Fecshop 介绍

源文档：<http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_product_image.html>

Fecshop，是由Terry从2015年初开始，一直坚持到今天的开源电商项目，非商业化运作，是真正 开源的电商系统， 遵循BSD-3-Clause开源协议，您可以免费的应用于您的线上商城。如有任何其他疑问， 可以联系邮箱：2358269014@qq.com

Fecshop 全称为Fancy ECommerce Shop，是基于php Yii2框架之上开发的一款优秀的开源电商系统， 遵循BSD-3-Clause开源协议，Fecshop支持多语言，多货币，架构上支持pc，手机web，手机app，和erp对接等入口，您可以免费快速的定制和部署属于您的电商系统。 截止到2017-12-15，appfront（pc前端web），appadmin（后台），apphtml5（wap端web）， appserver（VUE，手机app的api提供端），console（命令行）， appapi（和第三方系统通信端）都已经完成，现在已经趋于稳定， Terry在2010年开始进入跨境电商领域， 用了不少开源电商系统，譬如magento， 发现开源框架都有一定 的诟病，在并发方面差，后期扩展，业务发展后期重构难， 尤其是现在的移动端的发展，多入口的电商模式占据主流， 性能方面的要求越来越高，Fecshop基于Yii2的高效框架，在此基础上进一步封装，加入了 service层和block层，数据库采用了nosql和mysql结合的方式， 关系型表放到mysql中，譬如优惠券，购物车，订单等， 非关系型数据表（非关系型代表不会出现多表强事务类型操作） 放到mongodb中， 缓存用redis，搜索目前用的是mongodb的FullTextSearch功能， 支持一些主流语言的分词与搜索，对于中文搜索，使用的是xunsearch， 后期会扩展ElasticSearch来进行搜索（ElasticSearch有中文插件，安装后支持中文分词）。 总之，Fecshop目前的定位是为了让程序员们有一个方便学习，扩展，开发的电商框架系统， 如果您发现有哪些代码结构可以优化，调整，或者您有更加合理的建议，可以发送到邮箱： 2358269014@qq.com。

**Fecshop线上案例**：<http://www.fecshop.com/topic/55>

**关于Fecshop**：<http://www.fecshop.com/first>

**Fecshop的各种问题解答QA**：<http://www.fecshop.com/topic/400>

**Fecshop 【Appfront】PC Web Demo**：<http://fecshop.appfront.fancyecommerce.com/>

**Fecshop 【Apphtml5】Mobile Web Demo（WAP）**：[http://fecshop.apphtml5.fancyecommerce.com](http://fecshop.apphtml5.fancyecommerce.com/)

**Fecshop 【Appserver】Mobile VUE Demo(Appserver,前后端彻底分离模式)**：<http://demo.fancyecommerce.com/#/>

**Fecshop 开源协议**：<http://www.fecshop.com/license>

**Fecshop 使用说明文档地址：** <http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-README.html>

**Fecshop 二次开发文档地址：** <http://www.fecshop.com/doc/fecshop-guide/develop/cn-1.0/guide-README.html>

**FecShop Github地址：**<https://github.com/fancyecommerce/yii2_fecshop>

**FecShop oschina地址：** <https://www.oschina.net/p/fecshop>

**Terry个人博客：**<http://www.fancyecommerce.com/>

**TerryCSDN博客：**<http://blog.csdn.net/terry_water>

# Fecshop 配置结构

1. [配置文件以及优先级](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_config_construction.html)
2. [优先级部分配置细化 - yii\helpers\ArrayHelper::merge()部分](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_config_construction.html#yiihelpersarrayhelpermerge)
3. [配置加速](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_config_construction.html)

fecshop 配置结构是层级覆盖的方式，通过yii2的 yii\helpers\ArrayHelper::merge()函数 将所有的配置进行合并，最终成为一个大数组。配置结构通过层级覆盖的方式可以很方便的对fecshop的功能进行重写

## 配置文件以及优先级

#### 1.fecshop合并的配置文件：

fecshop配置是根据优先级，优先级高的配置可以覆盖优先级低的配置， 优先级由低到高的排列如下（其实就是index.php中的代码）：

$config = yii\helpers\ArrayHelper::merge(

require(\_\_DIR\_\_ . '/../../common/config/main.php'),

require(\_\_DIR\_\_ . '/../../common/config/main-local.php'),

require(\_\_DIR\_\_ . '/../config/main.php'),

require(\_\_DIR\_\_ . '/../config/main-local.php'),

# fecshop services config

require(\_\_DIR\_\_ . '/../../vendor/fancyecommerce/fecshop/config/fecshop.php'),

