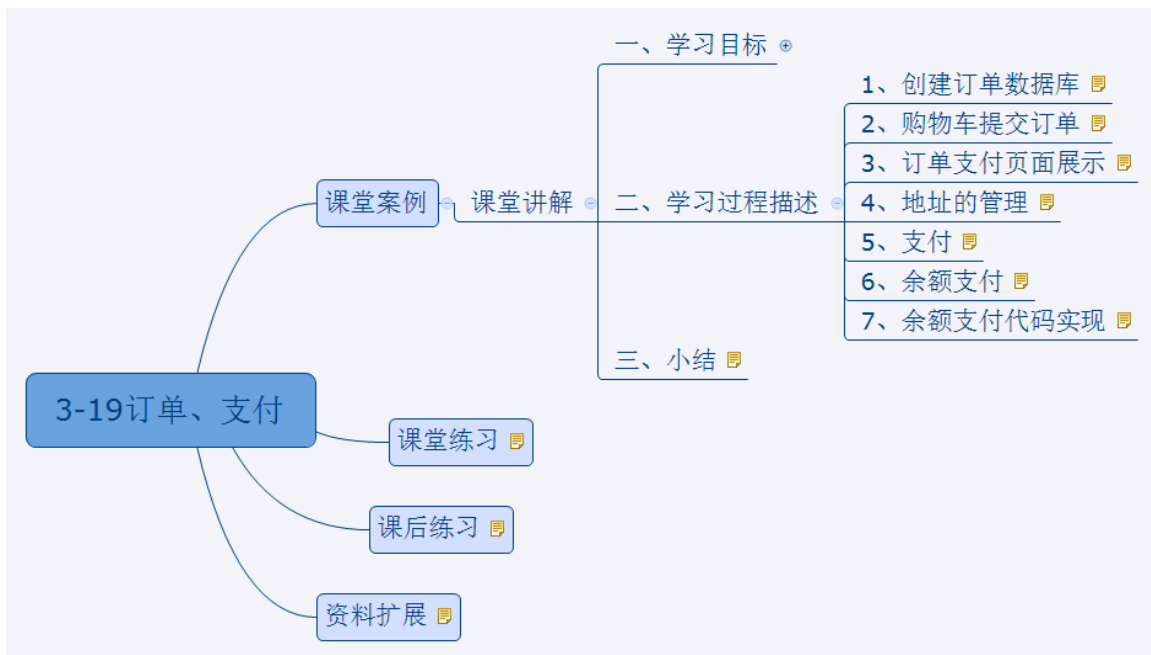


3-19订单、支付

3-19订单、支付	1
1. 课堂案例	2
课堂讲解	2
一、学习目标	2
订单数据库的创建	2
订单付款的实现	2
二、学习过程描述	2
1、创建订单数据库	2
2、购物车提交订单	3
3、订单支付页面展示	5
4、地址的管理	6
5、支付	6
6、余额支付	7
7、余额支付代码实现	8
三、小结	9
2. 课堂练习	9
3. 课后练习	9
4. 资料扩展	9



1. 课堂案例

课堂讲解

一、学习目标

订单数据库的创建

订单付款的实现

二、学习过程描述

1、创建订单数据库

当购物车提交后或者商品详情页点击购买时，此时进入的是购物付款的流程阶段，类似于独孤九剑的后面几剑，此时要考虑和处理的一些事情是，把购买人信息和商品信息结合存入一张订单表。

这个订单表里要有：

用户的信息，包括在这个网站的会员名、如果是实物商品那么得有收件人、联系方式、联系地址等等
商品信息，包括商品id, 价格, 数量

状态：等待付款、已付款、已发货、已收货、完成、申请退货、同意退货、同意退款、退款完成等逻辑

订单创建时间、订单完成时间等信息

参考如下：

rl_orders @goshopping (localhost) - 表 *

文件 编辑 窗口 帮助

新建 保存 另存为 添加栏位 插入栏位 删除栏位 主键 上移 下移

栏位 索引 外键 触发器 选项 注释 SQL 预览

名	类型	长度	小数点	允许空值 (
id	int	20	0	<input type="checkbox"/>	1
order_number	varchar	20	0	<input type="checkbox"/>	
userid	int	20	0	<input type="checkbox"/>	
goodsid	int	20	0	<input type="checkbox"/>	
num	int	20	0	<input type="checkbox"/>	
price	float	10	2	<input type="checkbox"/>	
linkman	varchar	20	0	<input type="checkbox"/>	
mobile	varchar	16	0	<input type="checkbox"/>	
address	varchar	255	0	<input type="checkbox"/>	
status	tinyint	1	0	<input type="checkbox"/>	
ctime	int	11	0	<input type="checkbox"/>	
mtime	int	11	0	<input type="checkbox"/>	

