

- [支付宝接口文档](#)
 - [网关URL:](#)
 - [请求参数](#)
 - [返回的参数](#)

支付宝接口文档

不懂得看支付宝开放平台

<https://opendocs.alipay.com/open/00a0ut?pathHash=b19b288a>

网关URL:

<http://192.168.5.87:10087/alipay/pay>

请求参数

参数名	类型	是否必选	最大长度	描述	实例值
out_trade_no	String	是	64	商家订单号，长度为64位，可以使用随机数生成，中英文不限，但是不能有中文	0a0328fc4132446fa7f98c491558c752
total_amount	BigDecimal	是	(10,2)	订单总金额，数据库中使用decimal，实体类中使用BigDecimal，精确到小数点后两位	88.88

参数名	类型	是否必选	最大长度	描述	实例值
subject	String	是	128	调接口的时候可以随便写，表示订单的标题，一个String字符串	用商品名称代替也可
return_url	String	是	256	支付宝同步回调时用到的地址，也即付款完成之后跳转的前端页面地址,详情见下方	http://内网穿透地址/success
notify_url	String	是	256	支付宝异步回调时用到的地址，业绩用来查询订单，并修改订单状态和付款时间的地址，详情见下方	http://内网穿透地址/自己本地项目的接口地址

前三个没什么好说的，下面详细说说return_url和notify_url

1.在本地调接口时，需要使用能做内网穿透的工具，如"路由侠"

return_url为前端地址+端口号

notify_url为后端接口+端口号

将return_url和notify_url修改为实例值得样式，连同上面的三个参数再传给网关

2.如果部署到服务器上，则项目和网关会被nginx统一管理，此时为nginx得统一网关地址

此时的ip地址为：m.jnbat.com:8080

修改return_url, notify_url时，不止需要修改修改ip地址，还需要在后面加上自己打包的项目名，后端需要加上项目名。

连同上面的三个参数再传给网关

返回的参数

返回的参数为code, data, msg的方式，想想办法拿到data里面的数据

data里面的数据是一个form表单，让浏览器自己打开

```
@NoArgsConstructor
@AllArgsConstructor
@ToString
@Data
public class Result<T> {

    private String code;
    private String msg;

    private T data;

}
```

尽量别用lombok，用getter和setter

网关返回的data里面的数据

```
<form name="punchout_form" method="post" action="https://openapi-
sandbox.dl.alipaydev.com/gateway.do?charset=utf-
8&method=alipay.trade.page.pay&sign=VSKj2n2q2dPja7ctHa6UIz2DwFqKjZ8o1vU10oZpxsQyMd3ZPtCV5
AKVGOgHBuQIQX043jNfW4QHn4eYOH%2BQieG1Ep7FY8spIoInClyGYBrK3pD68wpX7IsqD6obTqy2ghN9KSbERv9f
IOWBQsHG%2BmvXQ5715veZOUaYokC6nymk1zzcTC4LfJDW7uQpqKkE%2B4QRWGDyr4A4n21C0dB0cbQi8UAvYX2Ud
sc%2F77%2FzmjIWkSNBzpFnfkZBwnV7A3cbNTnCKxB96Erxsh4E82u%2Bv7kyuiQhLPXFj4HdGXi4FngB%2BUnYU8
PP8z7a%2FH2Nyh2hkNIggWN20lhEhSdR6F678Q%3D%3D&return_url=http%3A%2F%2F127.0.0.1%3A8080%2FS
uccess&notify_url=http%3A%2F%2Fgjg115109920.e1.luyouxia.net%3A24764%2Falipay%2Fnotify&ver
sion=1.0&app_id=9021000122691707&sign_type=RSA2&timestamp=2023-07-
05+17%3A40%3A17&alipay_sdk=alipay-sdk-java-dynamicVersionNo&format=json">
<input type="hidden" name="biz_content" value="
{"notify_url":"http://gjg115109920.e1.luyouxia.net:24764/alipay/notify","out_trade_no":"5
f0bf8eed3ed473cb14cba616da7686d","product_code":"FAST_INSTANT_TRADE_PAY","redisNo":"alipa
y:5f0bf8eed3ed473cb14cba616da7686d","return_url":"http://127.0.0.1:8080/Success","subject
":"fdsdfdgddgh","total_amount":0.10}">
<input type="submit" value="立即支付" style="display:none" >
</form>
<script>document.forms[0].submit();</script>
```

代码实例

以下是前端调insertOrder接口插入订单，并返回给浏览器一个地址，打开这个地址，浏览器会自动请求后端接口，再由后端接口向网关发送请求，拉起支付宝支付页面

```
/**
 * 接受订单数据
 * @return
 */
@PostMapping("/insertOrder")
public Result insertOrder(@RequestBody OrderReq orderReq, HttpServletRequest request)
throws IOException {
    String token = request.getHeader("token");
    String userName = Jwtutils.getTokenName(token);
    String addrMsg = orderReq.getAddrMsg();
    Double totalPrice = orderReq.getTotalPrice();
    int status = orderReq.getStatus();
    String orderNo = UUID.randomUUID().toString().replace("-", "").toLowerCase();

    Result result = orderService.insertOrder(userName,
        orderReq.getGoodId(), orderReq.getGoodsName(),
        orderReq.getNum(), addrMsg, totalPrice, status, orderNo);

    String url = "http://127.0.0.1:8085/market/toPay"+"?out_trade_order="+orderNo;

    result.setData(url);

    return result;
}
```