

beanpay

提供钱包帐户的加款、扣款、退款功能

提供服务包数量的增加、减少、撤销功能

- ✓ 支持钱包余额的加、扣、退
- ✓ 支持服务包数量的增、减、撤
- ✓ 独立服务运行，调用api,rpc服务集成
- ✓ 集成服务运行，使用源代码直接集成
- ✓ 高并发支持
- ✓ 基于 hydra 构建

[接口开发规范](#)

一、独立服务

独立启动 apiserver 对外提供 api , rpc 服务

1、创建数据库

使用beanpay创建数据库，支持mysql, oracle

- 编译beanpay

```
~/work/bin$ go install github.com/micro-plat/beanpay #mysql
```

或

```
~/work/bin$ go install -tags "oci" github.com/micro-plat/beanpay # oracle
```

- 创建数据表

beanpay [注册中心地址] [平台名称] 即可将 beanpay 需要的表创建到 /平台/var/db/db 配置的数据库中。

```
~/work/bin$ beanpay zk://192.168.0.109 mall #读取/mall/var/db/db 创建数据库
```

或

```
~/work/bin$ beanpay zk://192.168.0.109 mall mdb #读取 /mall/var/db/mdb 创建数据库
```

2、启动 api 服务

启动后对外提供 http , rpc 服务

- 编译 apiserver 服务

```
~/work/bin$ go install -tags "prod" github.com/micro-plat/beanpay/apiserver
```

- 安装服务, 根据安装向导设置数据库连接串

```
apiserver install -r zk://192.168.106.18 -c t
```

- 启动服务

```
apiserver start -r zk://192.168.106.18 -c t
```

3、测试服务

- 传入用户编号、名称创建钱包帐户。返回创建好的帐户信息

```
~/work/bin$ curl "http://192.168.4.121:9090/account/create?eid=86001&name=colin"
```

```
{"account_id":86000,"account_name":"colin","balance":200,"credit":0}
```

- 传入用户编号或商户编号,外部交易编号(幂等判断), 加款金额进行钱包加款。返回加款记录信息

```
~/work/bin$ curl "http://192.168.4.121:9090/account/balance/add?eid=86001&trade_no=89708"
```

```
{"account_id":"86000","amount":"200","balance":"200","change_type":"1","create_time":"2016-08-10 10:10:10"}
```

- 传入用户编号或商户编号,外部交易编号(幂等判断), 加款金额进行钱包扣款。返回扣款记录信息

```
~/work/bin$ curl "http://192.168.4.121:9090/account/balance/deduct?eid=86001&trade_no=89709"
```

```
{"account_id":"86000","amount":"-200","balance":"0","change_type":"2","create_time":"2016-08-10 10:10:10"}
```

- 传入用户编号、外部扣款交易编号(幂等判断)，退款金额进行钱包退款，退款金额不能大于扣款金额，同一笔扣款只允许一次退款操作。返回退款记录信息

```
~/work/bin$ curl "http://192.168.4.121:9090/account/balance/refund?eid=86001&trade_no=86001" -X POST -d '{"account_id":"86000","amount":"200","balance":"200","change_type":"3","create_time":"2020-01-01 00:00:00"}'
```

4、服务列表

接口开发规范

分类	服务名	类型	说明
钱包	/account/create	api、rpc	添加钱包账户
	/account/balance/query		账户信息查询
	/account/balance/add		账户余额加款
	/account/balance/deduct		账户余额扣款
	/account/balance/refund		账户余额退款
	/account/balance/query		账户余额查询
	/account/record/query		账户变动查询
服务包	/package/create	api、rpc	添加服务包
	/package/query		服务包信息查询
	/package/capacity/add		服务包数量添加
	/package/capacity/deduct		服务包数量扣除
	/package/capacity/refund		服务包数量退回
	/package/capacity/query		服务包数量查询
	/package/record/query		服务包变动查询

二、代码集成

将数据表创建到业务系统，或独立库，业务系统使用本地代码，直接进行帐户、服务包操作

1、创建数据库

使用beanpay创建数据库，支持mysql, oracle

- 编译beanpay

```
~/work/bin$ go install github.com/micro-plat/beanpay #mysql
```

或

```
~/work/bin$ go install -tags "oci" github.com/micro-plat/beanpay # oracle
```

- 创建数据表

beanpay [注册中心地址] [平台名称] 即可将 beanpay 需要的表创建到 /平台/var/db/db 配置的数据库中。

```
~/work/bin$ beanpay zk://192.168.0.109 mall #读取/mall/var/db/db 创建数据库
```

或

```
~/work/bin$ beanpay zk://192.168.0.109 mall mdb #读取 /mall/var/db/mdb 创建数据库
```

2、编码

- 创建帐户

```
account, err := beanpay.CreateAccount(ctx,
    ctx.Request.GetString("eid"),
    ctx.Request.GetString("name"))
if err != nil {
    return err
}

return account
```

- 帐户加款

```
record, err := beanpay.AddAmount(ctx,
    ctx.Request.GetString("eid"),
    ctx.Request.GetString("trade_no"),
    ctx.Request.GetInt("amount"))
if err != nil {
    return err
}
return record
```

- 帐户扣款

```
record, err := beanpay.DeductAmount(ctx,
    ctx.Request.GetString("eid"),
    ctx.Request.GetString("trade_no"),
    ctx.Request.GetInt("amount"))
if err != nil {
    return err
}
return record
```

其它操作请查看<http://github.com/micro-plat/beanpay/beanpay/service.go>