



电商项目面试

电商总概.....	2
电商项目的重要性.....	2
电商项目里的技术特点.....	2
系统功能.....	2
电商项目的思维图.....	3
项目团队构成.....	3
项目开发流程.....	4
涉及技术.....	4
开发环境及工具.....	4
电商面试常见问题.....	5
1. 说说你最近做的这个项目的背景，简单的介绍一下你这个项目？.....	5
2. 项目的整体结构（从框架和具体技术来说）.....	5
3. 分布式架构，对比传统架构的优势。.....	5
4. 本项目提供了哪些模块（功能和服务）.....	5
5. 你负责的哪些模块.....	6
6. 简述下这些模块怎么实现的？（描述举例只给出部分功能的过程）.....	6
商品类型管理.....	6
属性管理.....	6
商品栏目管理.....	7
商品管理.....	7
权限管理.....	8
邮件发送验证管理.....	9
购物车管理.....	11
订单管理.....	12
网上支付管理.....	13
7. 对于大流量的网站,您采用什么样的方法来解决访问量问题?.....	14
8. 数据库的存储引擎? 有什么区别?.....	14
9. Sql 语句的优化?.....	14
10. 什么是页面静态化?.....	15
11. 如何实现用户的安全登录?.....	15
12. 使用过 Memcache 缓存吗? 请简述一下.....	15
13. 说出 JQUERY 常用的几个函数?.....	15
14. 什么是二次开发? 二次开发的弊端?.....	15
15. 单例模式.....	16
16. Redis 有几种持久化的方式? redis 缺点是什么?.....	16
17. 前后台数据不一致怎么办?.....	16
18. 数据库的读写分离? 主从复制?.....	16
19. 高并发如何处理?.....	16
20. 前后台数据交互时用到那些技术?.....	16
21. 什么是 Ajax, Ajax 异步处理?.....	17
22. 什么是 Bootstrap? 有什么优势?.....	17
23. http 协议请求的状态? 404 是什么? 500?.....	17

框架介绍:	17
一、 PHP 中的框架是什么?	17
二、 PHP 中常见的框架	18
TP 框架 (ThinkPHP) :	18
Laravel 框架	21

电商总概

电商项目的重要性

找工作面试的过程中，项目将会是整个流程的核心灵魂，也是能在面试中能让面试官尽快认同你的一个强有力的依据，所以在面试中把电商项目清晰的表述出来是极为关键的。

电商项目里的技术特点

- ①**技术更新较快**: 根据市场的需求，不断迭代更新
- ②**技术涉及面广**: 除了 PHP，还会用到 Python, GO 等其他的一些语言；数据库中 MySQL, nosql 是最频繁使用的（当然也有的公司会用 oracle，但是 PHP 一般都是以 MySQL 为主），服务器端使用 Linux（少部分公司会用到 Unix），还经常涉及到服务器安全、系统安全等安全方面的技术
- ③**分布式**: 从前的单一的机器上运行，现在是分散到不同机器上，最后将数据集中汇总。集中式向分布式进行发展是由需求来推动
- ④**高并发、集群（高可用集群）、负载均衡**: 由并发问题采用集群进行处理，其中，集群会涉及服务器的主从以及分布问题，使用负载均衡。（权重高低）高可用是对用户而言，用户的服务不中断（系统升级，服务不中断，公司电商系统的部分更新等）。
- ⑤**海量数据**: 每年商家的各类活动（双 11，双 12 等等）订单量、浏览数、商品量、活动相关数据都将会超级大超级多（一般随同高并发出现）
- ⑥**业务复杂**: 电商业务并不简单：并不是商品展示出来后，简单的加入购物车后购买就完成了。除此以外后台业务逻辑是相当复杂，比如优惠（包邮、满减），秒杀，抢购等
- ⑦**系统安全**: 系统上线必须通过系统安全部门审核通过，安全性问题正逐步的被放到台面上，而且很多企业对这块相当重视

系统功能

本商城系统是一个综合性的 B2C 平台，类似京东商城、天猫商城。

用户:

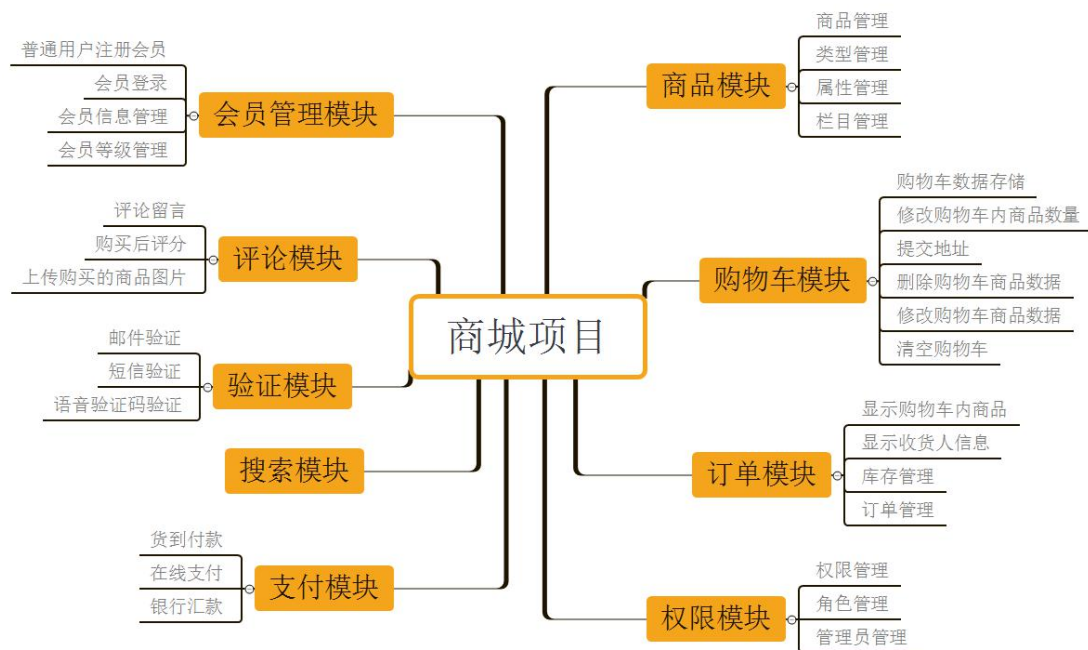
1. 用户可以通过注册成为商城的会员，登录商城
2. 根据会员等级获得相应的优惠和特权
3. 用户可以浏览商品，可以根据商品类型进行搜索
4. 用户可以对购物车进行操作（增删改查商品的数量、商品详情等）
5. 用户下订单、退订单，对订单的结算等

商家:



1. 商家可以入驻商城，在商城里开店售卖自己的商品
2. 商家也有商家等级，如果在商城好评和商品销售量到达一定数量，可获得商家的一些特权
3. 商家可以管理自己店铺的商品：上架下架商品；设定库存量；设定自己店铺的会员制度等