# fecshop module config

require(\_\_DIR\_\_ . '/../../vendor/fancyecommerce/fecshop/app/appfront/config/appfront.php'),

# thrid part confing

# common modules and services.

require(\_\_DIR\_\_ . '/../../common/config/fecshop\_local.php'),

# appadmin local modules and services.

require(\_\_DIR\_\_ . '/../config/fecshop\_local.php')

);

1到4，是Yii2框架的配置文件结构：

1.@common/config/main.php # Yii2 公用配置，一般是Yii2的一些组件的公用配置。

2.@common/config/main-local.php # Yii2 本地公用配置，一般是Yii2的一些数据库组件的配置

3.@app/config/main.php # Yii2 本app内的配置

4.@app/config/main-local.php # Yii2 本app内的本地配置

5和6部分，是fecshop的配置文件结构：

5.@vendor/fancyecommerce/fecshop/config/fecshop.php # Fecshop的公用配置，一般是组件和服务的配置。

6.@vendor/fancyecommerce/fecshop/app/appfront/config/appfront.php # Fecshop appfront这个入口的独有配置，里面包含模块和一些组件的配置。

第7部分为第三方开发的扩展的配置引入部分，您使用的第三方插件的配置，可以在这里引入进来。

7.第三方配置文件地址。

8和9部分，是二次开发者的配置文件部分：

8.@common/config/fecshop\_local.php # 二次开发者的各个入口的公用配置

9.@app/config/fecshop\_local.php # 二次开发者的当前入口的独有配置

上面8个配置文件的内容，通过yii\helpers\ArrayHelper::merge()函数合并，最后 形成一个数组，在合并的过程中，1的优先级最低，8个优先级最高，优先级高的配置 项可以覆盖优先级低的配置项。

覆盖顺序：

1.fecshop（5-6）可以覆盖前面Yii2的配置项(1-4)

2.第三方扩展（7），可以覆盖fecshop的配置项(1-6)

3.二次开发者（8-9）可以覆盖上面所有的配置(1-7)

#### 2.对于运费的配置，是通过csv的方式存储公式，路径为

@common/config/shipping/fast\_shipping.csv

#### 3.yii class map重写机制

$yiiClassMap = require(\_\_DIR\_\_ . '/../config/YiiClassMap.php');

if(is\_array($yiiClassMap) && !empty($yiiClassMap)){

foreach($yiiClassMap as $namespace => $filePath){

Yii::$classMap[$namespace] = $filePath;

}

}

通过classMap的配置，可以重写任意一个class文件，譬如

<?php

return [

'fecshop\app\appfront\helper\test\My' => '@appfront/helper/My.php',

];

具体使用方法参看文章：[通过配置的方式重写某个Yii2 文件 或第三方扩展文件](http://www.fancyecommerce.com/2016/10/13/%e9%80%9a%e8%bf%87%e9%85%8d%e7%bd%ae%e7%9a%84%e6%96%b9%e5%bc%8f%e9%87%8d%e5%86%99%e6%9f%90%e4%b8%aayii2-%e6%96%87%e4%bb%b6-%e6%88%96%e7%ac%ac%e4%b8%89%e6%96%b9%e6%89%a9%e5%b1%95%e6%96%87%e4%bb%b6/)

## 优先级部分配置细化 - yii\helpers\ArrayHelper::merge()部分

上面的9个配置文件，有的配置文件还包含其他的配置文件，下面逐一列举

### 1.@vendor/fancyecommerce/fecshop/config/fecshop.php

该配置文件包含:

1.1@vendor/fancyecommerce/fecshop/config/services/\*.php ,这个文件夹下面所有的php配置文件，这里面是各个Fecshop Services（服务）的配置

1.2@vendor/fancyecommerce/fecshop/config/components/\*.php ，这个文件夹下面的所有的php文件，这里面是各个Fecshop Component（组件）的配置

### 2.@vendor/fancyecommerce/fecshop/app/appfront/config/appfront.php

该配置文件包含：

2.1@vendor/fancyecommerce/fecshop/app/appfront/config/params.php 这里面是一些参数的默认配置。

2.2@vendor/fancyecommerce/fecshop/app/appfront/config/modules/\*.php 这个文件夹下面所有的php配置文件，这里面是各个模块对应的配置

