# estore商城微信支付

## 第一步：修改页面

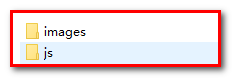
在订单详细页面生成二维码弹窗：



生成二维码弹窗我们需要借助js插件来完成，我们先引入js插件：

1 引入素材到项目





复制到WebContent下即可

2 引入素材到页面

|  |
| --- |
| <script type=*"text/javascript"* src=*"js/jquery.min.js"*></script>  <!-- 弹出层插件 -->  <script type=*"text/javascript"* src=*"js/jquery.wxdialog.js"*></script>  <script type=*"text/javascript"* src=*"js/qrcode.js"*></script>  <!-- 二维码中间的logo -->  <script type=*"text/javascript"* src=*"js/qrcode\_logo.js"*></script> |

3 调用显示弹窗的js函数——$.WXDialog(code\_url);

|  |
| --- |
| <script type=*"text/javascript"*>  $(**function**(){  $("#gopay").click(**function**(){  //显示弹窗的函数  $.WXDialog('weixin://wxpay/bizpayurl?pr=E88I6ZO');  });  });  </script> |

效果如最开始给大家展示的图一样。

4 发送支付请求给服务器：

|  |
| --- |
| <script type=*"text/javascript"*>  $(**function**(){  $("#gopay").click(**function**(){  //显示弹窗的函数  $.get("${root}/wxpay?oid=${order.id}",**function**(code\_url){  $.WXDialog(code\_url);  },"text");    });  });  </script> |

## 第二步：服务器接收请求，向微信服务器发送用户预支付请求，获取生成二维码所需的参数

|  |
| --- |
| **package** cn.itcast.web;  **import** java.io.IOException;  **import** javax.servlet.ServletException;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** cn.itcast.utils.PayUtils;  **public** **class** WXPayServlet **extends** HttpServlet {  **public** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  String oid = request.getParameter("oid");  String wxPay = PayUtils.*wxPay*(oid, "1");  response.getWriter().write(wxPay);  }  **public** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  doGet(request, response);  }  } |

## 第三步：生成二维码的弹窗，用户扫码即可

扫码支付即可。

## 第四步：主动查询用户的支付结果

为什么要主动查询用户的支付结果？

答可能出现以下情况：

1 当商户后台、网络、服务器等出现异常，商户系统最终未接收到支付通知；

2 调用支付接口后，返回系统错误或未知交易状态情况；

3 调用刷卡支付API，返回USERPAYING的状态；

4 调用关单或撤销接口API之前，需确认支付状态；

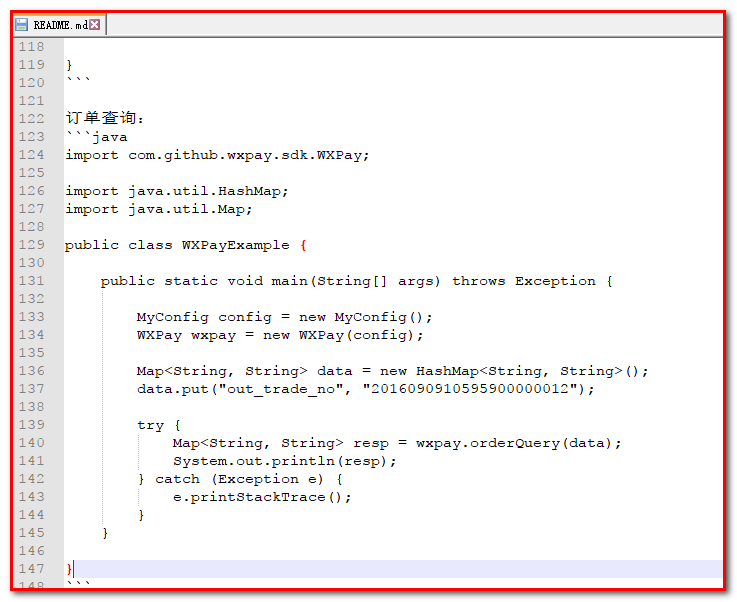
如何主动查询？

### 1 页面设置一个js函数，访问estore服务器：

|  |
| --- |
| //访问estore服务器，2秒一次  **var** query = setInterval(**function**(){  $.get("${root}/wxget?oid=${order.id}",**function**(data){  //返回为1的时候，支付已经完成，且订单状态已经修改  **if**(data == 1){  //清除定时器  clearInterval(query);  //跳转支付成功页面  location.href = 'pay\_success.jsp';    }  },"text");  },2000); |