电商项目的思维图



以上是大部分常用模块，如有其它模块自行补充。根据上图的体系架构，一层层的进行归纳记忆，从项目——模块——功能一点点的往深入熟悉记忆，并且加以理解

项目团队构成

- 产品经理：1 人，确定需求以及给出产品原型图。
- 项目经理：1 人，项目管理。
- 前端团队：1 人，根据产品经理给出的原型制作静态页面。
- 后端团队：3 人，实现产品功能。
- 测试团队：2 人，测试所有的功能。

（上述是以团队为单位，有的公司技术部不止一个团队，还有运维、UI、交互设计师等等，视情况而定）

项目开发流程



涉及技术

框架：ThinkPHP CI YII Laravel 等

前端：HTML CSS JS/JQ(Bootstrap、node.js、angular.js 等可以了解下) AJAX

PHP：PHP+MySQL session 和 cookie Nosql(redis memcache mongodb) 接口技术 缓存技术 优化技术等

服务器方面：Linux Nginx

开发环境及工具

WAMP 或者是 LAMP 比较常见，除此以外还有 LNMP 等

常用的工具：PHPStorm (PHP 最正统的工具) NetBeans Editplus

Webstorm (web 最正统的工具) Sublime Dreamweaver

SVN GIT

（简历中或者是面试中尽量不要提及版本号，如果要写则需要把版本号对齐，否则不要写版本号）

电商面试常见问题

1.说说你最近做的这个项目的背景，简单的介绍一下你这个项目？

我们公司之前主要以实体店为主，进行批发与零售，业务也相对比较传统。为了适应市场需求，增强公司竞争力，提升业务绩效，另一方面，也为基于互联网的商务模式创新奠定基础。所以开始 xxx 商城建设项目，其中包含商品管理、订单管理、类目管理、客户管理、合作商管理、客服管理、购物平台、内容管理等，很大程度上分担了人工的压力，对提高客户服务效率和客户满意度能够起到较好的作用。（先大体的描述下项目，然后能够挑一两个自己最为熟悉的模块进行叙述）

2.项目的整体结构（从框架和具体技术来说）

框架方面：从大处进行描述，比如本项目是属于电商项目，采用的是 TP 框架，运用 php+MySQL+Apache 进行开发。因为 TP 框架是开源的、轻量级的，所以用起来更加方便快捷，从效率和成本方面来讲都是性价比比较高的，所以那时候团队决定用这个框架进行开发。然后再从框架的一些优点特性进行说明下，让面试官觉得你对这个框架很了解

技术：描述技术时首先用从自己比较熟悉的模块入手（比如说注册验证时会用到的 JS 技术、正则匹配、ajax 无刷新验证等等），然后由此慢慢的向其他方面的技术进行拓展，在说具体技术的时候如果能说到目前比较新或者是有难度的技术，能够让面试官眼前一亮，这样能为自己的面试增加不少分。

3.分布式架构，对比传统架构的优势。

①最为明显的一点，在传统的架构中，如果某个功能需要进行维护，那么我们必须停掉整个服务，这对于公司的运营会造成损失。分布式系统在核心功能模块使用单独服务器，维护部分模块不影响用户的其他操作。

②在海量数据处理方面，传统架构显得比较乏力；分布式系统架构采用服务器集群，使用负载均衡，海量数据处理游刃有余！

③在性能(检索)以及维护方面，分布式系统架构也有较为明显的优势。

4.本项目提供了哪些模块（功能和服务）

- ◆ 商品模块：其中包括商品管理，类型管理，属性管理，栏目管理等等
- ◆ 订单模块：其中包括下单，退单，库存，收货人信息等
- ◆ 会员模块：会员注册，会员信息管理，会员等级管理，会员权限等
- ◆ 购物车模块：购物车数据存储，增删改查购物车商品，清空购物车等
- ◆ 提交订单页面：提交用户的订单信息，处理并发问题。
- ◆ 个人中心：包括用户的登录，个人信息的管理，收货地址的管理，用户所下的订单的管理
- ◆ 支付模块：支付方式管理（在线支付、货到付款）等

5.你负责的哪些模块

在项目中主要负责相关功能的开发，主要有：

- 1) 后台管理系统：主要实现商品管理、商品规格参数管理、订单管理、会员管理等、CMS(内容管理系统)等，并且提供了跨域支持；
 - 2) 前台系统：主要是面向用户访问，使用 js、ajax 进行前后台数据交互（一般是用 json 格式数据返回）
 - 3) 会员登录：提供和用户信息相关的接口，比如说用户注册、查询等接口（登录时需要进行多重验证，特别注意安全方面）
 - 4) 订单功能：主要是提供和订单相关的业务接口，在订单系统了做了严格的数据校验以及高并发写的支持（这里可以说使用队列实现），并且使用了定时器实现对下单的时间控制，比如说关闭超时未付款的订单；
 - 5) 搜索功能：主要是提供商品的搜索，可以采用 Sphinx 全文搜索，当然也有其他的搜索方式；
 - 6) 会员系统：主要是维护用户的信息，已购买订单、优惠券、系统消息、修改密码、绑定手机等功能；
 - 7) 缓存：主要是用 Redis 实现，并且对 Redis 做了集群来保证 Redis 服务的高可用（缓存方面除了 redis 外还有 memcached）
 - 8) 支付系统：主要是负责订单的支付、对账等功能，主要是对接了支付宝的接口；
- （根据个人的实际情况选择最为熟悉的模块，进行叙述）

6.简述下这些模块怎么实现的？（描述举例只给出部分功能的过程）

商品类型管理

包含功能：添加商品类型、显示商品类型列表、修改商品类别、删除商品类型

描述举例：

- 1、添加商品类型：在后台模块新建一个商品类型的控制器(TypeController)，并添加 add 方法，同时建立好对应的静态页。新建一个对应的商品类型表，在表里添加数据验证。在商品类型控制器中的 add 方法中完成商品类型的添加。（其中关键点是 create 验证和防止非法字段提交）
- 2、商品类型列表：在商品类型控制器中添加一个 list 方法，取出商品类型数据，并在对应的静态页中完成数据遍历，就能在页面中得出相应的商品类型列表

属性管理

包含功能：添加属性、显示属性列表、根据商品类型搜索属性、修改属性、删除属性

描述举例：

前提条件：建立一个属性表将各字段全部预先设置好，才可以进行一下操作

- 1、添加属性：在后台模块新建一个 AttributeController 的控制器，并添加 add 方法，并建立好对应的静态页面，修改页面里的表单。然后新建一个 attribute 表对应的模型文件，进行数据验证（验证规则可以查手册），在方法中完成属性添加的代码。在属性控制器中写入 list 方法，并且完成对应静态页，把取出的数据遍历到对应的静态页
- 2、根据商品类型搜索属性：（给 select 添加 change 事件，完成提交表单，表单提交到当前页面。）首先在方法中将商品显示出来，并将商品数据完成遍历，给显示商品 类