### 3.@common/config/fecshop\_local.php

该配置文件包含：

3.1@common/config/fecshop\_local\_modules/\*.php 这个文件夹下面所有的php配置文件，这里面是二次开发的公用部分，添加或重写的 模块方面的配置

3.2@common/config/fecshop\_local\_services/\*.php 这个文件夹下面所有的php配置文件，这里面是二次开发的公用部分，添加或重写的 服务方面的配置

4.@app/config/fecshop\_local.php

该配置文件包含：

4.1@app/config/fecshop\_local\_modules/\*.php 这个文件夹下面所有的php配置文件，这里面是二次开发的公用部分，添加或重写的 模块方面的配置

4.2@app/config/fecshop\_local\_services/\*.php 这个文件夹下面所有的php配置文件，这里面是二次开发的公用部分，添加或重写的 服务方面的配置

一开始说的9个配置文件，以及对应的这些细分的配置文件，共同组合起来合并，形成 fecshop的最终配置。

通过配置项的覆盖，我们就可以重写各个组件，服务等功能，二次开发自己的功能。

## 配置加速

配置文件的合并是在index.php文件中， 为了加速，您可以把合并后的数组放到缓存中，然后直接读取出来，节省数据合并带来的开销。 如果您更新了配置，您需要刷新一下缓存让更新的配置生效。

# Fecshop 手机检测跳转

当手机访问pc端(appfront)url，可以通过设置自动跳转到手机html5（apphtml5）端。

### 配置设置

打开配置文件 @appfront/config/fecshop\_local\_services/Store.php

可以看到配置代码：

'fecshop.appfront.fancyecommerce.com' => [

'language' => 'en\_US', // 语言简码需要在@common/config/fecshop\_local\_services/FecshopLang.php 中定义。

'languageName' => 'English', // 语言简码对应的文字名称，将会出现在语言切换列表中显示。

'localThemeDir' => '@appfront/theme/terry/theme01', // 设置当前store对应的模板路径。关于多模板的方面的知识，您可以参看fecshop多模板的知识。

'thirdThemeDir' => [], // 第三方模板路径，数组，可以多个路径

'currency' => 'USD', // 当前store的默认货币,这个货币简码，必须在货币配置中配置

/\*

\* 当设备满足什么条件的时候，进行跳转。

\* 这种方式不怎么高效，最好的方式是在nginx中配置。

\*/

'mobile' => [

'enable' => true,

'condition' => ['phone', 'tablet'], // phone 代表手机，tablet代表平板，当都填写，代表手机和平板都会进行跳转

'redirectDomain' => 'fecshop.apphtml5.fancyecommerce.com', // 如果是移动设备访问进行域名跳转，这里填写的值为store key

'https' => false, // 手机端url是否支持https,如果支持，设置https为true，如果不支持，设置为false

],

当 mobile.enable 设置为true，当访问pc端会跳转到html5端。

### 原理

打开@fecshop/services/Store.php ，您可以看到 html5DevideCheckAndRedirect()方法：

/\*\*

\* @property $store\_code | String

\* @property $store | Array

\* mobile devide url redirect.

\*/

protected function html5DevideCheckAndRedirect($store\_code, $store)

{

if (!isset($store['mobile'])) { //检查有没有配置html5跳转参数

return;

}

$mobileDetect = Yii::$service->helper->mobileDetect;

$enable = isset($store['mobile']['enable']) ? $store['mobile']['enable'] : false;

if (!$enable) { //检查有没有启用html5跳转

return;

}

$condition = isset($store['mobile']['condition']) ? $store['mobile']['condition'] : false; //获取跳转条件，如果没有设置跳转条件，默认为false

$redirectDomain = isset($store['mobile']['redirectDomain']) ? $store['mobile']['redirectDomain'] : false; //获取跳转地址,默认为false

if (is\_array($condition) && !empty($condition) && !empty($redirectDomain)) { //当满足跳转条件存在，并且跳转地址存在时，进行以下步骤...

