed\_funds表中，如果是用户自定义的也添加到customed\_funds表中。

**测试用例：**

http://localhost:9090/book/setting\_add\_out?uid=2&bookKeepingName=我的账簿1

{

    "json":["BO8","BO9","BO10","BO11","BO12","BO13","BO14","BO15","BO16","CO旅游","CO游戏"]

}

5. 在”记一笔”收入的设置页面对选择的账本修改(可以增加或删除)款项类型

**对于基本款项可以做到增加和删除、但对于自定义款项只能做的增加，删除功能见22号接口。原因是对于基本款项是对该字段进行一个覆盖操作，自定义款项是插入操作，所以删除就只能单拿出来**

请求方式：POST

请求路径：http://localhost:9090/book/setting\_add\_in

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |
| json | JSON | 是 | 款项类型名称的json，不能为空 |

**Json要从body里传过来**

Json 参数示例

{

    "json":["BI25","BI26","BI27","CI基金","CI股票"]

}

注意：json表示更改后还有的那些项

返回值：success 类型：String

说明：页面渲染出来的默认可添加项是basic\_funds表中存在的项，从前端传入款项类型的名称和账本类型和用户uid后，把款项类型的名称和uid添加到customed\_funds表中，如果是用户自定义的也添加到customed\_funds表中。

**测试用例：**

http://localhost:9090/book/setting\_add\_in?uid=2&bookKeepingName=我的账簿1

{

    "json":["BI25","BI26","BI27","CI基金","CI股票"]

}

6.在“记一笔“支出页面实现记账功能

请求方式：POST

请求路径：http://localhost:9090/book/out

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |
| fundId | String | 否 | 基本款项类型ID |
| customedFundId | String | 否 | 自定义款项类型ID |
| accountId | int | 是 | 账户id |
| amount | String | 是 | 金额大小 |
| comment | String | 否 | 备注 |
| enclosure | String | 否 | 附件 |
| time | String | 是 | 时间 格式  yyyy-MM-dd HH:mm:ss |

返回值：success 类型：String

说明：用户填好金额后选择对应的款项类型，在支出表中添加一条记录

**测试用例**：

http://localhost:9090/book/out?uid=1&bookKeepingName=我的账簿1&accountId=1&amount=400&time=2023-1-9 13:25:30&fundId=BO4&customedFundId&comment=无&enclosure=无

7.在“记一笔“收入页面实现记账功能

请求方式：POST

请求路径：http://localhost:9090/book/in

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |
| fundId | String | 否 | 基本款项类型ID |
| customedFundId | String | 否 | 自定义款项类型ID |
| accountId | int | 是 | 账户id |
| amount | String | 是 | 金额大小 |
| comment | String | 否 | 备注 |
| enclosure | String | 否 | 附件 |
| time | String | 是 | 时间 格式  yyyy-MM-dd HH:mm:ss |

返回值：success 类型：String

说明：用户填好金额后选择对应的款项类型，在收入表中添加一条记录

**测试用例：**

http://localhost:9090/book/in?uid=1&bookKeepingName=我的账簿1&accountId=1&amount=400&time=2023-1-9 13:25:30&fundId=BI17&customedFundId&comment=无&enclosure=无

8. 获取基本款项表的所有内容

请求方式：GET

请求路径：http://localhost:9090/book/basic

参数：无

返回值

[

    {

            "fund\_id": "BI17",

            "fund\_name": "工资",

            "icon": "icon-gongzi"

        },

        {

            "fund\_id": "BI18",

            "fund\_name": "红包",

            "icon": "icon-hongbao"

        },

]

作用：用于4号5号接口的json生成

**测试用例：**

http://localhost:9090/book/basic

9.新建一个账本

请求方式：POST

请求路径：http://localhost:9090/book/new

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |
| bookKeepingCover | String | 是 | 封面 |
| extraMember1 | Integer | 否 | 额外用户1 |
| extraMember2 | Integer | 否 | 额外用户2 |
| template | String | 是 | 模板对应的基本款项编号序列字符串 |
| bookKeepingTypeName | String | 是 | 账本类型名称 |