型的 select 添加 change 事件，完成表单提交，其中会引用 JQ 添加 change 事件，当触发 change 事件后，显示出当前显示的商品类型。在商品类型列表，添加“属性列表”的连接，在添加完成属性后，跳转到属性列表，直接显示所属当前商品类型的属性，最后实现分页

商品栏目管理

包含功能：创建栏目、栏目的增删改查（注：商品类型和商品栏目的区别，类型是大类，栏目是小类）

描述举例：

前提条件：要创建一个栏目表，以便进行下面的功能开发

1、添加栏目：在 admin 模块里面新建一个 category 的控制器，添加 add 方法，并完成对应页面设计。在页面上设计好表单，在添加栏目时将已经添加的栏目显示出来（新建一个商品栏目模型，在里面定义好方法，用于取出栏目的数据，并且已经格式化好的），然后在 add 方法里将数据取出来，并且进行遍历，在此方法中完成栏目的添加

2、栏目列表：在 category 的控制器里添加 lst 方法，并且完成对应页面的设计，在页面中遍历栏目数据，将其遍历显示到页面中

3、删除栏目：（注：在删除栏目时，如果有子栏目则不能删除）在栏目列表中添加删除栏目的链接（用 U 函数自动生成路径），在相应的 category 控制器里，写上 del 方法，接受传递过来的 id，实例化自定义模型，利用 sql 拼接进行查询，找出对应的栏目记录，将其删除

商品管理

包含功能：商品的增删改查，在线编辑器，属性选项卡，属性的增删改查，商品相册选项卡，添加相册，文件上传，相册上传等

描述举例：

前提条件：要将商品表创建好，具体字段根据需求开设计

1、添加商品：在后台模块（admin 模块）里新建一个 Goods 控制器，写入 add 方法，同时完成相应的页面设计，在 add 方法对应的页面中对表单进行相应的修改（具体的页面要求感觉需求来修改）。在 add 方法中完成商品的添加功能代码，例如生成唯一的商品 id 号、添加商品的时间等（利用前缀+时间+唯一码=唯一 id）

2、图片上传：可以利用 TP 已经定义好的钩子函数，来完成图片文件的上传，返回保存图片的路径。钩子函数在模型里面会自动调用，比如：入库之前：_before_insert() 和入库之后：_after_insert() 会被模型里面的 add 方法，自动调用。只需要在上传之前，把图片的路径、上传单个文件的最大值、允许上传格式限制好

3、缩略图：在图片上传的基础上利用 new \Think\Image() 按照原生图片的比例生成一个缩略像素*缩略像素的缩略图，每个缩略图都会生成一个唯一的 id 名，并且保存到预设好的路径文件夹中（这里要注意的是生成多张缩略图的时候，必须先生成大图，再生成小图），图片生成后将路径入库。

4、显示属性选项卡：（选项卡的显示重点是给标签设置一个 onclick 事件函数）要在选项卡中显示商品类型，在 goods 控制器里面的 add 方法中取出商品类型数据，然后在相应的页面商品属性选项卡位置遍历商品类型数据，分别显示出来

5、完成商品属性的添加：前提将商品属性表设计好，然后根据实际的静态页面进行表单的修改，修改指定表单的域名（即标签中的 name），修改完成后在 Goods 模型里定义一个钩子 _after_insert，在该方法中完成属性数据的入库，由此属性的添加完成

权限管理

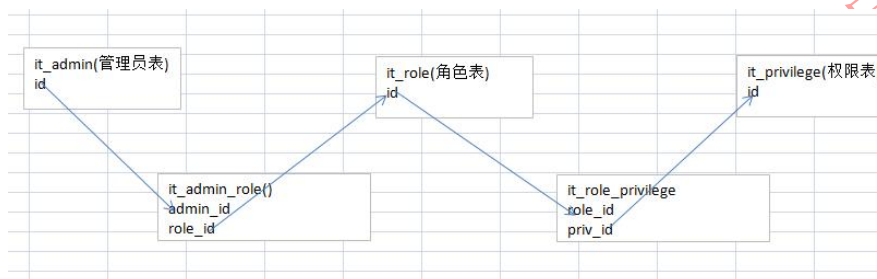
包含功能：管理员登录，权限管理（权限管理控制），角色管理，管理员管理

权限管理简述：权限管理是根据不同等级，不同管理员的情况下，授予不同的权限，各自操作各自授权。授权方式大体分为两种：一种是直接给管理员授权权限，适合于管理员比较少，权限数量也比较少的情况下；第二种是 RBAC 基于角色的访问控制。

在建表方面，采用五张表

其中有三张主表，权限表、角色表、管理员表，两张从表，角色与权限的中间表、管理员与角色的中间表。

角色表与权限表的关系：一个角色可以有多个权限，一个权限可以属于多个角色，权限表与角色表是多对多的关系，要体现这种关系，要使用角色与权限的中间表；管理员表与角色表的关系：一个管理员可以属于多个角色，一个角色可以包含多个管理员，角色表与管理员表是多对多的关系，要体现这种关系，要使用管理员与角色的中间表。五张表关系图如下：



描述举例：

1、权限管理：在项目中权限管理就是按钮，通过按钮给对应的管理员或者是角色分配权限。在后台模块新建一个权限的控制器，并写入 `add` 方法，完成对应的页面设计，修改好页面上的表单，完善页面。新建一个权限模型，并且在模型中添加数据验证，设置好验证规则（具体代码参考项目中的），在模型中完成权限添加。在父级权限完成后，继续添加子级权限：在模型里面定义一个方法，用于取出权限，并格式化好权限数据，然后在添加权限控制器里面，取出权限数据，并进行遍历，将遍历出来的子级权限显示在页面中

2、角色管理：角色管理中包含对角色的增删改查，授权等等。添加角色首先要在在后台模块新建一个 `RoleController` 的控制器，写入添加 `add` 方法，完成对应的静态页面，取出权限数据，并修改对应的表单。在后台模块里面，添加一个 `RoleModel` 模型，写入添加数据验证。接下来就是数据入库，在入库时要注意的是该表单提交的数据，是入库两张的表，一个是角色表，一个是角色与权限的中间表，其中角色与权限的中间表的数据，是使用钩子函数 `_after_insert()` 来完成。

3、角色删除：如果角色里面有管理员，则不能删除，在删除角色的同时，要删除角色与权限的中间表里面的对应的数据。首先在角色列表中添加删除的链接（用 `U` 函数自动生成路径），然后在角色控制器中添加删除 `del` 方法，用于删除角色，其次在角色模型中添加钩子函数 `_after_delete()` 用来删除角色与权限中间表对应的角色数据，删除完成

4、管理员管理：在添加管理员时，要分配管理员所属角色，并且将管理员各个数据根据要求设定好格式。创建管理员控制器，添加 `add` 方法，取出角色数据，并完成页面设计将数据显示到页面上。在管理员模型中添加数据验证，使用静态方法来完成验证，



```
class AdminModel extends Model{
    //定于静态的数据验证
    protected $_validate=array(
        array('admin_name','require','管理员名称不能为空'),
        //验证管理员是否唯一
        array('admin_name','','管理员名称已经存在',1,'unique'),
        //验证密码的长度要在6到12位之间
        array('password','6,12','密码长度要在6到12位之间',1,'length'),
        //验证确认密码
        array('rpassword','password','两次密码输入的不一致',1,'confirm')
    );
}
```