$mobile\_https = (isset($store['mobile']['https']) && $store['mobile']['https']) ? true : false;

if (in\_array('phone', $condition) && in\_array('tablet', $condition)) {

if ($mobileDetect->isMobile()) { //检查是不是手机

$this->redirectMobile($store\_code, $redirectDomain, $mobile\_https);

}

} elseif (in\_array('phone', $condition)) {

if ($mobileDetect->isMobile() && !$mobileDetect->isTablet()) { //检查是不是平板电脑

$this->redirectMobile($store\_code, $redirectDomain, $mobile\_https);

}

} elseif (in\_array('tablet', $condition)) {

if ($mobileDetect->isTablet()) {

$this->redirectMobile($store\_code, $redirectDomain, $mobile\_https);

}

}

}

}

/\*\*

\* @property $store\_code | String

\* @property $redirectDomain | String

\* 设备满足什么条件的时候进行跳转。

\*/

protected function redirectMobile($store\_code, $redirectDomain, $mobile\_https)

{

$currentUrl = Yii::$service->url->getCurrentUrl();

$redirectUrl = str\_replace($store\_code, $redirectDomain, $currentUrl);

// pc端跳转到html5，可能一个是https，一个是http，因此需要下面的代码进行转换。

if ($mobile\_https) {

if (strstr($redirectUrl,'https://') || strstr($redirectUrl,'http://')) {

$redirectUrl = str\_replace('http://','https://',$redirectUrl);

} else {

$redirectUrl = 'https:'.$redirectUrl;

}

} else {

if (strstr($redirectUrl,'https://') || strstr($redirectUrl,'http://')) {

$redirectUrl = str\_replace('https://','http://',$redirectUrl);

} else {

$redirectUrl = 'http:'.$redirectUrl;

}

}

header('Location:'.$redirectUrl);

}

html5DevideCheckAndRedirect方法检测是否是手机访问，如果是，则进行跳转。

# Fecshop 货币

1. [货币配置](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_currency.html)

fecsop 支持多货币

## 货币配置

您可以在@common/config/fecshop\_local\_services/Page.php中进行配置货币。

'currency' => [

'baseCurrecy' => 'USD', # 基础货币，后台产品的价格都使用基础货币填写价格值。

'defaultCurrency' => 'USD', # 默认货币，如果store不设置货币，就使用这个store默认货币

'currencys' => [

'USD' => [ # 货币简码，USD代表美元，这个是国际标准

'rate' => 1, # 汇率 当前货币/基础货币的比值，譬如，人民币/美元 = 7

'symbol' => '$', #货币符号

],

'EUR' => [ # 欧元

'rate' => 0.93,# 汇率

'symbol' => '€',

],

//'AUD' => [

// 'rate' => 1.33,

// 'symbol' => 'AU$',

//],

'GBP' => [ # 英镑

'rate' => 0.8,

'symbol' => '£',

],

'CNY' => [ # 人民币

'rate' => 6.87,

'symbol' => '￥',

],

],

],

设置货币后，在前端的顶部的语言切换选项中，就能看到您添加或者修改的货币配置。

另外在store中设置的货币，必须在此处存在，否则将会报错。

# ﻿Fecshop Store

fecshop是支持多store的，store一般是用来构建多语言站点， 可以为每一个store指定不同的语言和url结构，设置不同的语言，默认货币,模板等。

### 1.Store的配置：

您可以在@appfront/config/fecshop\_local\_services/Store.php中配置Store

示例代码：

'stores' => [

# store key：域名去掉http部分，作为key，这个必须这样定义。

'fecshop.appfront.fancyecommerce.com' => [

'language' => 'en\_US', # 语言简码需要在@common/config/fecshop\_local\_services/FecshopLang.php 中定义。

'languageName' => 'English', # 语言简码对应的文字名称，将会出现在语言切换列表中显示。

'localThemeDir' => '@appfront/theme/terry/theme01', # 设置当前store对应的模板路径。关于多模板的方面的知识，您可以参看fecshop多模板的知识。

'thirdThemeDir' => [], # 第三方模板路径，数组，可以多个路径

'currency' => 'USD', # 当前store的默认货币,这个货币简码，必须在货币配置中配置

'mobile' => [ # 当设备满足什么条件的时候，进行跳转。

'enable' => true,

'condition' => ['phone','tablet'], # phone 代表手机，tablet代表平板

'redirectUrl' => 'fecshop.apphtml5.fancyecommerce.com', # 如果是移动设备访问进行域名跳转，这里填写的值为store key

