下单：

POST http://120.79.77.207:8000/api/pub/payment/order

{

"appId": "demo",

"sign": "67DFA6921F642DAC47B2C62BC88E6DA2",

"title": "标题",

"detail": "详情",

"totalFee": 2.02,

"orderNum": "1235",

"paymentType": "WECHAT",

"tradeType": "WPA",

"notifyUrl": "http://127.0.0.1:8080/api/pub/payment/order/notify",

"customerData": "customer data",

“returnUrl”: “http://www.kequandian.net”,

“sandbox”: “true”

}

参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 必须 | 说明 |
| appId | String | 是 | 分配给应用的ID |
| sign | String | 是 | 签名，签名算法参考下面 |
| title | String | 是 | 商品描述交易字段 |
| detail | String | 是 | 商品详细描述 |
| totalFee | Double | 是 | 订单总金额 |
| orderNum | String | 否 | 业务系统的订单号 |
| paymentType | String | 否 | 支付类型，默认WECHAT |
| tradeType | String | 否 | 交易类型，默认 WPA 公众号支付。  NATIVE: 微信原生扫码  APP: app支付 |
| notifyUrl | String | 否 | 订单状态改变时通知业务系统的回调地址。使用POST方法。 |
| customerData | String | 否 | 业务系统的自定义数据，notify时会原样返回 |
| returnUrl | String | 否 | 业务系统提供的支付后的返回链接 |
| sandbox | String | 否 | 使用沙箱支付。  当提供这个字段时，API直接把returnUrl作为wpayUrl返回，并把账单设置为已支付，notify回调通知业务方。  只有demo appid才有效。 |

返回结果说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 |  | 说明 |
| billNum | String |  | 支付系统的账单编号 |
| wpayUrl | String |  | 使用公众号支付时的支付URL |
| wpayCode | String |  | tradeType 为 NATIVE 时生成的支付码，可以用这个数据生成二维码给用户扫码支付 |

返回例子：

{

"code": 200,

"data": {

"billNum": “1053236925112180738",

"wpayCode": "weixin://wxpay/bizpayurl?pr=RbKqLn7",

"wpayUrl": "http://www.kequandian.net/wechat/payment/wpay?title=%E6%A0%87%E9%A2%98&detail=%E8%AF%A6%E6%83%85&totalFee=2.02&orderNum=demo\_1235&appName=KQD&returnUrl="

},

"message": "操作成功"

}

账单查询接口

GET http://120.79.77.207:8000/api/pub/payment/order?appId=demo&sign=17DEE6374535B1A87921EB0298C26447

参数说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 | 必须 | 说明 |
| appId | String | 是 | 分配给应用的ID |
| sign | String | 是 | 签名，签名算法参考下面 |
| pageNum | Integer | 否 | 页码 |
| pageSize | Integer | 否 | 每页记录数 |
| status | String | 否 | 账单状态  PAID：已支付  PAY\_PENDING： 待支付 |

返回结果说明：

|  |  |  |  |
| --- | --- | --- | --- |
| 字段 | 类型 |  | 说明 |
| appId | String |  | 分配给应用的ID |
| billNum | String |  | 支付系统的账单编号 |
| outOrderNum | String |  | 业务系统传来的订单号，如果下单时没有传，则用billNum填充。 |
| tranId | String |  | 第三方支付系统的交易号 |
| status | String |  | 账单状态  PAID：已支付  PAY\_PENDING： 待支付 |

返回例子：

{

"code": 200,

"data": {

"current": 1,

"pages": 1,

"records": [

{

"appId": "demo",

"billNum": "1053221587691732993",

"createTime": "2018-10-19 09:49:47",

"customerData": "",

"detail": "详情",

"id": "6",

"lastNotifyTime": "",

"notifyAttemptCount": 0,

"notifyResult": 0,

"notifyUrl": "http://127.0.0.1:8080/api/pub/payment/order/notify",

"outOrderNum": "1235",

"paymentType": "WECHAT",

"status": "PAY\_PENDING",

"title": "标题",

"totalFee": 2.02,

"tranId": ""

}

],

"size": 50,

"total": 1

},

"message": "操作成功"

}

支付结果通知

定义一个可以外网直接POST访问到的url，支付系统会post以下内容

{

"appId": "demo",

"orderNum": "ordernumber",

"customerData": "data",

"sign": "XXXXXX"

}

收到通知后，对结果进行签名检查，签名通过了才是支付服务返回的，

需要在这个方法里返回以下数据给支付系统

{

"code": 200

}

签名算法：

对POST的参数（sign除外）按其参数名的字典序升序进行排序，将所有参数按"key=value"格式拼接在一起，再和appCode作为key参数的值使用MD5摘要。

对应给demo appId的appCode是 democode

String stringSignTemp = stringA + "&key=" + appCode;

String sign = md5(stringSignTemp).toUpperCase();

================================================

测试设置账单为支付(admin登录拿到token)

GET http://120.79.77.207:8000/api/payment/bills/pay/{appId}/{orderNum}

admin 登录:

POST http://120.79.77.207:8000/api/oauth/login

{

"account": "admin",

"password": "111111"

}