使用模型添加数据，接受明文密码，然后用双重 md5 进行加密，这个过程的代码可以直接写到钩子函数里。使用钩子函数添加管理员与角色中间表的数据，在管理员的模型里面，添加 `_after_insert()` 钩子函数

```
protected function _after_insert($data,$options){
    $admin_id = $data['id'];//获取管理员的id
    //获取角色的id
    $role_id = $_POST['role_id']+0;
    //向中间表插入数据it_admin_role
    M('AdminRole')->add(array(
        'admin_id'=>$admin_id,
        'role_id'=>$role_id
    ));
}
```

5、管理员列表：在管理员的控制器里面添加 `lst` 方法，用于取出管理员数据，并完成对应的静态页面，要注意：显示管理员的同时，要显示出所属角色的名称，因此要连表查询（管理员表 管理员角色表 角色表）

```
public function lst(){
    //取出管理员数据
    $adminmodel = D('Admin');
    $admindata = $adminmodel->field("a.admin_name,a.id,c.role_name")->join("a left join it_admin_role b on a.id=b.admin_id left join it_role c on c.id=b.role_id")->select();
    $this->assign('admindata',$admindata);
    $this->display();
}
```

然后在对应的界面中进行管理员数据遍历并且显示到页面上。

6、删除管理员：要操作的表是： `it_admin` 和 `it_admin_role`，在删除 `it_admin_role` 表里面数据时，则可以使用钩子函数 `_after_delete()`，钩子函数是 TP 框架里面提供的，主要有：`_before_insert()`、`_after_insert()`、`_before_update()`、`_after_update()`，`_before_delete()`，`_after_delete()`（该系列函数是在模型里面定义，会被自动调用。）

注意：超级管理员不能被删除（拥有最高权限）

邮件发送验证管理

包含功能：（原理要理解、）注册、登录、发送、密码找回等

描述举例：

1、原理：



邮件的 SMTP (Simple Mail Transfer Protocol)：即简单邮件传输协议，它是一组用于由源地址到目的地址传送邮件的规则，由它来控制信件的中转方式。SMTP 协议属于 TCP/IP 协议族，它帮助每台计算机在发送或中转信件时找到下一个目的地。通过 SMTP 协议所指定的服务器，就可以把 E-mail 寄到收信人的服务器上了，整个过程只要几分钟。SMTP 服务器则是遵循 SMTP 协议的发送邮件服务器，用来发送或中转发出的电子邮件。

2、用户注册：创建用户表，将所需字段设置好（在设置字段的时候可以预留一些字段以便后面拓展使用）。在前台模块中新建一个 user 控制器，添加 register 方法，并且完成对应静态页。在 user 模型中添加数据验证，返回数据

```
class UserModel extends Model{
    protected $_validate=array(
        array('username','require','用户名不能为空'),
        array('username','','用户名已经存在',1,'unique'),
        array('username','checkname','用户名含有非法的字符',1,'callback'),
        array('password','6,12','密码长度要在6到12位之间',1,'length'),
        array('rpassword','password','两次密码输入不一致',1,'confirm'),
        array('email','email','邮箱格式不合法'),
    );
    protected function checkname($username){
        if(strpos($username, '@')!==false || strpos($username, '#')!==false){
            return false;
        }else{
            return true;
        }
    }
}
```

在控制器中完成注册后，发送邮件：

```
if($usermodel->add()){
    //注册完成
    //发送邮件
    $email = I('post.email');
    $username = I('post.username');
    $title='xxx网站，用户注册激活';
    $url='http://www.shshop.com'.U('User/active',array('key'=>$validate,'email'=>$email));
    $content="尊敬的({$username})用户，你已经注册成功，单击链接激活<br/><a href='{$url}'>单击激活</a>";
    $fromuser='中国武林盟主网';
    if(sendEmail($title,$content,$email,$fromuser)){
        $this->success('注册成功，赶紧去邮箱激活',U('login'));
    }else{
        $this->error('注册失败，请重新注册');
    }
}else{
    //注册失败
    $this->error('注册失败');
}
}
```

邮件发送后，由用户进行激活认证，因此在控制器中新建 active 方法。方法中首先要接受数据（key 值，用户邮件数据），然后判断链接是否合法，根据传递的 email 查找出 validate

字段内容和 key 值进行匹配。如果匹配成功那么，就让用户进行激活，激活的同时将 active 的初始状态更改为 1，则功能完结

3、密码找回功能：实现方式，根据用户名->查找出注册时设置的问题->输入问题的答案->如果答案正确->发送一封邮件，根据该邮件里面的链接去完成密码的修改。首先在页面上将密码找回的链接设置好，在 user 控制器里添加 findpwd1 方法（用于找回密码的第一个方法），该方法显示出找回密码输入用户的一个界面，同时应该完成对应页面的编写。再在 user 控制器中添加 findpwd2 方法，根据用户名查找出注册时设置的问题，让用户输入问题的答案。第三是在 user 控制器中写入 findpwd3 方法，用于问题答案的匹配并发送邮件，通过邮件的链接来完成密码的修改。最后在 user 控制器中添加一个 update 方法，用于完成密码的修改

购物车管理

包含功能：提交商品到购物车、显示购物车列表、删除购物车里商品、修改购物车、清空购物车等等

1、购物车的实现方式：①可以把购物车的数据，给存储到 session 里面，一旦关闭浏览器，则商品数据就丢失；②可以把购物车的数据，给存储到 cookie 里面，可以长久的保存购物车里面的数据；③可以把购物车的数据，给存储到数据库里面，可以长久的保存购物车里面的数据；④可以把购物车的数据，给存储到缓存里（memcache,redis），也可以长久的保存购物车的数据。

注：Session 可能会引起并发问题，如果脚本很短，这通常没有问题。但如果脚本运行时间比较长，那就可能会产生问题。在现代 Web 应用程序开发中，有一个非常常见的情况，就是使用 AJAX 技术在同一个页面内发送多个请求获取数据。如果这些请求都需要使用 Session，那么第一个请求到达服务器后会取得 Session 锁，其它请求就必须等待，所有请求将串行处理，即使它们彼此之间并没有依赖关系。这将大大增加页面的响应时间。有一个方法可以避免这个问题，就是在使用完 Session 以后立即调用 session_write_close（）方法关闭 Session。这样 Session 锁就会释放，即使当前脚本还在等在处理。需要注意的是，调用该方法后，当前脚本就不能进一步操作 Session 了

2、实例：京东购物车的设置，如果用户已经登录，则把购物车数据保存到数据库里面，如果没有登录，则把购物车的数据给保存到 cookie 里面，当登录时，要判断 cookie 是否有购物车的数据，如果有，则把 cookie 里面的数据给移动到数据库里面。