**template 输入示例“****BO3-BO4-BO5-BO6-BO8-BI17-BI23-BI26-BI27”**

**注意要用 – 连接**

返回值：“success” 类型：String

说明：在bookkeeping表中添加一条记录，在bookkeeping\_type表中添加一条记录，创建的时候可以在前端提供模板，然后传到后端，后面用户可以通过自定义的方式修改

**测试用例**： 9号接口 用的时候改参数，注意同一个用户的所有账簿名不同

http://localhost:9090/book/new?uid=2&bookKeepingName=我的账簿2&bookKeepingCover=大海&extraMember1=2&extraMember2=3&template=BO3-BO4-BO5-BO6-BO8-BI17-BI23-BI26-BI27&bookKeepingTypeName=发财

10.修改账本

请求方式：PUT

请求路径：http://localhost:9090/book/change

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |
| bookKeepingNameNew | String | 是 | 账本更新后的名字，如果不修改名字就让它和原名称一致 |
| bookKeepingCover | String | 是 | 封面 |
| extra\_member1 | Integer | 否 | 额外用户1 |
| extra\_member2 | Integer | 否 | 额外用户2 |

返回值：“success” 类型：String

说明：修改账本的一些属性

前端要做账本名称的控制，必须要保证用户更改后的账本名称和该用户的其它账本名称不同

如果发生重名后端会报异常

**测试用例：**

http://localhost:9090/book/change?uid=2&bookKeepingName=我的账簿2&bookKeepingCover=天空&extraMember1=&extraMember2=2&bookKeepingNameNew=我的账簿3

11.获取账本对应的模板名称

请求方式：Get

请求路径：http://localhost:9090/book/getTypeNames

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |

返回值： 类型：List<String>

[

    "日常开销"

]

**测试用例：**

http://localhost:9090/book/getTypeNames?uid=1&bookKeepingName=我的账簿1

12. 查询对应账本的支出明细

请求方式：Get

请求路径：http://localhost:9090/book/getPayment

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |

返回结果 类型List<JSONObject>

{

    "code": "200",

    "message": "成功",

    "result": [

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2002-11-14T01:37:23.000+00:00",

            "款项名称": "餐饮",

            "备注": "无",

            "金额": "-100"

        },

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2023-03-18T04:27:20.000+00:00",

            "款项名称": "购物",

            "备注": "无",

            "金额": "-200"

        },

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2002-11-14T01:37:23.000+00:00",

            "款项名称": "蔬菜",

            "备注": "无",

            "金额": "-100"

        },

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2023-04-03T12:24:04.000+00:00",

            "款项名称": "蔬菜",

            "备注": "无",

            "金额": "-500"

        },

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2023-04-06T12:24:51.000+00:00",

            "款项名称": "日用",

            "备注": "无",

            "金额": "-300"

        },

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2023-04-08T12:28:05.000+00:00",

            "款项名称": "餐饮",

            "备注": "无",

            "金额": "-100"

        },

        {

            "款项类型": "自定义",

            "账户名称": "储蓄卡",

            "时间": "2023-05-10T12:28:24.000+00:00",

            "款项名称": "你好",

            "备注": "无",

            "金额": "-300"

        },

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2023-04-03T12:33:39.000+00:00",

            "款项名称": "蔬菜",

            "备注": "无",

            "金额": "-100"

        },

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2023-01-09T05:25:30.000+00:00",

            "款项名称": "交通",

            "备注": "无",

            "金额": "-400"

        }

    ]

}

**测试用例：**

http://localhost:9090/book/getPayment?uid=1&bookKeepingName=我的账簿1

13. 查询对应账本的收入明细

请求方式：Get

请求路径：http://localhost:9090/book/getIncome

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |

返回结果 类型List<JSONObject>

{

    "code": "200",

    "message": "成功",

    "result": [

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2023-04-03T01:37:23.000+00:00",

            "款项名称": "工资",

            "备注": "无",

            "金额": "100"

        },

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2023-04-04T01:37:23.000+00:00",

            "款项名称": "红包",

            "备注": "无",

            "金额": "100"

        },

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2023-04-05T02:50:08.000+00:00",

            "款项名称": "礼金",

            "备注": "无",

            "金额": "160"

        },

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2023-04-05T04:50:45.000+00:00",

