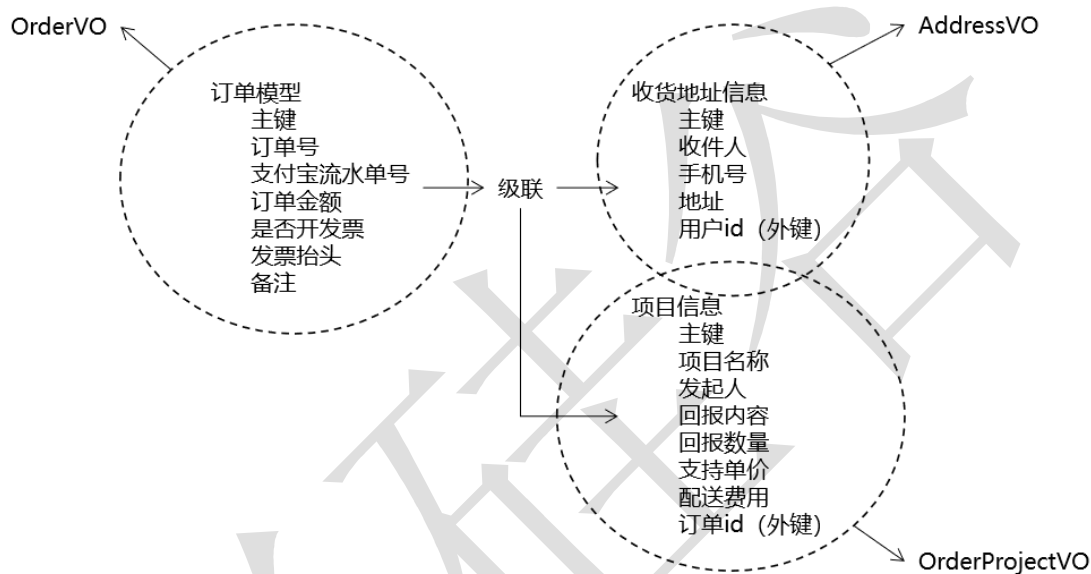


尚筹网

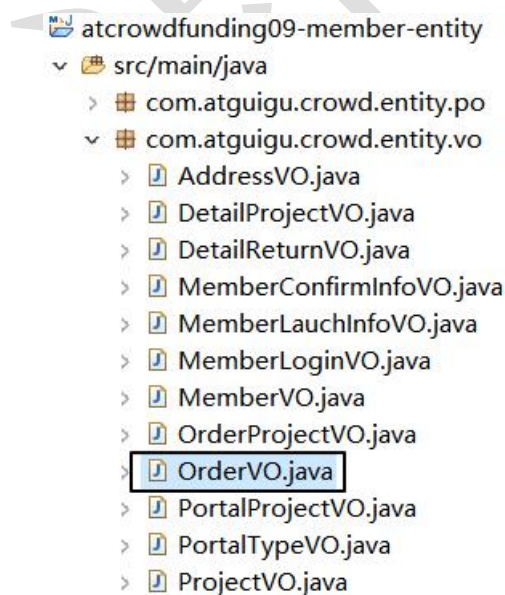
[21-尚硅谷-尚筹网-前台-支付]

1 订单部分建模回顾

1.1 实体类之间关系



1.2 创建 OrderVO 实体类



@Data

```
@NoArgsConstructor
@AllArgsConstructor
public class OrderVO {

    //    主键
    private Integer id;

    //    订单号
    private String orderNum;

    //    支付宝流水单号
    private String payOrderNum;

    //    订单金额
    private Double orderAmount;

    //    是否开发票
    private Integer invoice;

    //    发票抬头
    private String invoiceTitle;

    //    备注
    private String orderRemark;

    private Integer addressId;

    private OrderProjectVO orderProjectVO;

}
```

