

# 支付功能

电商项目有很多特点，但是支付绝对非常必要的条件。没有支付功能的项目不可称为电商项目。对于目前市面上电商项目占绝大多数的情况下，学习支付功能是有一定的竞争优势的。

## 微信支付

支付场景分两种：

- 1、手机APP支付
- 2、电脑网站支付

涉及到内网穿透：natapp

## 支付配置流程

### 1、下载微信的SDK

- a、可以官网上下载SDK
- b、可以maven导入。

### 2、继承WXPayConfig类，配置账号信息

```
package com.github.wxpay.sdk;

import java.io.InputStream;

/**
 * 微信支付的配置类，包含收款商家的账号ID等信息
 *
 * 注意：
 * 下面的这些账号和ID是需要公司注册的
 */
public class QianfengWXPayConfig extends WXPayConfig {
    /**
     * 商家注册微信支付之后会生成一个应用ID
     * @return
     */
    String getAppID() {
        return "wx632c8f211f8122c6";
    }
}

/**
```

```

    * 商户ID
    * @return
    */
    String getMchID() {
        return "1497984412";
    }

    String getKey() {
        return "sbNCm1JnevqI36LrEaxFwcaT0hkGxFnC";
    }

    InputStream getCertStream() {
        return null;
    }

    IWXPayDomain getWXPayDomain() {
        QianfengWXDomain wxDomain = new QianfengWXDomain();
        return wxDomain;
    }
}

```

### 3、实现IWXPayDomain接口，配置支付的域名。

```

package com.github.wxpay.sdk;

public class QianfengWXDomain implements IWXPayDomain {

    public void report(String domain, long elapsedTimeMillis, Exception ex) {

    }

    /**
     * 返回微信的域名
     * @param config 配置
     * @return
     */
    public DomainInfo getDomain(WXPayConfig config) {
        /**
         * 参数1: 微信支付域名
         * 参数2: 是否主域名
         */
        DomainInfo domainInfo = new DomainInfo("api.mch.weixin.qq.com", true);
        return domainInfo;
    }
}