            "款项名称": "礼金",

            "备注": "无",

            "金额": "130"

        },

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2023-04-10T06:57:37.000+00:00",

            "款项名称": "礼金",

            "备注": "无",

            "金额": "600"

        },

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2023-05-01T07:07:06.000+00:00",

            "款项名称": "礼金",

            "备注": "无",

            "金额": "300"

        },

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2022-03-17T07:20:49.000+00:00",

            "款项名称": "红包",

            "备注": "无",

            "金额": "300"

        },

        {

            "款项类型": "自定义",

            "账户名称": "储蓄卡",

            "时间": "2023-04-04T11:18:10.000+00:00",

            "款项名称": "遗产",

            "备注": "无",

            "金额": "200"

        },

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2023-01-09T05:25:30.000+00:00",

            "款项名称": "工资",

            "备注": "无",

            "金额": "400"

        },

        {

            "款项类型": "基本",

            "账户名称": "储蓄卡",

            "时间": "2023-01-20T05:25:30.000+00:00",

            "款项名称": "礼金",

            "备注": "无",

            "金额": "200"

        }

    ]

}

**测试用例**：

http://localhost:9090/book/getIncome?uid=1&bookKeepingName=我的账簿1

对于账本和账本模板的一些说明，由于初期理解的错误以为一个账本可以有多个模板，所以把账本和模板之间的关系变成了多对多，但这对使用上来说并没有问题。用户在新建账本的时候会有两个操作一个是在账本表中插入一条记录，一个是在模板表中插入一条记录，对于的名字字段是bookKeepingName和bookKeepingTypeName，可以让选择让两个名字都相同也可以就是说账本是一个名字，模板是一个名字，如果选择让两个名字相同，就要保证同一个用户的所有账本的名称都不同。前端可以提供一些模板供用户快速新建账本。新建账本用到的接口有9号和11号接口

14. 获取用户拥有的所有账本

请求方式：Get

请求路径：http://localhost:9090/book/getBookNames

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |

返回值： 类型：List<String>

{

    "code": "200",

    "message": "成功",

    "result": [

        "我的账簿1",

        "我的账簿2",

        "我的账簿3",

        "我的账簿4"

    ]

}

**测试用例：**

http://localhost:9090/book/getBookNames?uid=1

15. 统计-收入页面显示

请求方式：Get

请求路径：http://localhost:9090/book/getPeriodIncome

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |
| nowTime | String | 不一定 | 用于查询一周账单,提供查询时的时间,在按月/年查询时可为空 |
| startTime | String | 不一定 | 用于查询一个月/年的账单,提供该月/年开始时间,在按周查询时可为空 |
| endTime | String | 不一定 | 用于查询一个月/年的账单,提供该月/年结束时间,在按周查询时可为空 |
| getWhat | String | 是 | 用于区别你查询的是周/月/年 提供的输入为{week,month,year} |

返回值类型 HashMap<String,HashMap<String, Integer>>

**测试用例：**

测试统计周

http://localhost:9090/book/getPeriodIncome?uid=1&bookKeepingName=我的账簿1&nowTime=2023-04-03 03:26:54&startTime&endTime&getWhat=week

测试统计月

http://localhost:9090/book/getPeriodIncome?uid=1&bookKeepingName=我的账簿1&nowTime=&startTime=2023-04-01 03:26:54&endTime=2023-04-30 03:26:54&getWhat=month

测试统计年

http://localhost:9090/book/getPeriodIncome?uid=1&bookKeepingName=我的账簿1&nowTime=&startTime=2023-01-01 03:26:54&endTime=2023-12-31 03:26:54&getWhat=year

16.统计-支出页面显示

请求方式：Get

请求路径：http://localhost:9090/book/getPeriodPayment

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |
| nowTime | String | 不一定 | 用于查询一周账单,提供查询时的时间,在按月/年查询时可为空 |
| startTime | String | 不一定 | 用于查询一个月/年的账单,提供该月/年开始时间,在按周查询时可为空 |
| endTime | String | 不一定 | 用于查询一个月/年的账单,提供该月/年结束时间,在按周查询时可为空 |
| getWhat | String | 是 | 用于区别你查询的是周/月/年 提供的输入为{week,month,year} |