1.3 修改 t_order 表

增加 address_id 字段

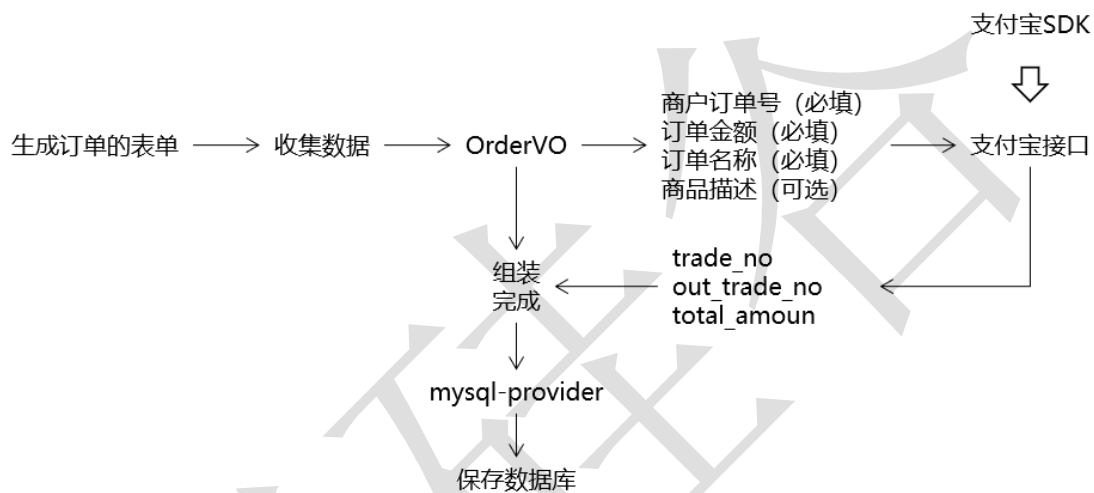
```
CREATE TABLE `project_crowd`.`t_order`
(
    `id` INT NOT NULL AUTO_INCREMENT COMMENT '主键',
    `order_num` CHAR(100) COMMENT '订单号',
    `pay_order_num` CHAR(100) COMMENT '支付宝流水号',
    `order_amount` DOUBLE(10,5) COMMENT '订单金额',
    `invoice` INT COMMENT '是否开发票（0 不开，1 开）',
    `invoice_title` CHAR(100) COMMENT '发票抬头',
```

```
`order_remark` CHAR(100) COMMENT '订单备注',
`address_id` CHAR(100) COMMENT '收货地址 id',
PRIMARY KEY (`id`)
);
```

逆向工程重新生成，用新的资源对旧的资源进行替换。

2 支付功能流程

2.1 思路



2.2 操作第一步：搭建环境

2.2.1 基础环境

参照 order-consumer

2.2.2 支付接口调用所需环境

```
<!-- https://mvnrepository.com/artifact/com.alipay.sdk/alipay-sdk-java -->
<dependency>
  <groupId>com.alipay.sdk</groupId>
  <artifactId>alipay-sdk-java</artifactId>
  <version>3.3.49.ALL</version>
</dependency>
```

2.2.3 创建 PayProperties 类维护支付接口参数

```

atcrowdfunding15-member-pay-consumer [boot] [devtools]
├── src/main/java
│   ├── com.atguigu.crowd
│   ├── com.atguigu.crowd.config
│   │   └── PayProperties.java
│   ├── com.atguigu.crowd.handler
│   └── src/main/resources

```

```
@Data
@NoArgsConstructor
@AllArgsConstructor
@Component
@ConfigurationProperties(prefix = "ali.pay")
public class PayProperties {

    private String appId;
    private String merchantPrivateKey;
    private String alipayPublicKey;
    private String notifyUrl;
    private String returnUrl;
    private String signType;
    private String charset;
    private String gatewayUrl;

}
```