],

# 第三方账号登录配置

'thirdLogin' => [

# facebook账号登录

'facebook' =>[ #fb api配置 ，fb可以一个app设置pc和手机web两个域名

'facebook\_app\_id' => '1849609081926823',

'facebook\_app\_secret' => '2e097a6d5a424531770fc05760dd7139',

],

# google账号登录

"google" => [ #谷歌api visit https://code.google.com/apis/console to generate your google api

'CLIENT\_ID' => '380372364773-qdj1seag9bh2n0pgrhcv2r5uoc58ltp3.apps.googleusercontent.com',

'CLIENT\_SECRET' => 'ei8RaoCDoAlIeh1nHYm0rrwO',

],

]

],

需要注意的是您的store key，如果您的首页是 http://fecshop.appfront.fancyecommerce.com，那么您的key为fecshop.appfront.fancyecommerce.com ，如果您的首页为 http://fecshop.appfront.fancyecommerce.com/fr，那么您的key为 fecshop.appfront.fancyecommerce.com/fr

您可以为您的每一个store指定特定的域名，用不同的子域名或用不同的语言路径的 方式，来构建您的多语言的网站。

### 2.多语言站点的构建方式：

#### 1.子域名的方式构建多语言站点，譬如：

www.fecshop.com（英文）,fr.fecshop.com（法文） de.fecshop.com（德文）

这种方式可以在@app/web/index.php中设置 ini\_set('session.cookie\_domain', '.fancyecommerce.com'); 来进行子域名之间session的共享。

#### 2.url路径的方式构建多语言站点，譬如：

www.fecshop.com（英文）,www.fecshop.com/fr/（法文）,www.fecshop.com/de/（德文） 这种多语言方式因为都是同样的域名，只是后缀不同，因此， 不需要像第1步那样做session共享

#### 3.独立多域名方式:

www.fecshop.com（英文）,www.fecshop.fr（法文）,www.fancyecommerce.de（德文）

这种方式，session不能共享（当然，您可以做跨域session共享，但是是比较麻烦的， 因此，建议使用第一种，子域的方式），因此购物车和登录信息在切换站点的时候，需要重新 进行登录。

### 4.store原理

4.1 在入口文件 @app/web/index.php ，可以加载所有的配置文件，合并

4.2 对于appfront appserver apphtml5这三个入口是需要store的， 因此需要在Yii2 bootstrap初始化过程中就开始执行，下面以appfront入口为例子讲解

@fecshop/app/appfront/config/appfront.php ， 可以看到配置：

'bootstrap' => ['store'],

也就是在初始化的时候执行store component，关于Yii2的bootstrap可以查看资料： [启动引导（Bootstrapping）](http://www.yiichina.com/doc/guide/2.0/runtime-bootstrapping) , [yii2 初始化的bootstrap过程 -引导](http://www.fancyecommerce.com/2016/05/18/yii2-%E5%88%9D%E5%A7%8B%E5%8C%96%E7%9A%84bootstrap%E8%BF%87%E7%A8%8B-%E5%BC%95%E5%AF%BC/)

下面我们去找store component

4.3 打开配置文件： @fecshop/config/components/Store.php 文件可以看到：

return [

'store' => [

'class' => 'fecshop\components\Store',

],

];

通过配置文件，我们找到store component , @fecshop\components\Store.php, 代码如下：

class Store extends Component implements BootstrapInterface

{

public function bootstrap($app)

{

Yii::$service->store->bootstrap($app);

}

}

通过上面的代码，我们可以看到，就是执行store services 的bootstrap方法 ，通过配置文件：@fecshop/config/services/Store.php，我们可以看到

'store' => [

'class' => 'fecshop\services\Store',

我们打开 @fecshop\services\Store.php ,查看 bootstrap()方法， 可以看到很多store的初始化代码，譬如语言，货币等， 都是在这个函数里面进行的，您可以细看里面的代码，具体您可以查看这个文件，这里不做代码粘贴了。

# Fecshop 多语言

1. [一：语言的配置](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_mutil_lang.html)
2. [二：翻译](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_mutil_lang.html)
3. [三：各个入口的语言配置](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_mutil_lang.html)
4. [四：如何添加一门新语言](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_mutil_lang.html)

fecshop 支持多语言翻译，可以通过多个途径进行语言的切换，一共分为四个大部分： 数据库数据的翻译，网页内容数据的翻译，网站提示和报错等信息的翻译，邮件内容的 语言翻译，当您为一个store指定了语言，那么就会按照上面四种途径进行相应的语言翻译， ,进而形成对应语言的翻译语言。

