

食方科技智慧餐台支付对接说明

修订记录

日期	作者	版本	说明
2021年1月13日	韩志林	v1.0	支付对接说明
2021年5月11日	韩志林	v1.1	完善SDK对接说明
2021年6月22日	韩志林	v1.2	增加异步调用接口
2021年8月6日	韩志林	v1.3	增加退款接口
2021年9月24日	韩志林	v1.4	支付接口增加设备id和收银员id；接口参数描述完善；错误码描述；

Http对接

Http对接方式主要针对餐台和支付设备（如：刷卡机、扫码器、人脸机）统一采用食方解决方案的场景，支付服务提供方只需要实现订单支付、订单取消和订单状态查询接口即可，餐台客户端可以设置支付服务域名。

订单支付接口

当餐台发起支付请示时，餐台首先会唤起支付设备接收用户卡、码、脸信息，接收到支付信息后，会调用支付接口完成支付。

请求URL

- [域名]/dish/order/pay

请求方式

- POST

解释

- Y->必传/必反
- N->非必传/必反

参数

Header参数：

参数	值
Content-Type	application/json

Body请求参数：

参数名称	必填	类型和范围	说明
order_id	Y	string	餐台生成的订单id
amount	Y	int	实付金额，单位：分
discount	Y	float	折扣率，如：0.8
pay_type	Y	int	支付方式：1-物理卡，2-付款码，3-现金，4-会员编号，5-人脸，7-支付宝，8-微信，9-合作支付
pay_id	Y	string	支付id，与pay_type对应，如pay_type传1，则pay_id传物理卡号；人脸支付对应的pay_id，按照实际人脸绑定的账号类型解析即可，通常人脸会绑定账号id或会员编号。
device_id	N	string	设备id
cashier_id	N	string	收银员id
nonce_str	Y	string	随机串，最长32位
create_time	Y	string	订单创建时间
dishes	N	array	订单明细
+name	N	string	菜品名称
+price	N	float	菜品价格，单位：元

请求示例：

返回说明

接口需返回支付结果

返回参数

序号	参数名	是否必填	类型	注释
1	status	Y	int	状态码：成功返回200，其他返回错误码，见文末《错误码》章节
2	msg	Y	string	返回结果描述，失败，返回错误原因
3	data	Y	json	结果数据
4	+order_id	Y	string	餐台生成的订单id
5	+user_id	Y	string	用户编码
6	+trans_id	Y	string	支付订单号
7	+cash_fee	Y	int	实付金额，单位：分
8	+balance	Y	int	账户余额：单位：分

返回示例：

```
{
  "status": "200",
  "msg": "请求成功",
  "data": {
    "orderid": "2018051223252",
    "trasn_id": "1220221515",
    .....
  }
}
```

订单取消接口（无需实现：http对接，不允许取消）

请求URL

- [域名]/dish/order/cancel

请求方式

- POST

解释

- Y->必传/必反
- N->非必传/必反

参数

Header参数：

参数	值
Content-Type	application/json

Body请求参数：

参数名称	必填	类型和范围	说明
order_id	Y	string	餐台生成的订单id

请求示例：

返回说明

序号	参数名	是否必填	类型	注释
1	status	Y	int	状态码：成功返回200，其他返回错误码，见文末《错误码》章节
2	msg	Y	string	返回结果描述，失败，返回错误原因

返回示例：

```
{
  "status": "200",
  "msg": "取消成功",
}
```

订单状态查询接口

请求URL

- [域名]/dish/order/status/{order_id}

请求方式

- GET

解释

- Y->必传/必反
- N->非必传/必反

参数

Header参数：

参数	值
Content-Type	application/json

请求参数：

参数名称	必填	类型和范围	说明
order_id	Y	string	餐台生成的订单id
device_id	N	string	设备id

请求示例:

```
{
  "amount": 1000,
  "cashier_id": null,
  "create_time": "2021-09-24 16:22:45",
  "device_id": null,
  "discount": 1,
  "dishes": [
    {
      "name": "定价商品",
      "price": 10
    }
  ],
  "nonce_str": "bc86f64267868d317f4546289273d0e2",
  "order_id": "20210924162245520",
  "pay_id": "88888888",
  "pay_type": 1
}
```

返回参数

序号	参数名	是否必填	类型	注释
1	status	Y	int	状态码：成功返回200， 错误返回204
2	msg	Y	string	返回结果描述， 失败， 返回错误原因
3	data	Y	json	结果数据
4	+order_status	Y	int	订单状态（0：等待交易， 1：交易成功， 2：交易失败）
5	+order_id	Y	string	餐台生成的订单id
6	+user_id	Y	string	用户编码
7	+trans_id	Y	string	支付订单号
8	+cash_fee	Y	int	实付金额， 单位： 分
9	+balance	Y	int	账户余额： 单位： 分

返回示例:

```
{
  "status": "200",
  "msg": "请求成功",
  "data": {
    "order_id": "2018051223252",
    "order_status": "1",
    .....
  }
}
```