2.2.4 配置支付参数



```
atcrowdfunding15-member-pay-consumer [boot] [devtools]
> src/main/java
  src/main/resources
    application.yml
src/test/java
```

```
ali:
  pay:
    alipay-public-key:
MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAt7QC0UZk4KBH3YD+AwXvFpamX9xTAsP
KsXbbRVUR1gFN9syX/Mg4NpaUkIFr5Im9gIOI9Gr7WDyMZ61oSQGttS7S/qf10j8MZOqvLeKguPPEa
qNqm205q4rQk2R5jKMpjDK3lwBJ7PkuaJyJ8YUHz6uS1Wvb1ZxitxLL2SBwTdPIS/HUOpq3Z1A7DEj
Yc3gcvJ5dYMDGfPQQvrkVBZCBYHMcFjuB8rFJ+USeFQevtGNyYj/YTiLrTVnb/Zmu2+bNWm8MfOuIn
2Z+9D5HbFxxQL3XorU1Ud+YMfNjXWzGXEuZ07duk3qM1nhG13LrXQ+SSGvbGUfodzCs25iJaOwl
DAQAB
    app-id: 2016101800716558
    charset: utf-8
    gateway-url: https://openapi.alipaydev.com/gateway.do
    merchant-private-key:
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCYZt9OA6XVPf9YcbDwaPADIFtCzB
j3HbQjI9m/3Zu+xxMdBmq4ljvgxmgyhPDfrYFVZgAbelljkt58hCb0wR/6+0GftclY2bwrDkQ/SiBDY9jq
o9m572+HUSiEyCvndbEKr6zeZ+sLIE1e3lb4zshsmzwei5qUMRkml6/INjh4KW7OeqvsmzNMijCOU
VcQNLf06B4+b3Jky8uBhOAI334KOYE+YEE7/Y9EfDNCHNqC7o5+91obOD0HvfrbFXNF9Q7zdMMVc
FzuOrhMRMy6ixsQGtcXrLMP47Pj169ow2w4paOAgszR5yW9T4Y+B/2t4mVjfaZzdMLnfdwf8s755Kx
```

```
AgMBAAECggEAe0MI7UJKCDTCp7gckfVt2jl7VoVY09W0co92eJY2d5AguWMNQax+nYgdmBGp+F+
OsZF0GMVQVfANiAwirebdhNNe+lu5hvvkE4mi6GB9w8OoF+86FAayP9HGYc68o34RaUE5lhEOq8sli
kmzuxalJvL25W+8UATPM0Wofw9WcylGo5+rKMvOLzU5vbeeh3SPR6djsm4sYRHKhEJDkqncCOOVP
wx3bulO7tzaSFZc7oCg2jc4vZLQgt0qu4CHRSGS6iyNIQ0N3aY0tamTXi8Se5i6hE0gClqjl+AvpKpvZY0
VtsXVxKMiqaXEaMN59hOicyCJdmmgYxk57Lm65qlcAQKBgQDzWV8ehBnjD2v0oWfn4rr3Df5qNk
ME8LbtDocAb0VMlkmSGLmmxJ2iZVy+i2hWr9PqnzRNO2J5PdctNVj/LR9L2Cjuycapljje6lzm9HNfh
OczrWnG9EGE/ocbGZpyFH9x9corYnbs2kaMzvtNu5KB5QmymEPVLFcBrQ/VsblQKBgQCgUx9uwM
xTEeXVgjis5i6oj7ZGloyh/RHJ/rRomXfUTaV+3xYpljN1KwVeIDQEIVK8jdqVhfrulceEh67RkDUK+QYJZY
2BCy8FnteqUU3juKWUjt27ahgydXivntHHuei9X1ZNCfu3fjeCR/l87NcPK8b3Kq7Sq76Ng67/mAIVkQ
KBgQCqQH2u4Ocn7ub9isEX80wDRdgXoEzyCSVMBuSc0Y6TM5jQotqzAY1CR3G6tlOUTuQH/ZRD3c
7OqdVY8YzPq01hx8KjhaH56e+lzWemyb5YEXvxiGw33w2z2Ut250vJwpODjF28SrA/d1riirEBKlk
O7coCHetg6hcNEqJAK8YQKBgCIDu79h2DU5UhKWJcg6pp+ZOQ0KCsBWxpFbAUw60+DM61+wQ6
AljM4HeKY1W64iUN0/FZeOmHtaLmQqGBDQMRbV6XvZq4YfPscPjyrDKm0GRX6cPu3R7NBv89Qf6
Jajfwe1fOVxyV/UwlxTEGfl+s/tUJAtiZ7FdyOnCglE8RBAoGAL1YRvgQSB8n9kkeLXPMjZF+HS6kec4C
5Xv3vecyl2VJ3/yqcY34QMTi4EURX0eGqUSFX9oP59FAL1Wf8Kvyt/tNV5gpHzqxbj7nG1GFy+IKB4e
mFN3f0ianWIRbbQ0XecY90k5ijdTALfzW+w7Xsgia5h2Rp4rapFbzYOFDiTIY=
    notify-url: http://xxx
    return-url: http://xxx
    sign-type: RSA2
```