3、提交商品到购物车：在前台模块里新建一个 CartController 控制器，添加 addCart 方法



```
public function addCart(){
    // p($._POST);
    //接收表单提交的数据
    $goods_id = I('post.goods_id');
    $goods_count = I('post.goods_count');
    //接收提交的属性信息，
    $goods_attr_id = '';
    $attr = I('post.attr');//接收的是一维数组
    if($attr){
        //有属性，则组建一个用逗号隔开的字符串
        $goods_attr_id = implode(',',$attr);
    }
    $cartmodel = D('Cart');
    $cartmodel->addCart($goods_id,$goods_attr_id,$goods_count);
    $this->success('加入购物车成功',U('cartList'));//成功加入后，添加到购物车列表。
}
```

4、购物车列表：在购物车模型里添加 cartList 方法，用于显示列表。如果用户已经登录，则直接从购物车数据库里面取出数据，如果用户没有登录，则直接从 cookie 里面获取数据。最后在购物车控制器里面的 cartList 的方法中，并取出购物车数据，并在对应的静态页面里完成遍历，显示出来

5、cookie 移到 DB 中：用户登录后，要判断 cookie 里面是否有购物车数据，如果有，则就移动到数据库，在购物车模型里面定义一个方法 cookie2db

```
//把cookie里面的数据移动到数据库
public function cookie2db(){
    //思路：要从cookie里面取出数据，判断是否为空，如果不为空，则就不移动
    $cart = isset($_COOKIE['cart'])?unserialize($_COOKIE['cart']):array();//返回的是一维数组，
    //判断数据是否为空
    if(empty($cart)){
        //开始移动 把数据插入到数据库，并清空cookie就是移动。
        //一维数组的格式就类似与array('1-3,6'=>10,'2-4,5'=>20)
        $user_id = $_SESSION['user_id'];
        foreach($cart as $k=>$v){
            $a = explode('-', $k);
            $goods_id = $a[0];
            $goods_attr_id = $a[1];
            $goods_count = $v;
            //入库之前，要判断，该商品是否已经在数据库里面，如果在，则修购买商品数量，如果不在，则是添加
            $info = $this->where("goods_id=$goods_id and goods_attr_id='$goods_attr_id' and user_id=$user_id")->find();
            if($info){
                //有该商品，则修改购买商品数量
                $this->where("goods_id=$goods_id and goods_attr_id='$goods_attr_id' and user_id=$user_id")->setInc('goods_count',$goods_count);
            }else{
                //添加商品数据
                $this->add(array(
                    'goods_id'=>$goods_id,
                    'goods_attr_id'=>$goods_attr_id,
                    'goods_count'=>$goods_count,
                    'user_id'=>$user_id
                ));
            }
        }
    }
}
```

登录成功后调用 cookie2db 方法即可

订单管理

描述举例：

1、建立订单控制器，添加 flow 方法，完成对应的静态页面。在 order 控制器里面添加一个 done 方法，用来完成下订单功能：首先接受传递过来的订单信息数据，判断购物车里面是否有商品，防止出现能够直接访问该方法的情况，如果没有商品则提示无法下单，接着判断用户是否已经登陆，如果没有登录把当前访问的地址存储到 session 里，跳转登录页面，完成登录后再跳转回来，再判断是否填写收货人的信息，同理，如果没有填写则跳转填写页面。下单时会自动算出订单总额，生成唯一的订单号，并且从 session 中取出用户 id，获取收货人信息。随后将订单信息和订单商品信息入库结算完成后，清空购物车，订单完成

2、下订单注意事项：



①在下订单之前要判断库存是否充足：判断商品属性 id 是否有值，如果有说明该商品有属性，直接从 product 表里进行判断，如果没有属性则直接从 goods 表里判断库存

②下完订单后要减掉库存：在入库 it_order_goods 表时，要完成减掉库存

③高并发下订单问题：使用文件锁，对文件锁定后，不是操作文件，是锁定文件后，执行下订单，查看库存之前，就要加锁，一直到购买完成

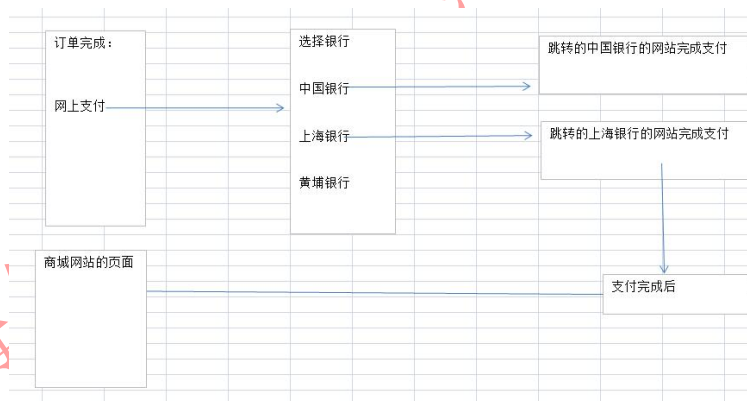
```
<?php
$fp = fopen("/tmp/lock.txt", "w+");
if (flock($fp, LOCK_EX)) { // 进行排它型锁定
    fwrite($fp, "Write something here\n");
    flock($fp, LOCK_UN); // 释放锁定
} else {
    echo "Couldn't lock the file !";
}
fclose($fp);
```

④添加事务：防止某些操作只完成一半

网上支付管理

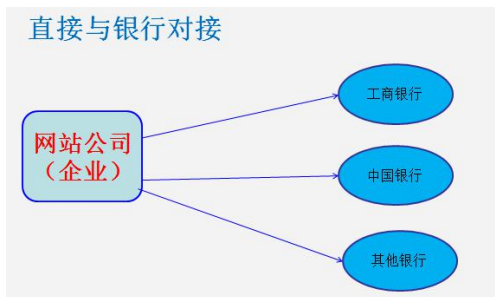
支付过程主要是调用第三方支付接口

网上支付流程：



商城与银行对接方案：

1、直接与银行对接



优点：

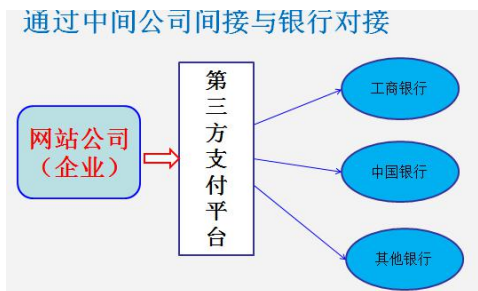
因为直接与银行进行财务结算，交易资金结算比较安全。适合资金流量比较大的

企业，这种方案适合于，每月结算金额百万以上的企业。

缺点：

开发工作量比较大，而且银行会不定期升级交易系统，随着银行系统的升级，企业也需要做相应的改动，所以维护工作量也是比较大的，而且企业每年还需要向银行交纳一定数量的接口使用费用。

2、通过中间公司间接与银行对接



优点：

开发工作量较少，银行升级系统，不需要企业做相应的修改，除非中间企业的接入规范发生了改变，相对前一种接入方案，这种方案的维护工作量比较少的。因为只与一家企业对接，所以接入费用相对比较低。这种方案适合于：每月结算资金在几十万以下的中小企业。

缺点：

因为是与中间企业进行资金结算，目前所有中间企业都是私企，资金安全是个大问题

7.对于大流量的网站,您采用什么样的方法来解决访问量问题?

