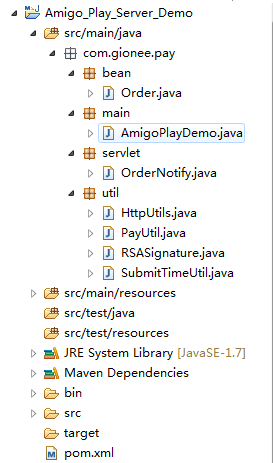
Amigo\_Play\_Server\_Demo说明

# 包结构说明:



Amigo\_Play\_Server\_Demo为maven项目，需要使用maven构建。

包含两个主要的java类

AmigoPlayDemo.java,创建订单

OrderNotify.java，响应支付结果通知。

# 使用说明

**1、解压后，作为maven项目导入，部署项目。**

**2、运行AmigoPlayDemo.java,模拟创建订单**

**3、测试notify通知**

# 类说明

## AmigoPlayDemo.java类说明

用于创建支付订单

|  |
| --- |
| **public** **class** AmigoPlayDemo {  **private** **static** **final** String *GIONEE\_PAY\_INIT* =  "https://pay.gionee.com/order/create";  // 跑通demo后替换成商户自己的api\_key  **private** **static** **final** String *API\_KEY* = "DDFDAEC3DBF544DD99EB9F508B429905";  // 跑通demo后，替换成商户自己的private\_key  **private** **static** **final** String *PRIVATE\_KEY* = "";  // 网游类型接入时固定值  **private** **static** **final** String *DELIVER\_TYPE* = "1";  // 成功响应状态码  **private** **static** **final** String *CREATE\_SUCCESS\_RESPONSE\_CODE* = "200010000";  **public** **static** **void** main(String[] args) {  Order order = **new** Order("201405280000001", "amigo\_player001", "testPaymobile",  *API\_KEY*, **new** BigDecimal("0.01"), **new** BigDecimal("0.01"),  **new** Timestamp(System.*currentTimeMillis*()), **null**, **null**);  String requestBody = **null**;  **try** {  requestBody = PayUtil.*wrapCreateOrder*(order, *PRIVATE\_KEY*, *DELIVER\_TYPE*);  } **catch** (InvalidKeyException | NoSuchAlgorithmException |  InvalidKeySpecException | SignatureException | IOException e) {  // **TODO** Auto-generated catch block  e.printStackTrace();  }  String response = **null**;  **try** {  response = HttpUtils.*post*(*GIONEE\_PAY\_INIT*, requestBody);  } **catch** (Exception e) {  // **TODO** : 处理异常;  **return**;  }  JSONObject json = (JSONObject) JSONValue.*parse*(response);  **if** (*CREATE\_SUCCESS\_RESPONSE\_CODE*.equals(json.get("status"))) {  String orderNo = (String) json.get("order\_no");  **if** (StringUtils.*isBlank*(orderNo)) {  // **TODO**: 如果返回orderNo为空，处理异常  **return**;  }  // **TODO** : 处理商户逻辑  **return**;  }  // **TODO** : 处理异常状态  **return**;  }  } |

## Order.java类说明

创建订单封装的对象说明

|  |
| --- |
| **public** **class** Order {  // 商户订单号  **private** **final** String outOrderNo;  // 玩家playerId  **private** **final** String playerId;  // 商品名称  **private** **final** String subject;  // 商户apiKey  **private** **final** String apiKey;  // 需支付金额，dealPrice值需要等于totalFee  **private** **final** BigDecimal totalFee;  // 商户总金额，dealPrice值需要等于totalFee  **private** **final** BigDecimal dealPrice;  // 订单提交时间，格式为yyyyMMddHHmmss，由商户服务器提供，客户端调起支付收银台时需要使用这个值  **private** **final** String submitTime;  // 订单失效时间，可选，格式为yyyyMMddHHmmss，如果有值必须参与签名  **private** **final** String expireTime;  // 服务器通知地址，可选，不能超过1024个字符，如果有该字段，必须参与签名  **private** **final** String notifyURL;  /\*\*  \* **@param** outOrderNo  \* 商户订单号 [必填]  \* **@param** playerId  \* 玩家playerId [必填]  \* **@param** subject  \* 商品名称 [必填]  \* **@param** apiKey  \* 商户apiKey [必填]  \* **@param** totalFee  \* 需支付金额，dealPrice值需要等于totalFee [必填]  \* **@param** dealPrice  \* 商户总金额，dealPrice值需要等于totalFee [必填]  \* **@param** submitTime  \* 订单提交时间，格式为yyyyMMddHHmmss，由商户服务器提供，客户端调起支付收银  台时需要使用这个值 [必填]  \* **@param** expireTime  \* 订单失效时间，可选，格式为yyyyMMddHHmmss，如果有值必须参与签名 [可选]  \* **@param** notifyURL  \* 服务器通知地址，不能超过1024个字符，如果有该字段，必须参与签名(如果商户需  要自定义参数，可以在创建订单时以  "http://www.partner.com/notifyReceiver?param1=value1&param2=value2  "的形式定义url) [可选]  \*/  **public** Order(String outOrderNo, String playerId, String subject, String apiKey, BigDecimal totalFee, BigDecimal dealPrice, Timestamp submitTime, Timestamp expireTime, String notifyURL) {  **this**.outOrderNo = outOrderNo;  **this**.playerId = playerId;  **this**.subject = subject;  **this**.apiKey = apiKey;  **this**.totalFee = totalFee;  **this**.dealPrice = dealPrice;  **this**.submitTime = SubmitTimeUtil.*toString*(submitTime);  **this**.expireTime = SubmitTimeUtil.*toString*(expireTime);  **this**.notifyURL = notifyURL;  }  …  } |