默认:

注释: 0 : 未付款, 1 : 已付款, 2 : 已发货, 3 : 已收货, 4 : 已退货

☐ 自动递增

☒ 无符号

☐ 填充零

注意: 订单标明可以用orders, 但是order是关键词, 千万不要拿它当表名。如果用oder做表名, 让你从此开始怀疑人生, 生不如死, 不想活。。。。。我要自由。。。开玩笑的, 就是告诉你, 不能用那个做表名

2、购物车提交订单

下面代码是购物车提交订单的代码参考, 商品详情点购买直接去付款的逻辑就是少了删除购物车数据这个环节, 其他处理类似

提示一点: 一般一个订单都会生成一个订单号来处理付款这个环节, 为什么呢?

哎, 问的好, 因为如果在用银行卡付款、支付宝付款、微信支付, 作为付款的唯一标识, 你的这个标识不能跟自己的订单号重复也不能跟别人的订单号重复。所以就会生成一个随机复杂的订单号, 可看看淘宝的订单号, 如此长, 长, 长.....嘿嘿

```

<script type="text/javascript">
    $('next').click(function(){
        // alert(12345)
        // 获取勾选的数据
        var data = [];
        $('.cateid:checked').each(function(){
            var select = [];
            select.push($(this).parents('tr').find('goods_id').val());
            select.push($(this).parents('tr').find('txt_num').val());
            select.push($(this).parents('tr').find('goods_price').val());
            data.push(select);
        });
        // 代码调试
        // console.log(data);
        // alert(data.join(','));
        var msg = data.join('|');
        if(data='') {
            layer.alert('请勾选需要付款项');
        } else {
            $.post(":U('change')",{data:msg},function(data){
                if(data==0) {
                    layer.alert('提交失败');
                } else {
                    location.href=":U('check',array('order_number'=>"+data+"))";
                }
            });
        }
    })
}
</script>

```

```

// 生成订单操作
public function change()
{
    if(IS_AJAX){
        if(!empty(session('user_id'))){
            $map = array();
            $order_number = date('Ymd').mt_rand(100,999);
            // $data = json_decode(I('data'));
            $data = explode('|', I('data'));
            $demo = array();
            foreach($data as $v){
                $temp = explode(',', $v);
                $demo[] = $temp;
            }
            // file_put_contents('cao.txt', $data[1]);
            $data = array();
            foreach($demo as $val){
                $map['order_number'] = $order_number;
                $map['userid'] = session('user_id');
                $map['goodsid'] = $val[0];
                // 收集goodsid
                $data[] = $val[0];
                $map['num'] = $val[1];
                $map['price'] = $val[2];
                $map['ctime'] = time();
                $result += M('orders')->add($map);
            }
            // 添加菜单成功，则删除购物车的内容
            if($result){
                $where['goodsid'] = array('in', $data);
                M('cart')->where($where)->delete();
            }
            $this->ajaxReturn($result);
        } else {
            $this->ajaxReturn(0);
        }
    }
}
}

```

3、订单支付页面展示

从购物车过来，或从个人详情页过来，那么数据已经关联了个人账户和商品信息，在此页面就是关联个人的收货信息，无误下一步，此功能就是核实收件信息

如果没有地址，就添加新收件地址来，然后选中----->下一步

[+添加新地址](#)

商品信息				
订单号	商品名称	数量	小计	操作
20170612331	 欧莱雅洗发水	3	¥ 75	2017-06-12

支付宝

余额付款

商品数量总计: 3件
 商品金额总计 (不含运费): **¥ 75**

[上一步](#)

[下一步](#)

4、地址的管理

地址的管理

就是一张地址信息表关联了此网站唯一用户表里的用户id

订单支付页面有罗列好几个该用户的地址, 就是根据这用户的id来到这个地址信息表查出所有该用户的地址来

那么在这个数据表里呢, 可以做一些限制:

- 1、比如一个用户最多只能存几个地址(淘宝也是这么做的)
- 2、会弄一个字段来控制默认是那个地址(做法就是某条数据该字段为1, 其他为0, 那么默认地址就出来了)

5、支付

比较常见的支付方式, 支付宝支付, 微信支付, 余额支付, 聚成的第三方平台(支付宝和微信支付其实也是第三方支付)

1、支付宝支付:就是调用支付宝支付api, 其官网提供的demo可以在本地localhost环境下测试, 开发环境也是在其沙漏环境下, 只有等项目上线通过阿里审核才能正式开始支付宝支付。(如果在微信开发想应用支付宝支付, 只能在客户端调用支付宝app支付, 想在项目这端调支付, 目前是不可能的事情, 也许以后可以)