- ①确认服务器硬件是否足够支持当前的流量
- ②优化数据库访问
- ③禁止外部的盗链
- ④控制大文件的下载
- ⑤使用不同主机分流主要流量、集群
- ⑥使用流量分析统计软件

8.数据库的存储引擎？有什么区别？

MyISAM、InnoDB

构成上，MyISAM 的表在磁盘中有三个文件组成，分别是表定义文件（.frm）、数据文件（.MYD）、索引文件（.MYI），而 InnoDB 的表由表定义文件(.frm)、表空间数据和日志文件组成。

安全方面，MyISAM 强调的是性能，其查询效率较高，但不支持事务和外键等安全性方面的功能，而 InnoDB 支持事务和外键等高级功能，查询效率稍低。

对锁的支持，MyISAM 支持表锁，而 InnoDB 支持行锁。

9.Sql 语句的优化？

- 1)尽量选择较小的列
- 2)将 where 中用的比较频繁的字段建立索引
- 3)select 子句中避免使用 ‘*’
- 4)避免在索引列上使用计算、not in 和<>等操作

5)当只需要一行数据的时候使用 limit 1

6)保证单表数据不超过 200W，适时分割表。针对查询较慢的语句，可以使用 explain 来分析该语句具体的执行情况。

10.什么是页面静态化？

就是把一个动态的页面（操作数据库的 php 页面）变成一个静态页面，后续用户直接访问静态页面。

页面静态化技术分为两种：真静态和伪静态。

真静态：把一个动态的页面，实实在在的转成一个静态的页面，即.html 文件

伪静态：所谓伪静态是从 url 地址上看是一个静态页面，但是实际上还是对应一个动态页面

11.如何实现用户的安全登录？

①防止 Sql 注入，对特殊字符进行转义、过滤或者使用预编译的 sql 语句绑定变量。
最小权限原则，特别是不要用 root 账户，为不同的类型的动作或者组建使用不同的账户。

②当 sql 运行出错时，不要把数据库返回的错误信息全部显示给用户，以防止泄露服务器和数据库相关信息

③XSS 攻击

④暴力破解

12.使用过 Memcache 缓存吗？请简述一下

Memcache 是把所有的数据保存在内存中，采用 hash 表的方式，把每条数据有 key 和 value 组成，每个 key 独一无二的，当要访问的某个值的时候先按照找到值，然后在返回结果，Memcache 采用 LRU 算法来逐渐把过期的数据清除掉

13.说出 JQUERY 常用的几个函数？

\$(selector).hide() 隐藏被选元素

\$(selector).show() 显示被选元素

\$(selector).toggle() 切换（在隐藏与显示之间）被选元素

\$(selector).slideDown() 向下滑动（显示）被选元素

\$(selector).slideUp() 向上滑动（隐藏）被选元素

\$(selector).slideToggle() 对被选元素切换向上滑动和向下

\$(selector).fadeIn() 淡入被选元素 \$(selector).fadeOut() 淡出被选元素

\$(selector).fadeTo() 把被选元素淡出为给定的不透明度

\$(selector).animate() 对被选元素执行自定义动画

14.什么是二次开发？二次开发的弊端？

二次开发，简单的说就是在现有的软件上进行定制修改，功能的扩展，然后达到自己想要的功能，一般来说都不会改变原有系统的内核。dede 二次开发就是以 dede 织梦系统为基础进行的二次开发。

弊端：1) 插件限制太多；

2) 修改源文件对升级有影响。



15.单例模式

```
class MySQLDB {
    //私有属性用来保存
    private static $instance;
    //私有的构造函数阻止在类的外部实例化
    private function __construct() {

    }
    //私有的__clone阻止克隆对象
    private function __clone() {

    }
    //获得实例的方法
    public static function getInstance() {
        if(!self::$instance instanceof self) //当前实例不属于MySQLDB类就实例化
            self::$instance=new self;
        return self::$instance; //返回当前对象
    }
}
```

16.Redis 有几种持久化的方式？ redis 缺点是什么？

两种，分别是 RDB 和 AOF 快照；

Redis 的主要缺点是数据库容量受到物理内存的限制，不能用作海量数据的高性能读写，因此 Redis 适合的场景主要局限在较小数据量的高性能操作和运算上

17.前后台数据不一致怎么办？

前后操作人员录入数据保持一致，开启验证功能，由于通讯故障造成的前台数据积压，维护时及时备份

18.数据库的读写分离？主从复制？

读写分离，基本的原理是让主数据库处理事务性增、改、删操作（INSERT、UPDATE、DELETE），而从数据库处理 SELECT 查询操作。数据库复制被用来把事务性操作导致的变更同步到集群中的从数据库。

至少两台数据库服务器，可以分别设置主服务器和从服务器，对主服务器的任何操作都会同步到从服务器上

原理：mysql 中有一种日志，叫做 bin 日志（二进制日志），会记录下所有修改过数据库的 sql 语句。主从复制的原理实际是多台服务器都开启 bin 日志，然后主服务器会把执行过的 sql 语句记录到 bin 日志中，之后把这个 bin 日志发给从服务器，在从服务器再把 bin 日志中记录的 sql 语句同样的执行一遍。这样从服务器上的数据就和主服务器相同了。

19.高并发如何处理？

需要对服务器的架构分层，重新布局，负载均衡，集群策略。

负载均衡器（硬件和软件）

硬件：F5-Bigip:立竿见影，价格昂贵，网游公司或大网站用的比较多

软件：lvs(linux virtual server 虚拟服务，集成到内核中)，nginx(可以做 web 服务器，也可以做负载均衡使用)

负载均衡策略：

- （1）轮询技术：把客户端的请求轮流分发给服务器。
- （2）最少连接：负载均衡把请求给最空闲的服务器
- （3）ip 哈希：同一地址的客户端，始终请求同一台服务器。

20.前后台数据交互时用到那些技术？



Ajax, Ajax 的跨域请求常用的有两种方式:

1) 使用中间层过渡的方式:

中间过渡,很明显,就是在 AJAX 与不同域的服务器进行通讯的中间加一层过渡,这一层过渡可以是 PHP、JSP、c++等任何具备网络通讯功能的语言,由中间层向不同域的服务器进行读取数据的操作。拿 PHP 做一个例子,如果需要对不同域的某一个 php 进行通讯,现在客户端的 xmlhttprequest 先 query 本域的一个 PHP,然后由本域的这个 PHP 去和不同

域的 PHP 进行通讯,然后由本域的 PHP 输出 response;

2) 使用标签

这个方法是利用标签中的 src 来 query 一个 PHP 获得 response,因为标签的 src 属性不存在跨域的问题。

21.什么是 Ajax, Ajax 异步处理?