## PayUtil.java类说明

用于组装请求的json字符串

|  |
| --- |
| **public** **class** PayUtil {  **public** **static** String wrapCreateOrder(Order order, String privateKey, String deliverType)  **throws** InvalidKeyException, NoSuchAlgorithmException, InvalidKeySpecException, SignatureException,  IOException {  JSONObject jsonReq = **new** JSONObject();  String expireTime = order.getExpireTime();  String notifyURL = order.getNotifyURL();  StringBuilder signContent = **new** StringBuilder();  signContent.append(order.getApiKey());  jsonReq.put("api\_key", order.getApiKey());  signContent.append(order.getDealPrice());  jsonReq.put("deal\_price", order.getDealPrice().toString());  signContent.append(deliverType);  jsonReq.put("deliver\_type", deliverType);  **if** (!StringUtils.*isBlank*(expireTime)) {  signContent.append(expireTime);  jsonReq.put("expire\_time", expireTime);  }  **if** (!StringUtils.*isBlank*(notifyURL)) {  signContent.append(notifyURL);  jsonReq.put("notify\_url", notifyURL);  }  signContent.append(order.getOutOrderNo());  jsonReq.put("out\_order\_no", order.getOutOrderNo());  signContent.append(order.getSubject());  jsonReq.put("subject", order.getSubject());  signContent.append(order.getSubmitTime());  jsonReq.put("submit\_time", order.getSubmitTime());  signContent.append(order.getTotalFee());  jsonReq.put("total\_fee", order.getTotalFee().toString());  String sign = RSASignature.*sign*(signContent.toString(), privateKey, CharEncoding.*UTF\_8*);  jsonReq.put("sign", sign);  // player\_id不参与签名  jsonReq.put("player\_id", order.getPlayerId());  **return** jsonReq.toJSONString(JSONStyle.*NO\_COMPRESS*);  }  } |

## OrderNotify.java类说明

用于响应支付的结果通知

|  |
| --- |
| /\*\*\*\*\*\*\*\*\*\*\*\*\*\*设置请求以及响应的内容类型、编码方式\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  resp.setContentType("text/html;charset=UTF-8");  req.setCharacterEncoding("utf-8");  Map<String, String> map = **new** HashMap<String, String>();  String sign = req.getParameter("sign");  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*组装重排序参数\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  Enumeration<String> attributeNames = req.getParameterNames();  **while** (attributeNames.hasMoreElements()) {  String name = attributeNames.nextElement();  map.put(name, req.getParameter(name));  }  StringBuilder contentBuffer = **new** StringBuilder();  Object[] signParamArray = map.keySet().toArray();  Arrays.*sort*(signParamArray);  **for** (Object key : signParamArray) {  String value = map.get(key);  **if** (!"sign".equals(key) && !"msg".equals(key)) {// sign和msg不参与签名  contentBuffer.append(key + "=" + value + "&");  }  }  String content = StringUtils.*removeEnd*(contentBuffer.toString(), "&");  **if** (StringUtils.*isBlank*(sign)) {  *printResponse*(resp, "fail");  **return**;  }  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 签名验证 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  **boolean** isValid = **false**;  **try** {  isValid = RSASignature.*doCheck*(content, sign, *PUBLIC\_KEY*, CharEncoding.*UTF\_8*);  } **catch** (Exception e) {  *printResponse*(resp, "fail");  **return**;  }  **if** (isValid) {  *printResponse*(resp, "success");  **return**;  }  *printResponse*(resp, "fail"); |

## 工具类说明：

Order.java 订单对象封装类

PayUtil.java 创建订单数据工具类

HttpUtils.java http通信工具类

RSASignature.java 签名和验签工具类

SubmitTimeUtil.java 时间处理工具类

## 其它文件说明

pom.xml 标准pom文件，用于maven构建