退款

请求URL

- [域名]/dish/order/refund

请求方式

- POST

解释

- Y->必传/必反
- N->非必传/必反

参数

Header参数：

参数	值
Content-Type	application/json

请求参数：

参数名称	必填	类型和范围	说明
order_id	Y	string	退款订单号（餐台生成），注：非原订单号
trans_id	N	string	原始支付订单号（支付平台生成）
amount	Y	int	退款金额，单位：分
reason	N	string	退款原因

请求示例：

返回参数

序号	参数名	是否必返	类型	注释
1	status	Y	int	状态码：成功返回200，错误返回204
2	msg	Y	string	返回结果描述，失败，返回错误原因
3	data	Y	json	结果数据
4	+order_id	Y	string	退款订单号（与请求参数一致）
5	+trans_id	Y	string	退款产生的新的支付订单号
6	+refund_amount	Y	int	实际退款金额，单位：分

返回示例：

```
{
  "status": "200",
  "msg": "请求成功",
  "data": {
    "order_id": "2018051223252",
    "trans_id": "2018051223666",
    "refund_amount": "200",
    .....
  }
}
```

SDK对接

SDK对接方式主要针对支付设备由支付服务商提供，食方餐台只提供菜品的识别数据的场景。该场景下，支付服务方需要集成食方科技智慧餐台支付SDK，实现与餐台的数据通信。

集成食方智慧餐台支付SDK

SFPaymentSDK（目前只有Windows版本），提供了一个与餐台进行数据通信的服务，负责响应餐台的订单支付和取消订单请求。

SDK集成步骤

- 支付服务商需要集成SDK，按照回调接口的要求实现订单支付和取消订单接口，完成与支付设备的对接

```
namespace SFPayment
{
#ifdef __cplusplus
    extern "C" {
#endif

    static const char Cmd_Payment[] = "payment";//支付请求
    static const char Cmd_Refund[] = "refund";//退款请求
    static const char Cmd_CancelPayment[] = "cancel";//取消支付请求

    //支付相关回调函数：支付、退款、取消订单
```

```

//回调函数即时返回，具体支付结果调用writeData通知sdk
typedef bool(*requestHandler)(const char *jsonReq);

/*!
 * \brief    启动sdk服务
 * \returns  SF_PAYMENTSdk_EXPORT_API bool
 * \remark
 */
SF_PAYMENTSdk_EXPORT_API bool startServer();

/*!
 * \brief    写入支付结果数据
 * \param    const char * result 结果数据[JSON，格式见接口文档]
 * \returns  void
 * \remark
 */
SF_PAYMENTSdk_EXPORT_API void writeData(const char* result);

/*!
 * \brief    支付终止
 * \returns  void
 * \remark    取消支付成功或支付超时调用该接口
 */
SF_PAYMENTSdk_EXPORT_API void aborted();

/*!
 * \brief    注册支付相关回调函数
 * \param    const char * cmd 命令[payment|refund|cancel]
 * \param    requestHandler handler 支付回调函数
 * \returns  void
 * \remark
 */
SF_PAYMENTSdk_EXPORT_API void registerCmdHandler(const char *cmd,
requestHandler handler);

#ifdef __cplusplus
}
#endif
}

```

- 调用SDK接口设置回调方法

```
void registerCmdHandler(const char *cmd, requestHandler handler);
```

- 调用SDK接口启动服务

```
void startServer();
```

- 详见Demo

接口规范

支付数据统一采用JSON格式。接口采用异步调用方式，支付（或退款）结果通过writeData接口写入。

- 订单支付

参数名称	必填	类型和范围	说明
cmd	是	string	payment
orderid	是	string	餐台生成的订单id
data	是		
+amount	是	float	实付金额
+totalAmount	是	float	总金额
+discount	是	float	折扣率
+nonce	是	string	随机串，最长32位
+createtime	是	string	订单创建时间
+dishes	否	array	订单明细
++name	是	string	菜品名称
++price	是	float	菜品价格

- 订单退款

参数名称	必填	类型和范围	说明
cmd	是	string	refund
orderid	是	string	原始餐台订单id
tradeNo	是	string	原始支付订单id
amount	是	float	退款金额

- 取消支付

参数名称	必填	类型和范围	说明
cmd	是	string	cancel
orderid	是	string	餐台生成的订单id

- 返回结果

参数名称	必填	类型和范围	说明
cmd	是	string	与对应请求一致
code	是	string	错误码，见【错误码】章节
message	是	string	成功；失败+原因
data	是	float	退款金额
+orderId	是	string	餐台订单号（餐台生成）
+tradeNo	否	string	支付订单号 【支付 退款】
+payId	否	string	支付id，即卡、码、脸编号 【支付】
+payType	否	int	支付类型：1-刷卡；2-扫码；3-现金；5-人脸 【支付】
+balance	否	float	账户余额 【支付】

错误码

编码	说明
200	成功
201	余额不足，返回当前余额balance
202	支付取消
203	支付超时
204	支付服务异常

DEMO

SFPaymentDemo