说明：notify-url、return-url 等启动内网穿透客户端再填写具体值

2.3 操作第二步：提交订单表单

```
atcrowdfunding14-member-order-consumer [boot]
> src/main/java
  > src/main/resources
    > templates
      > confirm_order.html
      > confirm_return.html
```

2.3.1 构造页面不可见的表单

```
<!-- 为了收集当前页面中的所有数据，构造空表单 -->
<form id="summaryForm" action="pay/generate/order" method="post"></form>
```

2.3.2 给“立即付款”按钮绑定单击响应函数

```
$("#payButton").click(function(){

    // 1.收集所有要提交的表单项的数据
    var addressId = $("[name=addressId]:checked").val();
    var invoice = $("[name=invoiceRadio]:checked").val();
    var invoiceTitle = $.trim($("[name=invoiceTitle]").val());
    var remark = $.trim($("[name=remark]").val());
```

```
// 2.将上面收集到的表单数据填充到空表单中并提交
$("#summaryForm")
    .append("<input type='hidden' name='addressId' value='"+addressId+"'>")
    .append("<input type='hidden' name='invoice' value='"+invoice+"'>")
    .append("<input type='hidden' name='invoiceTitle' value='"+invoiceTitle+"'>")
    .append("<input type='hidden' name='orderRemark' value='"+remark+"'>")
    .submit();
});
```

2.4 操作第三步：生成订单号

2.4.1 支付宝对订单号的要求

https://opensecurity.alipay.com/support/knowledge/20582/201602075893?ant_source=zsearch

如何设置商户订单号out_trade_no

参数: out_trade_no

类型: String (64)

描述: 商户订单号,64个字符以内、可包含字母、数字、下划线;需保证在商户端不重复

注:

- 1、该参数值是商户根据要求自定义生成。
 - 2、需保证在商户端不重复,建议格式当前时间+自定义数字。
- 更多内容请参见相关[开放平台](#)接口文档中的请求参数。

2.4.2 格式设计

当前时间(年月日小时分秒)+用户 ID

时间格式: yyyyMMddHHmmss

用户 ID: 通过 UUID 生成

2.4.3 handler 方法

```
atcrowdfunding15-member-pay-consumer
├── src/main/java
│   ├── com.atguigu.crowd
│   ├── com.atguigu.crowd.config
│   └── com.atguigu.crowd.handler
│       └── PayHandler.java
└── src/main/resources
```

// 这里必须加@ResponseBody 注解,让当前方法的返回值成为响应体,在浏览器界面上显示支付宝支付界面

@ResponseBody

```
@RequestMapping("/generate/order")
public String generateOrder(HttpSession session, OrderVO orderVO) throws AlipayApiException {

    // 1.从 Session 域获取 orderProjectVO 对象
    OrderProjectVO orderProjectVO = (OrderProjectVO) session.getAttribute("orderProjectVO");

    // 2.将 orderProjectVO 对象和 orderVO 对象组装到一起
    orderVO.setOrderProjectVO(orderProjectVO);

    // 3.生成订单号并设置到 orderVO 对象中
    // ①根据当前日期时间生成字符串
    String time = new SimpleDateFormat("yyyyMMddHHmmss").format(new Date());

    // ②使用 UUID 生成用户 ID 部分
    String user = UUID.randomUUID().toString().replace("-", "").toUpperCase();

    // ③组装
    String orderNum = time + user;

    // ④设置到 OrderVO 对象中
    orderVO.setOrderNum(orderNum);

    // 4.计算订单总金额并设置到 orderVO 对象中
    Double orderAmount = (double) (orderProjectVO.getSupportPrice() *
    orderProjectVO.getReturnCount() + orderProjectVO.getFreight());
    orderVO.setOrderAmount(orderAmount);

    // 5.调用专门封装好的方法给支付宝接口发送请求
    return sendRequestToAliPay(orderNum, orderAmount, orderProjectVO.getProjectName(),
    orderProjectVO.getReturnContent());
}
```