返回值类型 HashMap<String,HashMap<String, Integer>>

测试用例：

测试统计周

http://localhost:9090/book/getPeriodPayment?uid=1&bookKeepingName=我的账簿1&nowTime=2023-04-03 03:26:54&startTime&endTime&getWhat=week

测试统计月

http://localhost:9090/book/getPeriodPayment?uid=1&bookKeepingName=我的账簿1&nowTime=&startTime=2023-04-01 03:26:54&endTime=2023-04-30 03:26:54&getWhat=month

测试统计年

http://localhost:9090/book/getPeriodPayment?uid=1&bookKeepingName=我的账簿1&bookKeepingTypeName=日常开销&nowTime=&startTime=2023-01-01 03:26:54&endTime=2023-12-31 03:26:54&getWhat=year

17.查询指定账本的月度预算

请求方式：Get

请求路径：http://localhost:9090/book/getBudget

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |
| month | String | 是 | 表示几月份 |

**month的输入应为jan/feb/mar/apr/may/jun/jul/aug/sept/oct/nov/dec其中之一**

返回值：

{

    "code": "200",

    "message": "成功",

    "result": {

        "该月预算为": "1000"

    }

}

**测试用例**：

http://localhost:9090/book/getBudget?uid=1&bookKeepingName=我的账簿1&month=feb

18.修改预算

请求方式：PUT

请求路径：http://localhost:9090/book/changeBudget

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |
| budget | String | 是 | 预算 |
| month | String | 是 | 表示几月份 |

**month的输入应为jan/feb/mar/apr/may/jun/jul/aug/sept/oct/nov/dec其中之一**

返回值：success

**测试用例**

http://localhost:9090/book/changeBudget?uid=1&bookKeepingName=我的账簿1&month=jan&budget=3500

19 统计对应账簿某月的总支出

使用16号接口的统计年支出，里面有每个月的总支出，取想要的那个月份就行

20.删除账本

请求方法：DELETE

请求路径：http://localhost:9090/book/deleteBookkeeping

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| Uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |

**记得改参数**

**测试用例：**

http://localhost:9090/book/deleteBookkeeping?uid=2&bookKeepingName=我的账簿3

21.根据几个大类查询，支出收入记录 饮食/娱乐/交通/生活/工资/理财/人情/其它

请求方式：GET

请求路径：http://localhost:9090/book/getPartRecord

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| uid | Int | 是 | 用户id |
| bookKeepingName | String | 是 | 账本名称 |
| classification | String | 是 | 表示哪一类 |

**Classification 的可选值有 {饮食/娱乐/交通/生活/工资/理财/人情/其它}**

饮食对应：餐饮、蔬菜、水果、零食

娱乐对应：通讯、社交、旅行、购物、娱乐

交通对应：快递、交通

生活对应：服饰、家庭、住房、日用、运动

工资对应：工资、年终奖、收款

理财对应：分红、理财

人情对应: 借入、礼金、红包

其它对应：用户的自定义款项、生活费、其它、租金

返回值

{

    "code": "200",

    "message": "成功",

    "result": [

        {

            "款项类型": "自定义",

            "账户名称": "储蓄卡",

            "时间": "2023-05-10T12:28:24.000+00:00",

            "款项名称": "你好",

            "备注": "无",

            "金额": "-300"

        },

        {

            "款项类型": "自定义",

            "账户名称": "储蓄卡",

            "时间": "2023-04-04T11:18:10.000+00:00",

            "款项名称": "遗产",

            "备注": "无",

            "金额": "200"

        }

    ]

}

**测试用例：**

使用时可以改改参数，查不到就返回空

http://localhost:9090/book/getPartRecord?uid=1&bookKeepingName=我的账簿1&classification=其它

22.删除自定义款项 配合4号5号接口使用，弥补不能删除自定义开款项的缺陷

请求方式：DELETE

请求路径：http://localhost:9090/book/deleteCustomFund

|  |  |  |  |
| --- | --- | --- | --- |
| 参数 | 类型 | 是否必须 | 说明 |
| customFundId | String | 是 | 自定义款项ID |

**测试用例：**

http://localhost:9090/book/deleteCustomFund?customFundId=COb2059833d41e11ed8ca2002b67dd5ef6