# 翌支付二期设计文档

## 企业资料修改模块

相关控制器:

1. \Home\Controller\MerchantInfoController (企业门户)
2. \Operate\Controller\MerchantInfoActionController (运营管理)

### 1.1企业门户

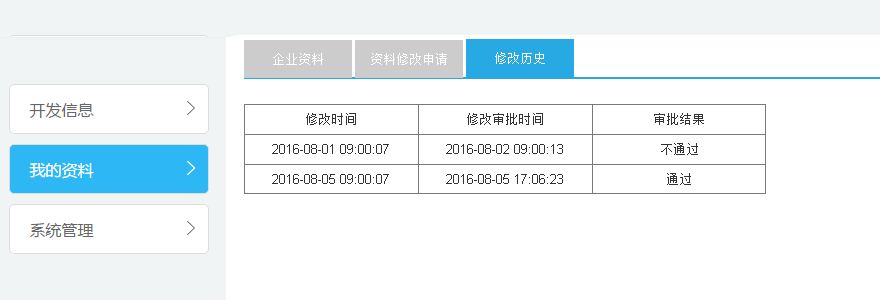
1. 需求改动:

在”我的资料”下, 分为3个标签页,分别为”企业资料”(商户正在生效的企业资料),”资料修改申请”(正在提交审核的资料)以及”修改历史”(屡次修改资料的主要信息).

一期图:



二期图:



2.数据库改动:

新增”企业资料修改历史记录”(`merchant\_info\_log`)数据表;

CREATE TABLE `merchant\_info\_log` (

`merchant\_id` int(11) DEFAULT NULL COMMENT '关联merchant\_info表的merchant\_id键',

`merchant\_name` varchar(64) DEFAULT NULL COMMENT '商家名称',

`area\_id` varchar(6) DEFAULT NULL,

`member\_id` varchar(32) DEFAULT NULL,

`company\_type` tinyint(1) DEFAULT '0' COMMENT '企业类型 (1象翌内部, 2象翌外部,3翌商云) ',

`actual\_name` varchar(64) DEFAULT NULL,

`reg\_address` varchar(512) DEFAULT NULL,

`work\_address` varchar(512) DEFAULT NULL,

`contact\_name` varchar(64) DEFAULT NULL,

`contact\_mail` varchar(72) DEFAULT NULL,

`contact\_mobile` varchar(64) DEFAULT NULL,

`contact\_telphone` varchar(512) DEFAULT NULL,

`lisence\_image` varchar(72) DEFAULT NULL,

`org\_image` varchar(72) DEFAULT NULL,

`apply\_user` varchar(72) DEFAULT NULL,

`apply\_ts` datetime DEFAULT NULL,

`audit\_status` tinyint(1) DEFAULT '1' COMMENT '是否签约 ( 1 资料已提交，待审核，2 审核通过，已签约， 3 审核不通过 )',

`audit\_ts` datetime DEFAULT NULL,

`audit\_user` varchar(72) DEFAULT NULL,

`audit\_reason` varchar(200) DEFAULT NULL COMMENT '签约审核通过与否的原因',

`account` varchar(200) DEFAULT NULL COMMENT '提交审核的登入邮箱',

`admin\_op` tinyint(1) DEFAULT '0' COMMENT '是否管理员操作',

KEY `merchant\_id` (`merchant\_id`),

CONSTRAINT `merchant\_info\_action\_ibfk\_2` FOREIGN KEY (`merchant\_id`) REFERENCES `merchant\_info` (`merchant\_id`) ON DELETE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

3.代码改动:

\Home\Controller\MerchantInfoController->merchantInfoUpdate()

当商户修改”我的资料”中的信息时,添加一条数据到`merchant\_info\_log`

(与`merchant\_info\_action`完全一致,但后者可能是更新,前者总是新增);

\Home\Controller\MerchantInfoController->getMerchantLog()

返回与merchant\_id相关资料修改历史记录,支持分页

### 1.2管理平台

1. 需求改动:

在”企业资料修改审核”的每条数据的”详情”下, 分为3个标签页,分别为”待审核资料”(正在提交审核的资料),”资料修改详情”(对比列出与正在生效的企业资料的有差异项)以及”修改历史”(屡次修改资料的主要信息).