2.4.4 调用支付宝接口的专门方法

```
/**
 * 为了调用支付宝接口专门封装的方法
 * @param outTradeNo 外部订单号，也就是商户订单号，也就是我们生成的订单号
 * @param totalAmount 订单的总金额
 * @param subject 订单的标题，这里可以使用项目名称
 * @param body 商品的描述，这里可以使用回报描述
 * @return 返回到页面上显示的支付宝登录界面
 * @throws AlipayApiException
```



```
*/
private String sendRequestToAliPay(String outTradeNo, Double totalAmount, String subject,
String body) throws AlipayApiException {
    //获得初始化的 AlipayClient
    AlipayClient alipayClient = new DefaultAlipayClient(
        payProperties.getGatewayUrl(),
        payProperties.getAppId(),
        payProperties.getMerchantPrivateKey(),
        "json",
        payProperties.getCharset(),
        payProperties.getAlipayPublicKey(),
        payProperties.getSignType());

    //设置请求参数
    AlipayTradePagePayRequest alipayRequest = new AlipayTradePagePayRequest();
    alipayRequest.setReturnUrl(payProperties.getReturnUrl());
    alipayRequest.setNotifyUrl(payProperties.getNotifyUrl());

    alipayRequest.setBizContent("{\"out_trade_no\":\""+ outTradeNo +"\", \"
        + \"total_amount\":\""+ totalAmount +"\", \"
        + \"subject\":\""+ subject +"\", \"
        + \"body\":\""+ body +"\", \"
        + \"product_code\":\"FAST_INSTANT_TRADE_PAY\"}");

    // 若想给 BizContent 增加其他可选请求参数，以增加自定义超时时间参数
    timeout_express 来举例说明
    //alipayRequest.setBizContent("{\"out_trade_no\":\""+ out_trade_no +"\", \"
    //    + \"total_amount\":\""+ total_amount +"\", \"
    //    + \"subject\":\""+ subject +"\", \"
    //    + \"body\":\""+ body +"\", \"
    //    + \"timeout_express\":\"10m\", \"
    //    + \"product_code\":\"FAST_INSTANT_TRADE_PAY\"}");
    //请求参数可查阅【电脑网站支付的 API 文档-alipay.trade.page.pay-请求参数】章节

    //请求
    return alipayClient.pageExecute(alipayRequest).getBody();
}
```

2.5 操作第四步：returnUrl 方法

@ResponseBody


```
@RequestMapping("/return")
public String returnUrlMethod(HttpServletRequest request) throws AlipayApiException,
UnsupportedEncodingException {
    // 获取支付宝 GET 过来反馈信息
    Map<String,String> params = new HashMap<String,String>();
    Map<String,String[]> requestParams = request.getParameterMap();
    for (Iterator<String> iter = requestParams.keySet().iterator(); iter.hasNext();) {
        String name = (String) iter.next();
        String[] values = (String[]) requestParams.get(name);
        String valueStr = "";
        for (int i = 0; i < values.length; i++) {
            valueStr = (i == values.length - 1) ? valueStr + values[i]
                : valueStr + values[i] + ",";
        }
        // 乱码解决，这段代码在出现乱码时使用
        valueStr = new String(valueStr.getBytes("ISO-8859-1"), "utf-8");
        params.put(name, valueStr);
    }

    boolean signVerified = AlipaySignature.rsaCheckV1(
        params,
        payProperties.getAlipayPublicKey(),
        payProperties.getCharset(),
        payProperties.getSignType()); //调用 SDK 验证签名

    // ——请在这里编写您的程序（以下代码仅作参考）——
    if(signVerified) {
        // 商户订单号
        String orderNum = new
String(request.getParameter("out_trade_no").getBytes("ISO-8859-1"),"UTF-8");

        // 支付宝交易号
        String payOrderNum = new
String(request.getParameter("trade_no").getBytes("ISO-8859-1"),"UTF-8");

        // 付款金额
        String orderAmount = new
String(request.getParameter("total_amount").getBytes("ISO-8859-1"),"UTF-8");

        // 保存到数据库
        // .....
```

```
        return  
        "trade_no:"+payOrderNum+"<br/>out_trade_no:"+orderNum+"<br/>total_amount:"+orderAmount;  
    }else {  
  
        // 页面显示信息：验签失败  
        return "验签失败";  
    }  
}
```