Ajax 是一种在页面没有刷新的情况下,通过客户端(浏览器)与服务器交互的一种技术。

Ajax 语言的载体是 javascript,最大特点:页面不刷新完成请求。

open 方法:如 xhr.open('get','demo.php',true) open 方法,第三个参数表示是异步请求,默认 true 表示异步请求处理

22.什么是 Bootstrap? 有什么优势?

简单、直观、强悍的前端开发框架,让 Web 开发更快速、简单

优势:入门简单,支持响应式,优先支持移动设备

23.http 协议请求的状态? 404 是什么? 500?

200 (成功): 服务器已成功处理了请求。通常,这表示服务器提供了请求的网页

201 (已创建): 请求成功并且服务器创建了新的资源

202 (已接受): 服务器已接受请求,但尚未处理

203 (非授权信息): 服务器已成功处理了请求,但返回的信息可能来自另一来源

204 (无内容): 服务器成功处理了请求,但没有返回任何内容

205 (重置内容): 服务器成功处理了请求,但没有返回任何内容

206 (部分内容): 服务器成功处理了部分 GET 请求

404 (未找到): 服务器找不到请求的网页

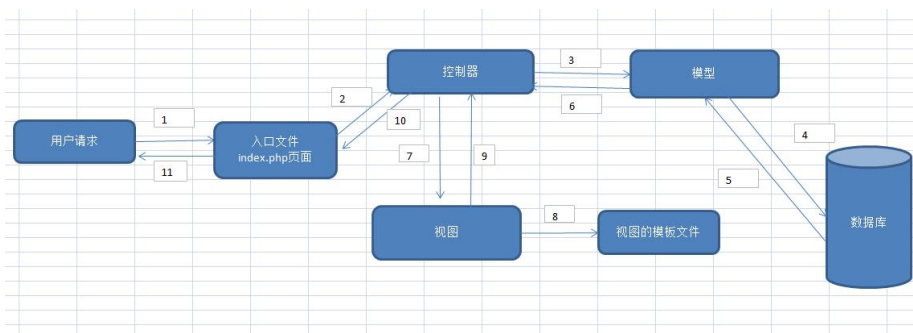
500 (服务器内部错误): 服务器遇到错误,无法完成请求

框架介绍:

一、PHP 中的框架是什么?

框架就是通过提供一个开发 Web 程序的基本架构,PHP 开发框架把 PHPWeb 程序开发摆到了流水线上。换句话说,PHP 开发框架有助于促进快速软件开发(RAD),这节约了开发时间,有助于创建更为稳定的程序,并减少开发者的重复编写代码的劳动。

框架专门用于为这些常用元素(数据库交互、表示层、应用程序逻辑)提供结构,以便可以花费更少的时间来编写数据库接口代码或者表示层接口,而花费更多的时间来编写应用程序本身。以这种方式分解应用程序,这种架构被称为模型-视图-控制器(Model-View-Controller, MVC)。模型(Model)指数据,视图(View)指表示层,而控制器(Controller)指应用程序逻辑或业务逻辑。



MVC 示意图

二、PHP 中常见的框架

TP 框架 (ThinkPHP) :

ThinkPHP 框架是 PHP 最常见的框架之一，也是目前市面上的主流框架。ThinkPHP 是一个免费开源的，快速、简单的面向对象的 轻量级 PHP 开发框架，是为了敏捷 WEB 应用开发和简化企业应用开发而诞生的。ThinkPHP 从诞生以来一直秉承简洁 实用的设计原则，在保持出色的性能和至简的代码的同时，也注重易用性。并且拥有众多的原创功能和特性，在易用性、扩展性和性能方面不断优化和改进，已经成长为国内最领先和最具影响力的 WEB 应用开发框架，众多的典型案例确保可以稳定用于商业以及门户级的开发。

① ThinkPHP 基于 MVC 的 PHP 框架

M - Model	模型	工作：负责数据的操作
V - View	视图 (模板)	工作：负责前台页面显示
C - Controller	控制器 (模块)	工作：描述功能

② ThinkPHP 核心文件介绍

- ├─ThinkPHP.php 框架入口文件
- ├─Common 框架公共文件
- ├─Conf 框架配置文件
- ├─Extend 框架扩展目录
- ├─Lang 核心语言包目录
- ├─Lib 核心类库目录
 - ├─Behavior 核心行为类库
 - ├─Core 核心基类库
 - ├─Driver 内置驱动
 - ├─Cache 内置缓存驱动
 - ├─Db 内置数据库驱动
 - ├─TagLib 内置标签驱动
 - └─Template 内置模板引擎驱动
 - └─Template 内置模板引擎
- └─Tpl 系统模板目录

③ url 的 4 种访问方式

1.PATHINFO 模式

http://域名/项目名/入口文件/模块名/方法名/键 1/值 1/键 2/值 2

2. 普通模式

http://域名/项目名/入口文件?m=模块名&a=方法名&键 1=值 1&键 2=值 2

3. REWRITE 模式

http://域名/项目名/模块名/方法名/键 1/值 1/键 2/值 2

4. 兼容模式

http://域名/项目名/入口文件? s=模块名/方法名/键 1/值 1/键 2/值 2

④ TP 中的几种输出方式

ThinkPHP 中的输出:

a、通过 echo 等 PHP 原生的输出方式在页面中输出

b、通过 display 方法输出

想分配变量可以使用 assign 方法

c、修改左右定界符

若要修改配置文件中的配置项

'TMPL_L_DELIM' => '{ ' , //修改左定界符

'TMPL_R_DELIM' => ' }>' , //修改右定界符

⑤ 输出模板内容

a、display 方法

1. display 中没有参数

```
$this->display();
```

2. 可以带参数

```
$this->display(本模块文件夹下的其他模板文件);
```

```
$this->display('index2');
```

```
$this->display(其他文件夹下的模板文件);
```

```
$this->display('Public:error');//注意, 仅仅需要在 Tpl 下有 Public 文件夹以及其中的 error.html 即可, 不需要一定有 Public 模块
```

b、fetch 方法

获得模板文件中的内容, 以字符串形式返回

```
$content=$this->fetch('Public:error');
```

c、show 方法

不需要模板文件, 可以直接输出模板内容

```
$content=$this->fetch('Public:error');
```

```
dump($content);
```

```
$content=str_replace('h1','i',$content);
```

```
$this->show($content);
```

⑥ 模型中连贯操作

语法: `$model->方法 1()->方法 2()->方法 n()->select();`

⑦ 自动验证

主要是验证表单提交的数据的合法性, 验证表单数据的合法性, 有两种方式, 一种是通过前端 js 代码来验证, 一种是数据入库之前的 php 代码验证。在实际开发中, 两种方式都要有, 前端 js 验证。前端 js 代码用户是可以禁用的, 因此要进行 php 代码验证, 达到双重验证。