```

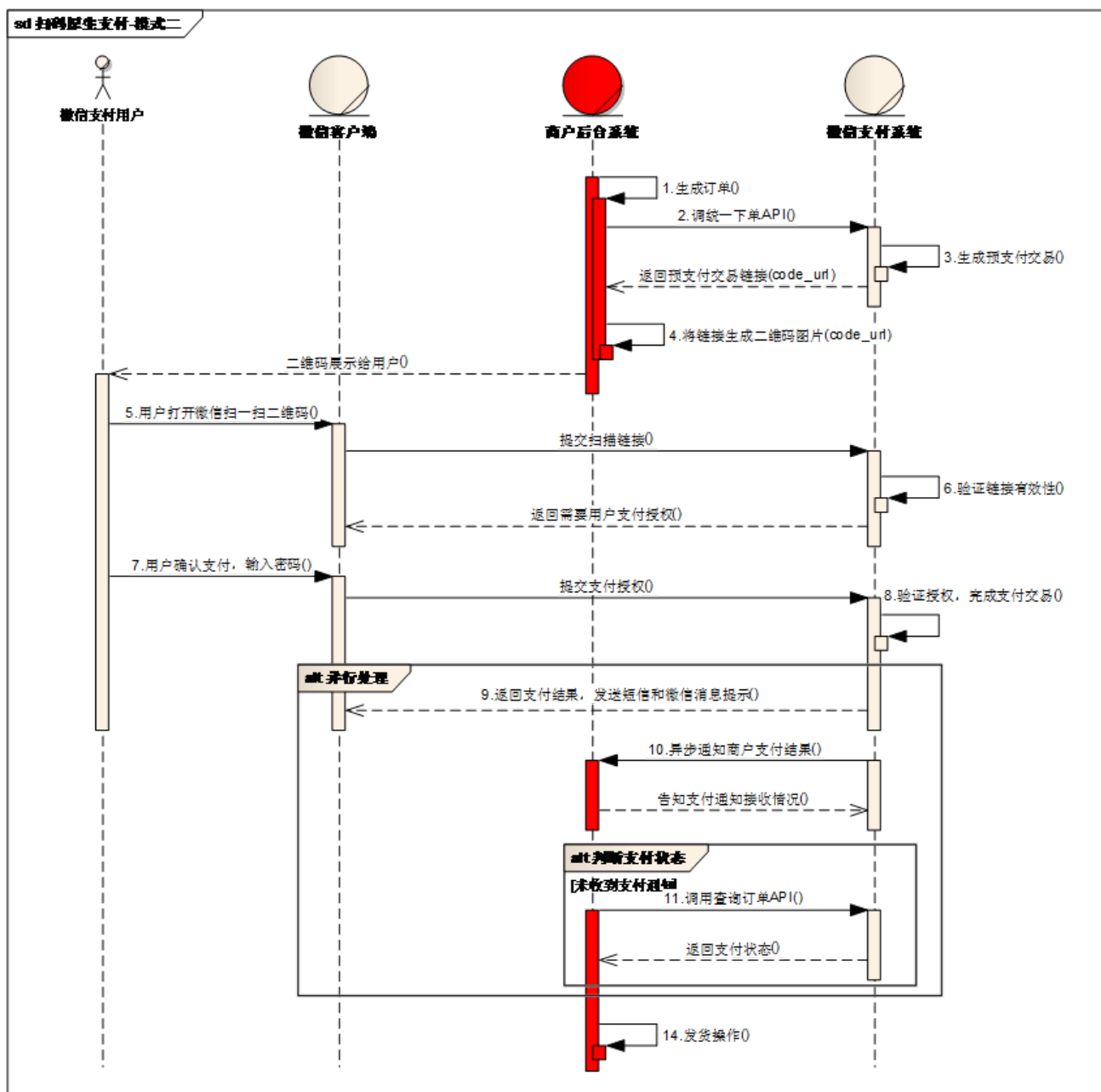
### 4、下单。

- a、在自己系统中下单
- b、再在微信系统中下单。
- c、生成二维码。

```
QianfengWXPAYConfig payConfig = new QianfengWXPAYConfig();
try {
    QFWXPAYConfig payConfig = new QFWXPAYConfig();
    WXPAY wxPay = new WXPAY(payConfig);
    //下单
    Map<String,String> data = new HashMap<String,String>();
    data.put("body","腾讯充值中心-QQ会员充值");//商品标题
    data.put("out_trade_no","2019090910595900000012");//订单编号
    data.put("device_info","WEB");//设备信息
    data.put("fee_type","CNY");//单位: 分
    data.put("total_fee","1");//支付金额
    data.put("spbill_create_ip","123.12.12.123");//终端ip ,记录ip,可以发现攻击我们的ip并进
行屏蔽

    //重点: 回调地址, 用来通知支付结果的地址
    data.put("notify_url","http://localhost:8090/wx/pay_notify_url");
    data.put("trade_type","NATIVE");
    data.put("product_id","12");
    //返回值: 下单成功之后的支付地址
    Map<String, String> order = wxPay.unifiedOrder(data);
    System.out.println(order);

} catch (Exception e) {
    e.printStackTrace();
}
```



## Zxing二维码生成

### 1、导包

```
<!-- https://mvnrepository.com/artifact/com.google.zxing/core -->
<dependency>
  <groupId>com.google.zxing</groupId>
  <artifactId>core</artifactId>
  <version>3.3.3</version>
</dependency>
<!-- https://mvnrepository.com/artifact/com.google.zxing/javase -->
<dependency>
  <groupId>com.google.zxing</groupId>
  <artifactId>javase</artifactId>
  <version>3.3.3</version>
</dependency>
```

## 2、生成二维码

```
@RequestMapping("/qrcode")
public void qrcode(HttpServletResponse response) throws URISyntaxException, IOException {
    //二维码需要包含的文本内容
    String uri = "http://www.baidu.com";
    HashMap<EncodeHintType, Object> hints = new HashMap<>();
    hints.put(EncodeHintType.CHARACTER_SET, "UTF-8");
    hints.put(EncodeHintType.ERROR_CORRECTION, ErrorCorrectionLevel.M);
    hints.put(EncodeHintType.MARGIN, 2);
    try {
        BitMatrix bitMatrix = new MultiFormatWriter().encode(uri, BarcodeFormat.QR_CODE,
200, 200, hints);
        MatrixToImageWriter.writeToStream(bitMatrix, "PNG", response.getOutputStream());
        System.out.println("创建二维码完成");
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

## Goeasy实现异步支付结果

<http://www.goeasy.io/>