2、微信支付:首先要申请公众号, 必须是服务号(个人、订阅号暂时无法申请微信支付接口), 服务号申请下来后在公众号后端可以继续申请微信支付接口(微信支付属于财付通类下)。其demo会在扩展资料里存放。(特别注意, 微信支付成功后会有不同时段来八次回调, 所以在处理付款后的业务逻辑中特别要注意, 处理完毕以后要设置一个字段来, 处理好了就不必再处理了, 否则会给你反反复复处理)

3、余额支付:一般是网站平台的虚拟货币,比如你把钱往平台的账户里充了100块,平台在你的账户下设置了一个字段,存储了等价的虚拟币。

可以拿来购买平台的商品或者申请提现,对方会把虚拟币转换成等价的人民币。

付款的概念就是对方账户,比如对方的微信账号收到钱了,改变订单的状态字段,说你付款了,给你发货。

注意:银行支付也是调用接口支付,但是一般的小公司申请不下来,所以比较少见些。有的也是应用第三方聚成的平台,间接的使用银行接口付款。

名	类型	长度	小数点	允许空值	
id	int	20	0	<input type="checkbox"/>	1
order_number	varchar	20	0	<input type="checkbox"/>	
userid	int	20	0	<input checked="" type="checkbox"/>	
goodsid	int	20	0	<input checked="" type="checkbox"/>	
num	int	20	0	<input checked="" type="checkbox"/>	
price	float	10	2	<input checked="" type="checkbox"/>	
linkman	varchar	20	0	<input checked="" type="checkbox"/>	
mobile	varchar	16	0	<input checked="" type="checkbox"/>	
address	varchar	255	0	<input checked="" type="checkbox"/>	
status	tinyint	1	0	<input checked="" type="checkbox"/>	
ctime	int	11	0	<input checked="" type="checkbox"/>	
mtime	int	11	0	<input checked="" type="checkbox"/>	

默认: 0

注释: 0: 未付款, 1: 已付款, 2: 已发货, 3: 已收货, 4: 已退款, 5: 待退款, 6: 待退货, 7: 退款成功

☐ 自动递增

☒ 无符号

☐ 填充零

6、余额支付

余额支付的业务逻辑是:

判断你购买的东西需要多少余额币,

如果不够, 提示你不够

如果需要支付的余额币比你账户里的少, 说明可以用余额币支付。

扣除你账户里的余额币, 扣除成功, 那么就把你的订单状态给改变, 提示你购买成功, 等待商家发货

注: 所有的支付程序都是一样的, 只是从哪里扣款不同。所以有些支付暂时还没有申请下来, 那么可以弄一个可以实现的把流程走通了

老司机告诉你: 可以把付款后处理的事情封装成一个类或者方法, 成功了就去调用, 减少很多很多工作。如此, 妈妈再也不担心我的学习啦!!!

7、余额支付代码实现

下面是余额支付代码实现, 其他支付就是付款手法不同, 后期要处理的事情一毛一样

代码参考如下:

```
<script type="text/javascript">
    $('.next').click(function() {
        // alert(123)
        // 获取金额
        var count = $(this).parents('.right').find('.count i').html();
        // 获取订单号
        var order_number = $('.order_number').val();
        $.post('pay', {count: count, order_number: order_number}, function(data) {
            if(data==0) {
                layer.alert('登陆失效', {end: function(data) {
                    location.href=":{U('User/login')}";
                }});
            } else if(data==1) {
                layer.alert('金额不够, 请充值');
            } else {
                location.href=":{U('Cart/message')}";
            }
        });
    });
</script>
```



```
// 付款
public function pay()
{
    if(IS_AJAX) {
        if(!empty(session('user_id'))) {
            $money = I('count');
            $order_number = I('order_number');
            // 判断此用户账户里有没有那么多钱
            $cash = M('user')->where(array('id'=>session('user_id'))->getField('cash'));
            if($money > $cash) {
                // 付款不够分支
                $this->ajaxReturn(1);
            } else {
                // 付款成功分支
                M('user')->where(array('id'=>session('user_id'))->setDec('cash', $money);
                // 改变订单状态为已付款
                $info = M('orders')->where(array('order_number'=>$reorder_number))->select();
                foreach($info as $val) {
                    M('orders')->where(array('id'=>$val['id']))->setInc('status', 1);
                }
                // 把存在session的订单号给删除了
                session('order_number', null);
                $this->ajaxReturn(2);
            }
        } else {
            // 用户登陆失效分支
            $this->ajaxReturn(0);
        }
    }
}
```

三、小结

理解付款的应用场景

知道付款的几种方式

测试demo和余额支付代码的实现

2. 课堂练习

创建订单数据库表

购物车提交订单制作

支付页面的制作

3. 课后练习

用余额支付, 走通整个购物流程

4. 资料扩展

支付宝支付demo和微信支付demo见资料扩展文件夹里