

- 支付宝网关
 - 请求支付表单字符串
 - 异步回调支付结果

支付宝网关

请求支付表单字符串

网关接口: <http://m.jnbat.com:8080//PayGateway/alipay/pay>

请求方式: post

ContentType: application/json

传递参数:

参数名称	参数类型	是否必选	最大长度	参数说明	备注
subject	String	必选	256	订单标题, 注意: 不可使用特殊字符, 如 /, =, & 等。	中英文不限
out_trade_no	String	必选	64	商户订单号, 由商家自定义, 64个字符以内 仅支持字母、数字、下划线且需保证在商户端不重复。	不可重复
total_amount	BigDecimal	必选	11	订单总金额, 单位为元, 精确到小数点后两位 取值范围: [0.01,1000000000]	取值范围内即可
return_url	String	必选	256	支付成功后前端将要跳转的指定页面, HTTP/HTTPS开头字符串	见下方注

参数名称	参数类型	是否必选	最大长度	参数说明	备注
notify_url	String	必选	256	支付宝服务器主动通知商户服务器里指定的页面http/https路径。	见下方注

接收类型：见实体1

注：服务器访问域名：m.jnbat.com

1. return_url和notify_url 在本地测试时要做内网穿透，用于网关回调；
2. 在部署到服务器时，notify_url的ip和port修改为http://m.jnbat.com:8080/项目名称/接口名
3. 在部署到服务器时，return_url修改为http://m.jnbat.com/前端路由path
4. 注意部署到服务器时，前端访问后端的方式会发生变化，不要再使用localhost和127.0.0.1了，尽量使用域名访问
5. 接4，因为前端和后端都将被nginx管理，所以前端访问后端需要特别注意

发送请求代码示例：

```
public String pay(AliPayDTO dto) throws AlipayApiException {
    String url = "http://m.jnbat.com:8080//PayGateway/alipay/pay";
    AliPay aliPay = INSTANCE.toEntity(dto);
    aliPay.setNotify_url(notifyUrl);
    aliPay.setReturn_url(returnUrl);
    String bizContent = JSON.toJSONString(aliPay);
    HttpHeaders headers = new HttpHeaders();
    headers.setContentType(MediaType.APPLICATION_JSON);
    HttpEntity<String> requestEntity = new HttpEntity<>(bizContent, headers);
    ResponseEntity<AliPayVO> responseEntity = restTemplate.postForEntity(url,
requestEntity, AliPayVO.class);
    AliPayVO aliPayVO = responseEntity.getBody();
    if ("200".equals(aliPayVO.getCode())) {
        System.out.println(aliPayVO.getData());
        System.out.println("调用成功");
    } else {
        System.out.println("调用失败");
    }
    return aliPayVO.getData();
}
```

返回结果：

```
{
    "code":200,
    "msg":"success",
    "data":"<form action=\"https://openapi.alipaydev.com/gateway.do\"
method=\"post\">\r\n<input type='hidden' name='app_id'
value='2016102600761371'/>\r\n<input type='hidden' name='biz_content'
value='{\"out_trade_no\": \"admin1607046479934\", \"product_code\": \"FAST_INSTANT_TRA
DE_PAY\", \"total_amount\": \"1699.0\", \"subject\": \"retailers_pay\"}' />\r\n<input
type='hidden' name='charset' value='utf-8'/>\r\n<input type='hidden' name='method'
value='alipay.trade.page.pay'/>\r\n<input type='hidden' name='notify_url'
value='http://129.211.1.159:8080/zfb/acceptId'/>\r\n<input type='hidden'
name='return_url' value='http://localhost:8080/retailers/transit.html'/>\r\n<input
type='hidden' name='sign'
value='nsq7WyFJ26P/808ydUNO9rBz7YXtISR5CX0nLJ1mUWTos1eNXy54s+kwzzz16SvZvugXvh+a0qt/
7bz3IyRheO6gd2A52qqkQN2gaoev6Hy8B6TMdn1nW8g7Q8NWQ3mDpPucpQVH+IT7HcfIrP7cPvMXs4SrKfi
5Kvw20A7SeZA/Zu306f1oCF381xtnTldWC8MZxPaBbYrp4WqT/yjZl4kEi90G+u0xKkQCCzpT8oUAGVzdwb
PHhge0kRDj01ck3I748Xd17im182A+yJ6IDVcCcw1l15hb5mLz6KY4WEB1T2JDSESp5oPk5M9JlhFDfKItI
/x9UnqeQshj8sI4mg==' />\r\n<input type='hidden' name='sign_type'
value='RSA2'/>\r\n<input type='hidden' name='timestamp' value='2020-12-10
12:42:47'/>\r\n<input type='hidden' name='version' value='1.0'/>\r\n<input
type='submit' value='支付'/>\r\n<script
type='text/javascript'>document.forms[0].submit()</script></form>"
}
```

将data的数据展示到页面后，通过 `document.forms[0].submit()` 自动跳转到支付页面

异步回调支付结果

网关会根据notify_url来返回支付状态信息，请在本地测试一下

接口方式：post

接收类型：实体1，此时T的类型为实体2

根据接收数据修改数据库即可

实体1：

```
package com.bobo.base;

public class Result<T> {
    //判断是否成功
    private String code;
    private String msg;
    private T data;
```

```

    public String getCode() {
        return code;
    }

    public void setCode(String code) {
        this.code = code;
    }

    public String getMsg() {
        return msg;
    }

    public void setMsg(String msg) {
        this.msg = msg;
    }

    public T getData() {
        return data;
    }

    public void setData(T data) {
        this.data = data;
    }
}

```

实体2:

```

package com.bobo.pojo.dto;

import java.math.BigDecimal;
import java.util.Date;

public class AlipayNotifyDTO {

    private String out_trade_no;//订单编号

    private BigDecimal receipt_amount;//实付金额

    private Date gmt_payment;//支付时间

    public String getOut_trade_no() {
        return out_trade_no;
    }

    public void setOut_trade_no(String out_trade_no) {
        this.out_trade_no = out_trade_no;
    }

    public BigDecimal getReceipt_amount() {
        return receipt_amount;
    }

    public void setReceipt_amount(BigDecimal receipt_amount) {

```

```
        this.receipt_amount = receipt_amount;
    }

    public Date getGmt_payment() {
        return gmt_payment;
    }

    public void setGmt_payment(Date gmt_payment) {
        this.gmt_payment = gmt_payment;
    }

}
```