一期图:



二期图:



1. 数据库改动:

同上

1. 代码改动:

\Operate\Controller\MerchantController->merchantEdit()

当管理员编辑商户信息时,添加一条数据到`merchant\_info\_log`

\Operate\Controller\MerchantInfoActionController->merchantActionDetail()

返回项中新增[‘livekey\_change’],标识哪些信息是有改变的

\Operate\Controller\MerchantInfoActionController->getMerchantLog()

返回与merchant\_id相关资料修改历史记录,支持分页

\Operate\Controller\MerchantInfoActionController->merchantActionVerify()

当管理员更改审核状态时,同时修改`merchant\_info\_log`中的状态(每条数据只应修改一次)

## 配置管理模块

相关控制器:

\Operate\Controller\SysConfigController(运营管理)

相关数据模型:

\Home\Model\SysConfigModel(企业门户)

\Operate\Model\SysConfigModel(运营管理)

### 2.1企业门户

原读取配置方法,如: C('REFUNDDAY') 改变为 D(‘SysConfigModel’)->REFUNDDAY

### 2.2管理平台

1. 需求改动:

将原本以配置文件形式存放的部分配置信息,改为数据库存储

1. 数据库改动:

新增”系统配置信息” (`sys\_config`)数据表;

CREATE TABLE `sys\_config` (

`sys\_config\_id` int(1) NOT NULL AUTO\_INCREMENT COMMENT '系统配置表主键ID ',

`config\_name` varchar(24) DEFAULT '' COMMENT '配置项名称',

`config\_key` varchar(24) DEFAULT '' COMMENT '配置项键',

`config\_value` varchar(2000) DEFAULT '' COMMENT '配置项取值',

`type` TINYINT(1) DEFAULT 1 COMMENT '配置项类型,1:一般系统配置,2:邮件内容配置,3:短信内容配置',

`apply\_user` varchar(72) DEFAULT 'system' COMMENT '配置更改人',

`modify\_ts` datetime DEFAULT '2000-01-01' COMMENT '配置更改时间',

PRIMARY KEY (`sys\_config\_id`),

UNIQUE KEY `config\_name` (`config\_name`),

UNIQUE KEY `config\_key` (`config\_key`)

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

1. 代码改动:

\Operate\Controller\SysConfigController->getList()

返回当前系统下的配置信息

\Operate\Controller\SysConfigController->updateConfig()

更新信息到数据库及缓存

## 短信与邮件模版管理模块

相关控制器:

\Operate\Controller\SysConfigController(运营管理)

相关数据模型:

\Home\Model\SysConfigModel(企业门户)

\Operate\Model\SysConfigModel(运营管理)

### 3.1企业门户

原读取配置方法,如: C('REFUNDDAY') 改变为 D(‘SysConfigModel’)->REFUNDDAY

### 3.2管理平台

1.需求改动:

(1).将原本以配置文件形式存放的短信与邮件模版信息,改为数据库存储



1. .邮件均有禁用启用控制



1. 数据库改动:

将原以配置文件形式存放的短信与邮件模版信息,放置于`sys\_config`数据表

1. 代码改动:

\Operate\Controller\SysConfigController->getListMail()

返回当前系统下的邮件配置信息列表

\Operate\Controller\SysConfigController->getListMessage()

返回当前系统下的短信配置信息列表

\Operate\Controller\SysConfigController->updateConfig()

更新信息到数据库及缓存

## 4.异常订单管理模块

相关控制器:

\Home\Controller\AbnormalRequestController(企业门户)

\Home\Controller\AppPayListController(企业门户)

\Operate\Controller\AbnormalRequestController(运营管理)

\Operate\Controller\PayRequestController(运营管理)

相关数据模型:

\Home\Model\AbnormalRequestModel(企业门户)

\Home\Model\PayRequestModel(企业门户)

\Operate\Model\AbnormalRequestModel(运营管理)

\Operate\Model\PayRequestModel(运营管理)

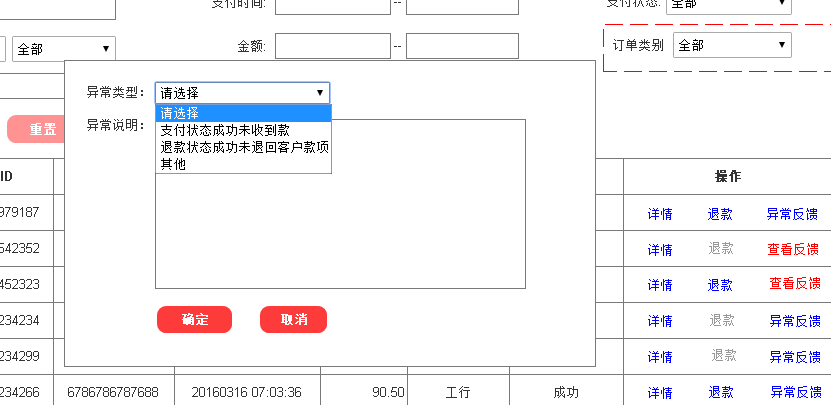
### 4.1企业门户

1.需求改动:

(1).原订单管理模块,增加异常订单筛选, 且每的订单依据异常状态展示不同的异常按钮(3种)



1. .依据不同的异常(状态)按钮,触发不同的弹窗操作



1. .新增单独的异常订单管理页面
2. 数据库设计

(1).在原订单表新增字段`abnormal\_status`表示订单异常状态,0:无异常,1:异常已反馈,2:异常已处理;

(2).新增 ”异常订单信息记录表” (`abnormal\_request`)

DROP TABLE IF EXISTS `abnormal\_request`;

CREATE TABLE `abnormal\_request` (

`pay\_request\_id` int(1) NOT NULL COMMENT '关联pay\_request表的pay\_request\_id键',

`apply\_user` varchar(72) NOT NULL DEFAULT '',

`apply\_ts` datetime NOT NULL DEFAULT '2000-01-01',

`apply\_reason` varchar(2000) NOT NULL DEFAULT '' COMMENT '申请异常描述',

`audit\_user` varchar(72)NOT NULL DEFAULT '' COMMENT '异常处理人',

`audit\_ts` datetime NOT NULL DEFAULT '2000-01-01' COMMENT '异常处理时间',

`audit\_reason` varchar(2000) NOT NULL DEFAULT '' COMMENT '异常处理描述',

`abnormal\_type` tinyint(1) DEFAULT '1' NOT NULL COMMENT '订单异类型,0:其他,1:支付状态正常为收到款,2:退款状态正常未退回客户款项',

`abnormal\_status` tinyint(1) DEFAULT '1' NOT NULL COMMENT '订单异常状态,1:异常已反馈,2:异常已解决,3:异常未能解决',

UNIQUE KEY `pay\_request\_id` (`pay\_request\_id`),

CONSTRAINT `abnormal\_request\_ibfk\_1` FOREIGN KEY (`pay\_request\_id`) REFERENCES `pay\_request` (`pay\_request\_id`) ON DELETE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=utf8;

1. 代码改动:

\Home\Controller\AppPayListController->getList()

增加对于`abnormal\_status`字段的接受与筛选,

同时每一条返回数据将包含`abnormal\_status`,

以区分异常状态0:无异常,1:异常已反馈,2:异常已解决,3:异常未能解决

\Home\Controller\AbnormalRequestController->getList()

新增异常订单控制器,支持搜索,分页,仅返回异常订单及状态;

\Home\Controller\AbnormalRequestController->reportAbnormal()

对于`abnormal\_status`=0的订单的异常报告新增

对于`abnormal\_status`=(1||2||3)的订单的异常查看

\Home\Controller\AbnormalRequestController->getAbnormalType()

返回”异常类型”选项的代号及中文释义;

### 4.2管理平台

1.需求改动:

(1).原订单管理模块不做改动,增加独立异常订单模块, 且每的订单依据异常状态展示不同的异常按钮(2种)



1. .依据不同的异常(状态)按钮,触发不同的弹窗操作



1. 数据库设计:

同企业门户

1. 代码改动:

\Operate\Controller\AbnormalRequestController->getList()

新增异常订单控制器,支持搜索,分页,仅返回异常订单及状态;

\Operate\Controller\AbnormalRequestController->dealAbnormal()

对于`abnormal\_status`=1的订单的异常报告处理

对于`abnormal\_status`=(2||3)的订单的异常查看

\Operate\Controller\AbnormalRequestController->getAbnormalType()

返回”异常类型”选项的代号及中文释义;

\Operate\Controller\AbnormalRequestController->exportExcel()

异常订单到导出到Excel;

## 5.对账单管理

### 5.1设计图



### 5.2功能描述

SE支付在T+1通过接口获取到企业会员在第三方支付平台及银行的对账单，保存在SE支付服务器中。企业会员在SE企业门户上最多可下载当前时间及之前三个月内的对账单，更早的对账单不提供下载，但是还会保存在SE平台上。

注：三个月内的账单，如今天19号，往前三个月的19号之前的不能下载

### 5.3输入数据

A、当前应用支持的支付渠道

|  |  |
| --- | --- |
| Input | 说明 |
| merchant\_app\_id | 当前应用id |

B、日账单

|  |  |
| --- | --- |
| Input | 说明 |
| merchant\_app\_id | 当前应用id |
| payment\_type\_id | 支付渠道id |
| created\_ts | 选择日期 |

C、月（某个时间段）账单

|  |  |
| --- | --- |
| Input | 说明 |
| merchant\_app\_id | 当前应用id |
| payment\_type\_id | 支付渠道 |
| begin\_date | 开始日期 |
| end\_date | 结束日期 |