### 2 estore服务器，如何查询微信支付服务器，订单支付情况？

在快速入门文档中有查询订单的方法：



### 3 复制其中的代码到Servlet：

|  |
| --- |
| **package** cn.itcast.web;  **import** java.io.IOException;  **import** java.util.HashMap;  **import** java.util.Map;  **import** javax.servlet.ServletException;  **import** javax.servlet.http.HttpServlet;  **import** javax.servlet.http.HttpServletRequest;  **import** javax.servlet.http.HttpServletResponse;  **import** com.github.wxpay.sdk.WXPay;  **import** cn.itcast.service.OrderService;  **import** cn.itcast.service.impl.OrderServiceImpl;  **import** cn.itcast.utils.PayConfig;  **public** **class** WXGetServlet **extends** HttpServlet {  **public** **void** doGet(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  String oid = request.getParameter("oid");  //向微信查询订单的支付情况  PayConfig config = **new** PayConfig();  WXPay wxpay = **new** WXPay(config);  Map<String, String> data = **new** HashMap<String, String>();  data.put("out\_trade\_no", oid);  **try** {  //获取订单支付结果  Map<String, String> resp = wxpay.orderQuery(data);  String trade\_state = resp.get("trade\_state");  System.***out***.println(resp);  **if**("SUCCESS".equals(trade\_state)) {  //支付成功修改订单的状态  OrderService orderService = **new** OrderServiceImpl();  orderService.update(oid,2);  //发生响应给浏览器  response.getWriter().write("1");  }**else** {  response.getWriter().write("-1");  }  } **catch** (Exception e) {  e.printStackTrace();  }  }  **public** **void** doPost(HttpServletRequest request, HttpServletResponse response) **throws** ServletException, IOException {  doGet(request, response);  }  } |

Service接口：

|  |
| --- |
| /\*\*  \* 根据订单，修改订单状态  \* **@param** oid  \* **@return**  \*/  **void** update(String oid,**int** status); |

Service实现类：

|  |
| --- |
| @Override  **public** **void** update(String oid, **int** status) {    orderDao.update(oid,status);  } |

Dao接口：

|  |
| --- |
| /\*\*  \* 修改订单状态的方法  \* **@param** oid  \* **@param** status  \* **@return**  \*/  **void** update(String oid, **int** status); |

Dao实现类：

|  |
| --- |
| @Override  **public** **void** update(String oid, **int** status) {  String sql = "update orders set status = ? where id = ?";  **try** {  qr.update(sql, status,oid);  } **catch** (SQLException e) {  e.printStackTrace();  }  } |

打印结果（未支付的时候）：

|  |
| --- |
| {nonce\_str=n8GbQTNzcbcm5bGS, out\_trade\_no=31aaa0f0181e4a5eb655f5514358a4b2, trade\_state=NOTPAY,  appid=wx8397f8696b538317, sign=743F12F15B46FA5E9E759F121FDAE9B0,  trade\_state\_desc=订单未支付,  return\_msg=OK,  result\_code=SUCCESS,  mch\_id=1473426802,  return\_code=SUCCESS} |

trade\_state=NOTPAY：这个返回值是支付结果。

可能出现的值有：

SUCCESS—支付成功

REFUND—转入退款

NOTPAY—未支付

CLOSED—已关闭

REVOKED—已撤销（刷卡支付）

USERPAYING--用户支付中

PAYERROR--支付失败(其他原因，如银行返回失败)

打印支付结果：（支付成功的时候）

|  |
| --- |
| {transaction\_id=4200000001201712028567274140, nonce\_str=ulqJgt40yAvW3UiG,  trade\_state=SUCCESS,  bank\_type=CFT,  openid=oNpSGwenvEi8kH28DPpT3o5KrA3M, sign=70C3A9959EDC749F53CA8E65260CC8FF,  return\_msg=OK,  fee\_type=CNY,  mch\_id=1473426802,  cash\_fee=1,  out\_trade\_no=ee4e681a1c3c46fb8fad22c33ce0d31e, appid=wx8397f8696b538317,  total\_fee=1,  trade\_type=NATIVE,  result\_code=SUCCESS,  attach=,  time\_end=20171202173147,  is\_subscribe=N,  return\_code=SUCCESS} |

trade\_state=SUCCESS,支付成功！！！！

页面效果：



数据订单效果：（订单状态为2，已支付）

