文件编号：UTRY-CMMI-RD-T-06

**Swagger接口定义规范说明书**

版本：V1.0

发布时间：2022-11-08

**浙江远传信息技术股份有限公司**

**变更记录**

\*A - 增加 M - 修订 D - 删除

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 变更  版本号 | 日期 | 变更类型 | 修改人 | 变更摘要 | 备注 |
| V1.0 | 2022-11-08 | A-新增 | 时李樊 | 新增Swagger接口定义规范说明书 |  |
| V1.1 | 2022-11-09 | A-新增M-修订 | 时李樊 | 会议评审后：  1.修订章节2.3黄色背景  2.新增章节2.5、2.6 |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

目录

[1. 简介 4](#_Toc118979210)

[1.1 swagger的特征 4](#_Toc118979211)

[1.2 使用目的 4](#_Toc118979212)

[1.3 访问链接 5](#_Toc118979213)

[2. 后端接口的参数配置 5](#_Toc118979214)

[2.1 SwaggerTagConst 5](#_Toc118979215)

[2.2 Controller 5](#_Toc118979216)

[**2.2.1** **@Api** 5](#_Toc118979217)

[**2.2.2** **@ApiOperation** 6](#_Toc118979218)

[**2.2.3** **@PostMapping** 6](#_Toc118979219)

[2.3 DTO 7](#_Toc118979220)

[**2.3.1** **@ApiParam** 7](#_Toc118979221)

[**2.3.2** **@ApiModelProperty** 7](#_Toc118979222)

[2.4 VO 8](#_Toc118979223)

[**2.4.1** **@ApiModel** 8](#_Toc118979224)

[2.5 接口url定义规范 10](#_Toc118979225)

[2.6 返回状态码 10](#_Toc118979226)

# 简介

Swagger是一款RESTFUL接口的文档在线自动生成+功能测试功能软件。Swagger是一个规范和完整的框架,用于生成、描述、调用和可视化RESTful风格的Web服务。目标是使客户端和文件系统作为服务器以同样的速度来更新文件的方法,参数和模型紧密集成到服务器。

这个解释简单点来讲就是说，swagger是一款可以根据resutful风格生成的生成的接口开发文档，并且支持做测试的一款中间软件。

## **swagger的特征**

1. 通过代码和注释自动生成文档。在Swagger框架下，开发人员可对服务进行归类说明，对方法，模型，返回结果等进行详细说明。方便开发人员在编写代码的同时，编写文档信息。自动生成，只需很少的编辑工作，就能获得完整的REST APIs文档

2. 提供了UI界面。既展示接口信息，又提供了参数校验，测试功能

3. 形成了文档规范，支持不同的语言

4. 提供丰富的组件。

## **使用目的**

1.对于后端开发人员来说：

* 不用再手写接口文档、拼写大量的参数，避免手写错误
* 对代码侵入性低，采用全注解的方式，开发简单
* 方法参数名修改、增加、减少参数都可以直接生效，不用手动维护
* 缺点：增加了开发成本，写接口还得再写一套参数配置

2.对于前端开发来说：

* 后端只需要定义好接口，会自动生成文档，接口功能、参数一目了然
* 联调方便，如果出问题，直接测试接口，实时检查参数和返回值,就可以快速定位是前端还是后端的问题