### 5.4输出数据

A、当前应用支持的支付渠道

|  |  |
| --- | --- |
| Output | 说明 |
| payment\_type\_id | 支付渠道id |
| name | 支付渠道名称 |

B、日账单

|  |  |
| --- | --- |
| Input | 说明 |
| Date.csv | 日账单文件或压缩包 |

C、月（某个时间段）账单

|  |  |
| --- | --- |
| Input | 说明 |
| Date.csv | 月账单文件或压缩包 |

### 5.5业务算法和流程

1. 当前应用支持的支付渠道

接口名称：appPayments

前端根据appPayments 接口可以获取支付渠道名称；

企业会员在进入对账单管理页面可以看到当前应用支持的支付渠道

B、日账单

接口名称：getAppDayAccount

企业会员在对账单管理页面可选择具体某一天的账单详情

C、月（某个时间段）账单

接口名称：getAppMonthAccount

企业会员选择查询日期起止点，即可下载对应日期段的账单详情

### 5.6函数说明

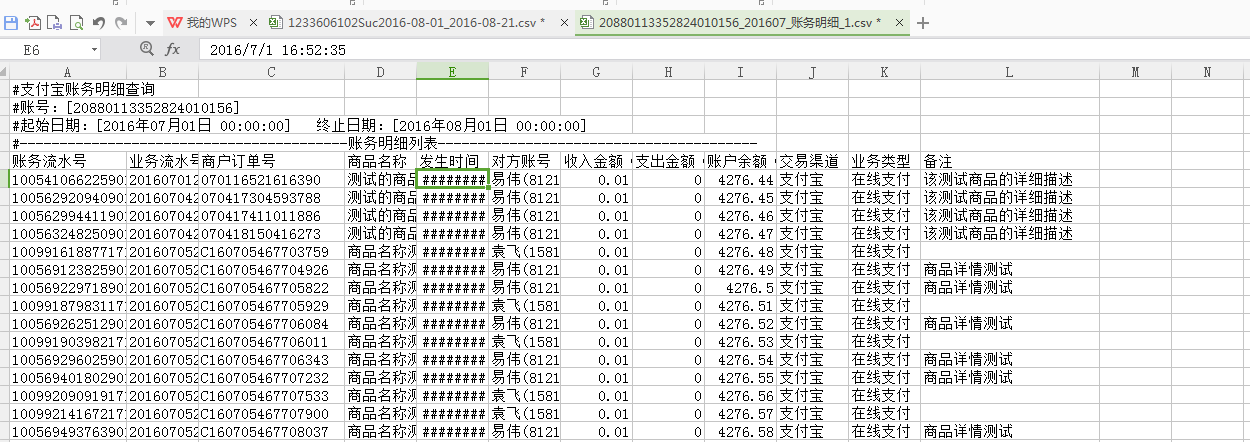
//获取当前应用支持的支付渠道

public function appPayments(){}

//根据第三方接口，获取各种支付渠道的账单

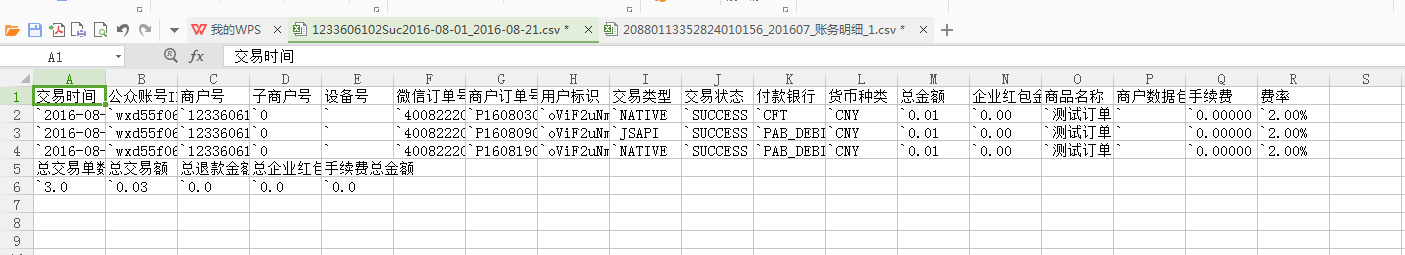
//获取支付宝账单（官方：一个是账务明细分页查询，一个是单笔交易查询，都需要签约才能给文档）

Public function getAlipayAccount(){}



//获取微信支付账单（官方：对账单接口只能下载三个月以内的账单，单天的）

Public function getWechatAccount(){}



//获取银联账单

Public function getChinapayAccount(){}