2.6 操作第五步：notifyUrl 方法

```
@RequestMapping("/notify")  
public void notifyUrlMethod(HttpServletRequest request) throws  
UnsupportedEncodingException, AlipayApiException {  
  
    //获取支付宝 POST 过来反馈信息  
    Map<String,String> params = new HashMap<String,String>();  
    Map<String,String[]> requestParams = request.getParameterMap();  
    for (Iterator<String> iter = requestParams.keySet().iterator(); iter.hasNext();) {  
        String name = (String) iter.next();  
        String[] values = (String[]) requestParams.get(name);  
        String valueStr = "";  
        for (int i = 0; i < values.length; i++) {  
            valueStr = (i == values.length - 1) ? valueStr + values[i]  
                : valueStr + values[i] + ",";  
        }  
        //乱码解决，这段代码在出现乱码时使用  
        valueStr = new String(valueStr.getBytes("ISO-8859-1"), "utf-8");  
        params.put(name, valueStr);  
    }  
  
    boolean signVerified = AlipaySignature.rsaCheckV1(  
        params,  
        payProperties.getAlipayPublicKey(),  
        payProperties.getCharset(),  
        payProperties.getSignType()); //调用 SDK 验证签名  
  
    //——请在这里编写您的程序（以下代码仅作参考）——  
  
    /* 实际验证过程建议商户务必添加以下校验：
```

- 1、需要验证该通知数据中的 `out_trade_no` 是否为商户系统中创建的订单号，
- 2、判断 `total_amount` 是否确实为该订单的实际金额（即商户订单创建时的金额），
- 3、校验通知中的 `seller_id`（或者 `seller_email`）是否为 `out_trade_no` 这笔单据的对应的操作方（有的时候，一个商户可能有多个 `seller_id/seller_email`）

- 4、验证 `app_id` 是否为该商户本身。

```
*/
if(signVerified) { //验证成功
    //商户订单号
    String out_trade_no = new
String(request.getParameter("out_trade_no").getBytes("ISO-8859-1"),"UTF-8");

    //支付宝交易号
    String trade_no = new
String(request.getParameter("trade_no").getBytes("ISO-8859-1"),"UTF-8");

    //交易状态
    String trade_status = new
String(request.getParameter("trade_status").getBytes("ISO-8859-1"),"UTF-8");

    logger.info("out_trade_no="+out_trade_no);
    logger.info("trade_no="+trade_no);
    logger.info("trade_status="+trade_status);

} else { //验证失败
    //调试用，写文本函数记录程序运行情况是否正常
    //String sWord = AlipaySignature.getSignCheckContentV1(params);
    //AlipayConfig.logResult(sWord);

    logger.info("验证失败");
}
}
```

3 订单信息保存到数据库

3.1 思路



3.2 SQL 设置

关键点：保存 OrderPO 时获取自增主键

atcrowdfunding10-member-mysql-provider [boot]

> src/main/java

▼ src/main/resources

▼ mybatis

▼ mapper

AddressPOMapper.xml

MemberConfirmInfoPOMapper.xml

MemberLaunchInfoPOMapper.xml

MemberPOMapper.xml

OrderPOMapper.xml

OrderProjectPOMapper.xml

```
<insert id="insert"
  parameterType="com.atguigu.crowd.entity.po.OrderPO"
  useGeneratedKeys="true" keyProperty="id">
```

```
insert into t_order (id, order_num, pay_order_num,
order_amount, invoice, invoice_title,
order_remark, address_id)
values ({id,jdbcType=INTEGER}, #{orderNum,jdbcType=CHAR},
#{payOrderNum,jdbcType=CHAR},
#{orderAmount,jdbcType=DOUBLE},                #{invoice,jdbcType=INTEGER},
#{invoiceTitle,jdbcType=CHAR},
#{orderRemark,jdbcType=CHAR}, #{addressId,jdbcType=CHAR})
</insert>
```

其他位置参考工程中的代码！