3.对于测试：

* 对于某些没有前端界面UI的功能，可以用它来测试接口
* 操作简单，不用了解具体代码就可以操作

## 访问链接

本地：localhost:8080/#/api-doc/swagger

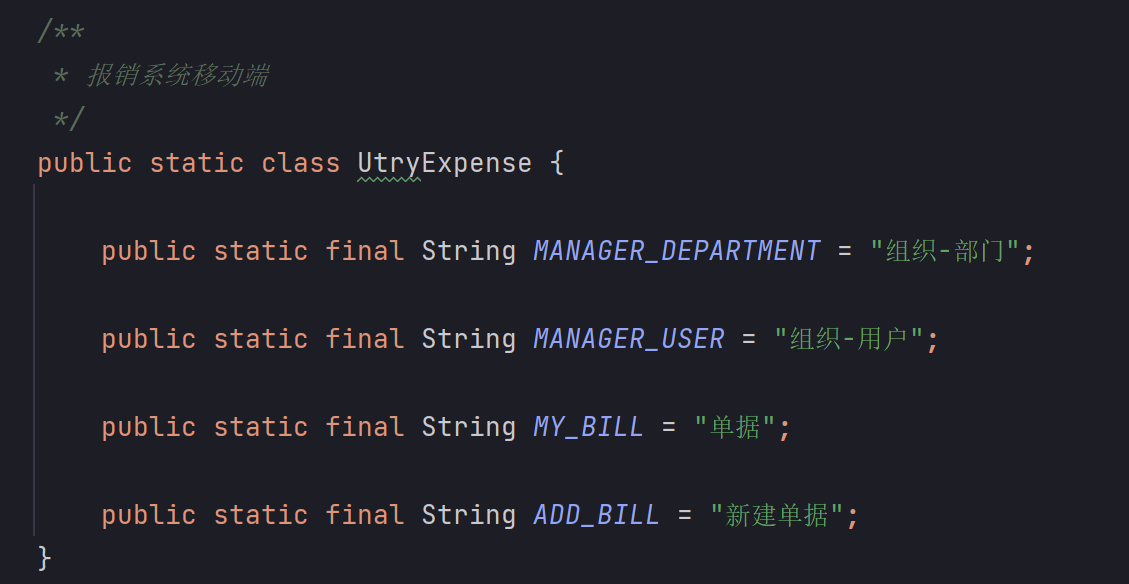
测试：10.0.2.176:8080/#/api-doc/swagger

# 后端接口的参数配置

## **SwaggerTagConst**

在SwaggerTagConst中新建一个类，以“报销系统移动端”系统为例，并在该类中声明常量参数，一个常量参数对应一个Controller.