字段验证的方式是集成到模型里面的 create 方法里面, 注意在使用 create 方法之前需要将配置文件中的 extension=php_mbstring.dll 打开



实用项

(1) 代码流程追踪：在配置文件里面 ‘SHOW_PAGE_TRACE’ =TRUE

(2) TP 中的事务：

\$User->startTrans()：开启事务处理

\$User->commit()：如成功，则提交

\$User->rollback()：如失败，则回滚

(3) 模型里面的 trueTableName 属性：

建立一个模型，在默认操作时，是操作的前缀+模型名称的表，可以通过 trueTableName 属性设置变成操作其他的表

(4) TP 中常用的字母函数：

S 函数：缓存管理方法，3.1.2 版本后因 cache 方法并入原 S 方法，所以 cache 方法不再建议使用，用 S 方法即可；

U 函数：U 方法用于完成对 URL 地址的组装，特点在于可以自动根据当前的 URL 模式和设置生成对应的 URL 地址：U(‘地址’, ‘参数’, ‘伪静态’, ‘是否跳转’, ‘显示域名’);

D 函数：D 方法是用的比较多的，用于实例化自定义模型类，是 ThinkPHP 框架对 Model 类实例化的一种封装，并实现了单例模式，支持跨项目和分组调用，调用格式如下：D(‘项目:’/[分组/]模型’, ‘模型层名称’), 方法的返回值是实例化的模型对象。

M 函数：M 方法用于实例化一个基础模型类。M 方法的调用格式：M(‘[基础模型名:]模型名’, ‘数据表前缀’, ‘数据库连接信息’)

注意：M 和 D 的区别：

1、M 不需要自定义模型类，减少 IO 加载，性能较好；

2、实例化后只能调用基础模型类（默认是 Model 类）中的方法；

3、可以在实例化的时候指定表前缀、数据库和数据库的连接信息；

A 函数：A 方法用于在内部实例化控制器，调用格式：A(‘[项目:’/[分组/]模块’, ‘控制器层名称’)

R 函数：R 方法用于调用某个控制器的操作方法，是 A 方法的进一步增强和补充，R 方法的调用格式：R(‘[项目:’/[分组/]模块/操作’, ‘参数’, ‘控制器层名称’)

C 函数：C 方法是 ThinkPHP 用于设置、获取，以及保存配置参数的方法，使用频率较高

N 函数：N 方法属于计数器方法，被用于核心的查询、缓存统计的计数和统计。但是其实可以用于应用的其他计数用途，用法比较简单，调用格式：N(‘计数位置’[, ‘步进值’])

L 函数：L 方法用于启用多语言的情况下，设置和获取当前的语言定义，调用格式：L(‘语言变量’[, ‘语言值’])

session 函数：Session 方法用于 Session 设置、获取、删除和管理操作

cookie 函数：cookie 函数也是一个多元化操作函数，完成 cookie 的设置、获取和删除操作。

TP 特性总结

1. 多表查询非常方便，在 model 里几句代码就可以完成完成多表的关联操作。
2. 融合了 smarty 框架技术，使前台和后台独立分开
3. 支持多种缓存技术，memcache 技术支持非常好
4. 命名规范，模型，视图和控制器严格遵循命名规则，通过命名一一对应
5. 类库扩展、驱动扩展、应用扩展、模型扩展、控制器扩展、标签库扩展、模板引擎扩展、Widget 扩展、行为扩展和模式扩展
6. 支持多种 URL 模式
7. 自动完成表单数据的验证过滤，生成安全的数据对象
8. 内置 AJAX 数据返回方法，支持 JSON、XML 和 EVAL 格式返回客户端
9. 内建丰富的查询机制，包括组合查询、复合查询、区间查询、统计查询、定位查询、动态查询和原生查询，让你的数据查询简洁高效
10. 系统支持多数据库连接和动态切换机制，支持分布式数据库

Laravel 框架

Laravel 是一套简洁、优雅的 PHP Web 开发框架。它可以让你从面条一样杂乱的代码中解脱出来；它可以帮你构建一个完美的网络 APP，而且每行代码都可以简洁、富于表达力。

① 路由

HTTP 请求类型路由：语法：Route::请求类型（URL 信息，处理逻辑的地址或者功能代码）

控制器路由（隐式路由）：Route::controller（URL 信息，控制器名称）

RESTful 资源控制器路由：语法：Route::resource（URL 信息，控制器名称）

② 自动加载文件

Laravel 的自动加载，其实也就是 Composer 的自动加载

Composer 根据声明的依赖关系，从相关库的源下载代码文件，并根据依赖关系在 Composer 目录下生成供类自动加载的 PHP 脚本，使用的时候，项目开始处引入“/vendor/autoload.php”文件，就可以直接实例化这些第三方类库中的类了。

Laravel 自动加载的过程就是这样实现的，总结为四种加载方式：

1. PSR0 加载方式一对应的文件就是 autoload_namespaces.php
2. PSR4 加载方式一对应的文件就是 autoload_psr4.php
3. 其他加载类的方式一对应的文件就是 autoload_classmap.php
4. 加载公用方法一对应的文件就是 autoload_files.php

③ 服务容器——Laravel 真正的核心

服务容器，也叫 IOC 容器，其实包含了依赖注入(DI)和控制反转(IOC)两部分，是 laravel 的真正核心。其他的各种功能模块比如 Route（路由）、Eloquent ORM（数据库 ORM 组件）、Request and Response（请求和响应）等等等等，实际上都是与核心无关的类模块提供的，这些类从注册到实例化，最终被开发者所使用，其实都是 laravel 的服务容器负责的。

Laravel 特性总结

1. 强大的 rest router：用简单的回调函数就可以调用，快速绑定 controller 和 router
2. artisan：命令行工具，很多手动的工作都自动化
3. 可继承的模版，简化 view 的开发和管理
4. blade 模版：据说渲染速度更快
5. ORM
6. migration：管理数据库和版本控制
7. 测试功能也很强大
8. composer 也是亮点，现在新更新的项目都支持

框架优缺点对比总结：

TP 框架：

优点

动态模型：可以实现不需要定义 Model

充分利用了 PHP5 的特性

部署简单只需一个入口文件，一切搞定，简单高效

中文文档齐全，入门超级简单

自带模板引擎，具有独特数据验证和自动填充功能，框架更新速度比较迅速

缺点

操作方法的命名就是当前的 ACTION_NAME，这也造成了 ACTION_NAME 的命名可能会和 PHP 关键字冲突的情况

耦合太深，很多组件没法替换

Laravel 框架

优点

Laravel 的设计思想是很先进的，非常适合应用各种开发模式 TDD，DDD 和 BDD，作为一个框架，它准备好了一切，composer 是个 php 的未来，没有 composer，PHP 肯定要走向没落。

laravel 最大的特点和优秀之处就是集合了 php 比较新的特性，以及各种各样的设计模式，Ioc 容器，依赖注入等。

缺点

基于组件式的框架，所以比较臃肿

命名空间，相比较太复杂

上海传智播客 · 黑马程序员