实例代码：



在 swagger-ui.html 中显示效果：



## **Controller**

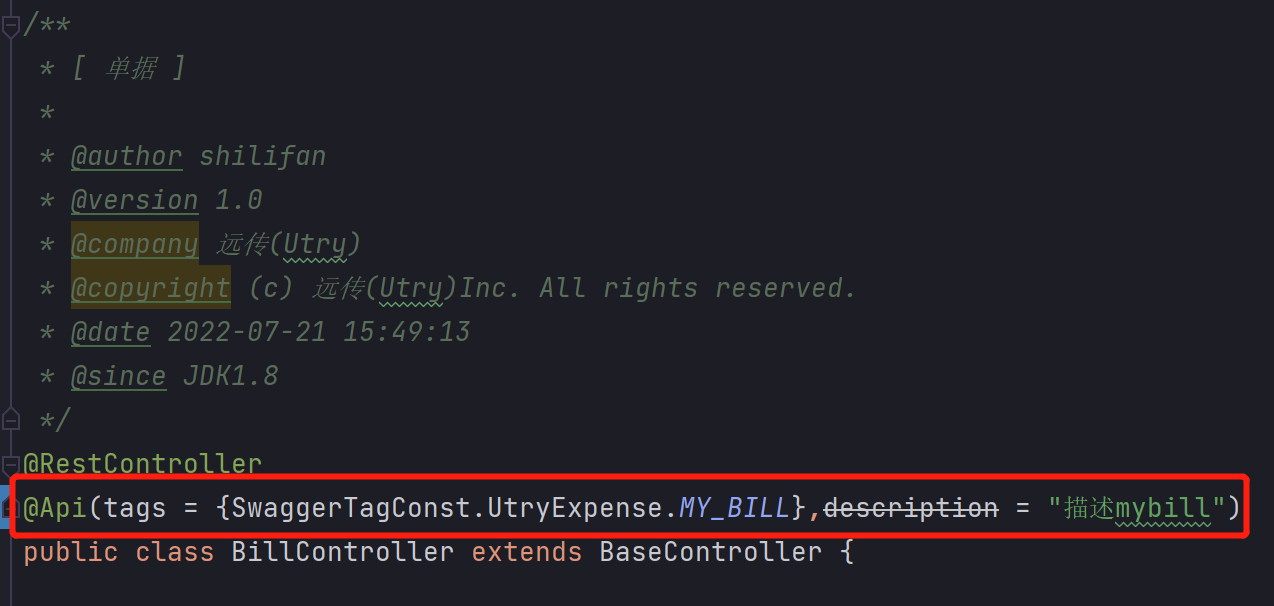
在Contriller中常用注解有以下几种：

### **@Api**

@Api 是类上注解。控制整个类生成接口信息的内容

* tags：类的名称。可以有多个值，多个值表示多个副本
* description：对该Controller的描述

实例代码：



在 swagger-ui.html 中显示效果：



### **@ApiOperation**

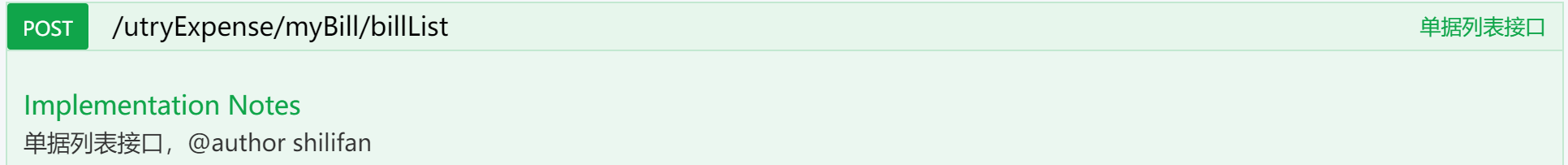
@ApiOperation 写在方法上，对方法进行总体描述

* value：接口描述
* notes：提示信息

实例代码：



在 swagger-ui.html 中显示效果：



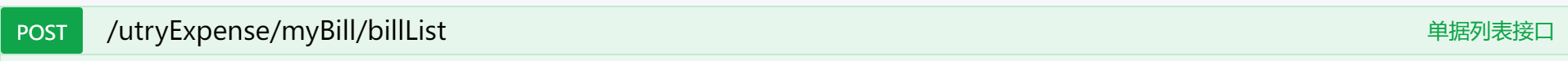
### **@PostMapping**

* value：接口路径

实例代码：



在 swagger-ui.html 中显示效果：





## **DTO**

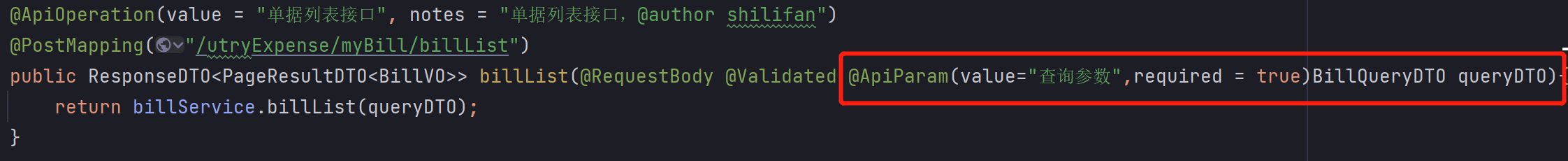
接口方法中的参数统一封装在DTO对象中（即使参数只有一个时，也封装在DTO中），在DTO中常用注解有以下几种：

### **@ApiParam**

@ApiParam作用于请求方法上，定义api参数的注解。

* value：接口描述
* required：true(必填)、false(非必填)

实例代码：



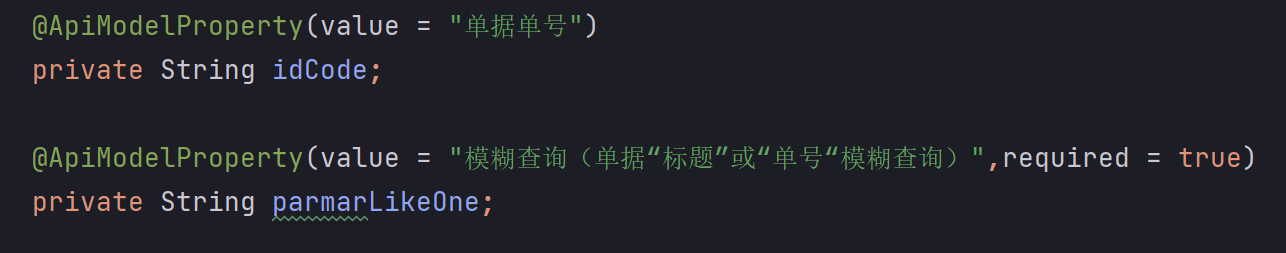
在 swagger-ui.html 中显示效果：

### **@ApiModelProperty**

@ApiModelProperty用在属性上,对属性作注解。注意：必须加上@Data注解，或者get、set方法，否则@ApiModelProperty的value不生效。

* value：接口描述
* required：true(必填)、false(非必填)

实例代码：



在 swagger-ui.html 中显示效果：

设置required=true的属性字段没有optionsl标识，反之有optionsl标识

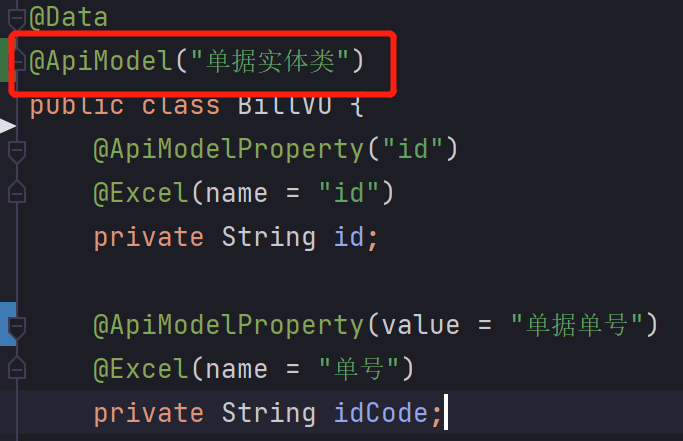


## **VO**

### **@ApiModel**

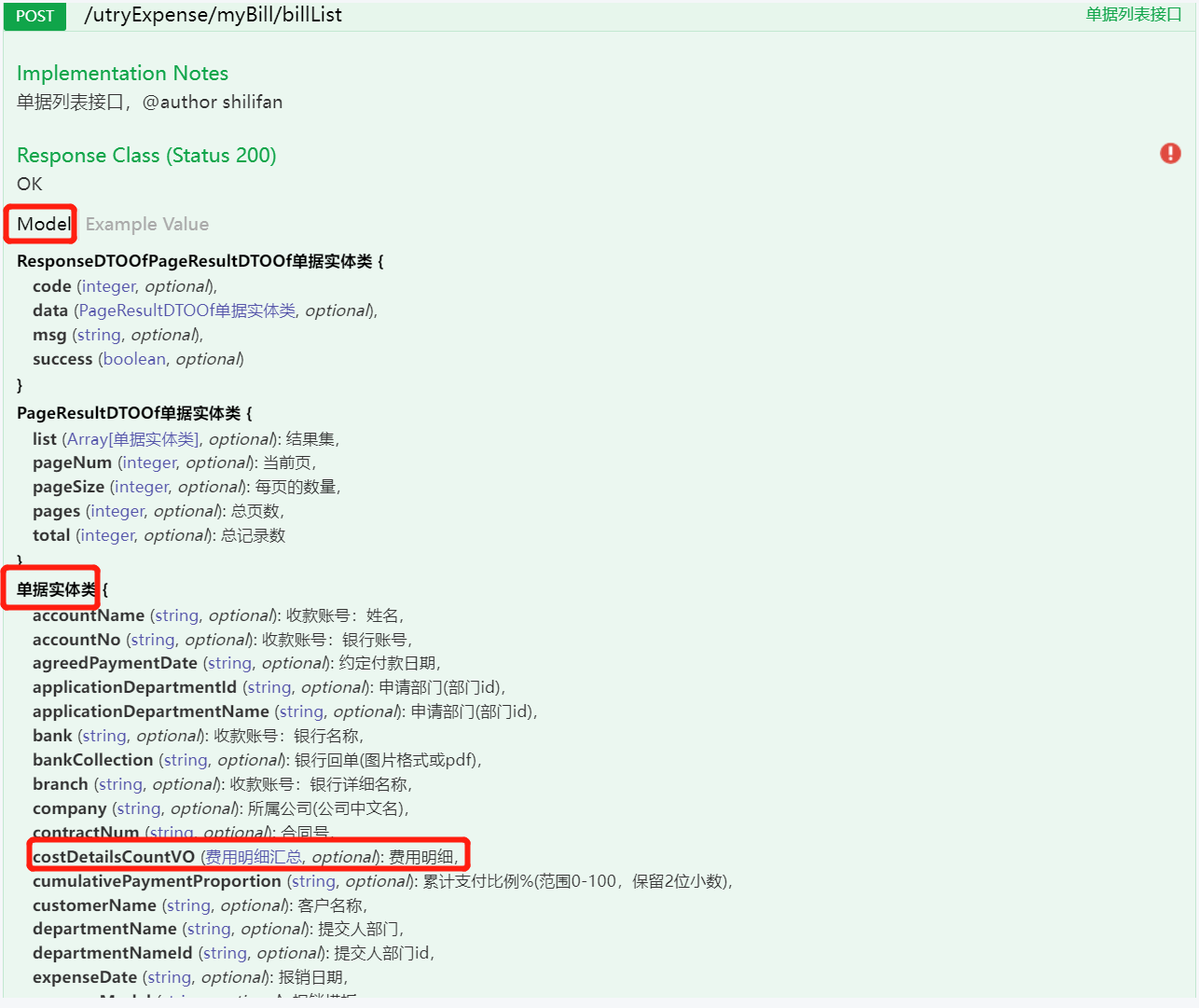
@ApiModel用于描述一个Model的信息。

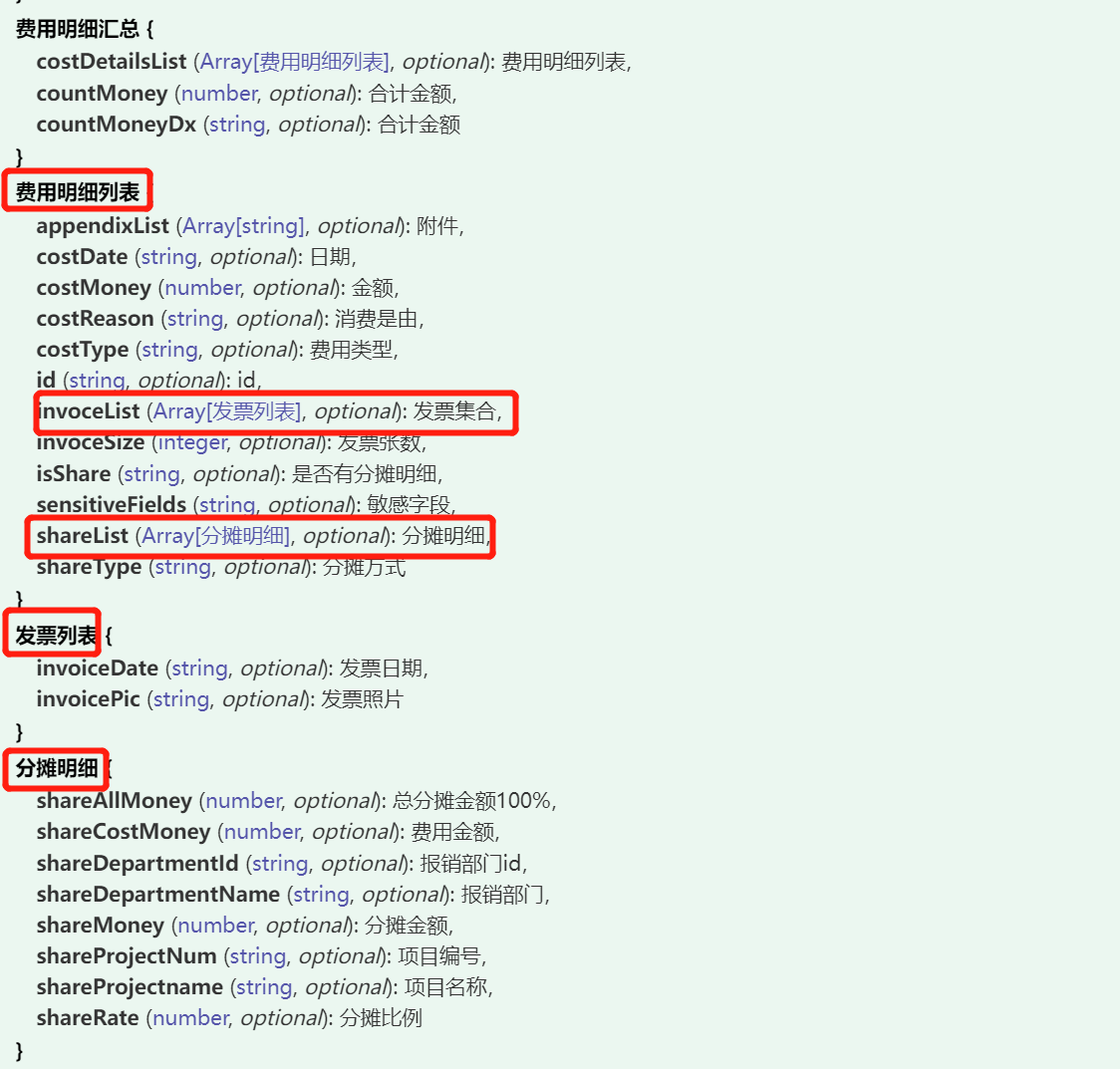
实例代码：



VO中属性值也需要加上@ApiModelProperty注解。当VO中的属性是一个对象时，也需要给该对象的属性值加上@ApiModelProperty注解。

在 swagger-ui.html 中显示效果：





## 接口url定义规范

采用小驼峰命名法

模板：/系统名称/ SwaggerTagConst中的常量名/操作动词英文/方法名

实例：

增：/utryExpense/myBill/add/bill

删：/utryExpense/myBill/delete/bill

改：/utryExpense/myBill/update/bill

查：/utryExpense/myBill/select/billList

## 返回状态码

|  |  |  |
| --- | --- | --- |
| code | msg | success |
| 1 | 操作成功! | true |
| 101 | 参数异常！ | false |
| 102 | %s参数异常！ | false |
| 111 | 系统错误 | false |
| 112 | 此功能正在开发 | false |
| 113 | 数据不存在 | false |
| 114 | 请求方式错误 | false |
| 115 | JSON格式错误 | false |
| 116 | 参数信息为空 | false |
| 117 | 请求频繁，请稍后重试 | false |