## 一：语言的配置

在fecshop FecshopLang 服务 中可以配置语言选项，你可以在 @common/config/fecshop\_local\_services/FecshopLang.php中进行配置语言项，譬如：

'fecshoplang' => [

//'class' => 'fecshop\services\FecshopLang',

// mongoSearchLangName 在各个语言下字段参考资料如下：（不支持中文）

// https://docs.mongodb.com/manual/reference/text-search-languages/#text-search-languages

'allLangCode' => [

# 'en\_US' 是标准语言简码 code对应的值en取 “标准语言简码”的前两位字符，

# 该值设置后，进行了产品分类数据的添加后，不能修改，否则会出现部分翻译语言丢失。

'en\_US' => [

'code' => 'en',

],

'fr\_FR' => [

'code' => 'fr',

],

'de\_DE' => [

'code' => 'de',

],

'es\_ES' => [

'code' => 'es',

],

'ru\_RU' => [

'code' => 'ru',

],

'pt\_PT' => [

'code' => 'pt',

],

'zh\_CN' => [

'code' => 'zh',

],

],

# 默认语言。

'defaultLangCode' => 'en',

],

上面是fecshop默认的几个语言配置，您可以根据自己的业务需求进行语言的设置和添加。

fecshop的每个store设置的不同语言，以及对应的产品搜索(mongodb full search需要根据语言进行切词)，都要依赖上面的配置。

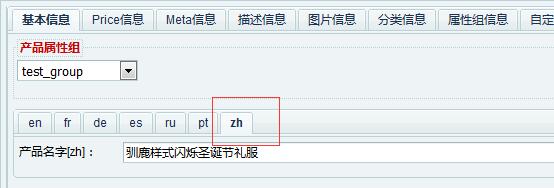
[语言简码表](http://blog.csdn.net/wed110/article/details/50886195)

## 二：翻译

### 数据库数据

数据库中的数据，譬如产品表中的产品名字，产品描述等，分类表中的分类名字和分类描述， 在数据库表中都是保存了多份语言数据





在网站前台展示数据的时候，根据当前的store设置的语言，将相应语言的数据取出来展示即可。 如果相应语言的值在数据库中没有添加，那么将会把默认语言的值取出来作为当前语言的值。

### 网页内容语言的翻译

网站内容语言的翻译，是依靠的翻译文件，在实现方面依赖的Yii::t()函数，关于 Yii2多语言翻译的知识，可以参看地址 [Yii2多语言](http://www.yiichina.com/doc/guide/2.0/tutorial-i18n)

### Fecshop的翻译文件路径为：（举例appfront入口）

@vendor/fancyecommerce/fecshop/app/appfront/languages/，在这个文件夹下面可以看到 各个语言的文件包，进入相应语言，就可以看到翻译文件，譬如中文的翻译文件路径为 ：@vendor/fancyecommerce/fecshop/app/appfront/languages/zh\_CN/appfront.php ，在这个文件里面就可以看到所有的中文翻译内容，如果您想重写或者添加新 的翻译，可以到 @appfront/languages/zh\_CN/appfront.php 中添加或者重写翻译 数组，注意，这里使用的是php的数组，需要按照相应格式填写，否则会报错。

在翻译文件中，您可能看到这样的带有{}的部分，譬如下面的 {passwdMinLength}, 这是一个动态变量，这个请不要翻译和做其他的任何改动，直接复制上去即可，譬如下面的 翻译，这个值是由php动态计算而来。

'Password length must be greater than or equal to {passwdMinLength}'

=> '密码长度必须大于或等于{passwdMinLength}',

'Password length must be less than or equal to {passwdMaxLength}'

=> '密码长度必须小于或等于{passwdMaxLength}',

'The passwords are inconsistent'

=> '密码不一致',

### 提示报错信息的翻译

在网站顶部，会出现一些提示信息和报错信息，譬如您注册邮箱的时候，会提示你的格式不正确等， 这些数据也是用翻译文件的方式，方法与上面类似，都是在在同一个文件中 添加翻译内容,方法和方式同上。

### 邮件内容的翻译

邮件内容的模板，是在theme中，譬如appfront的邮件模板，是在 @fecshop/app/appfront/theme/base/default/mailer下面,其中, @fecshop/app/appfront/theme是模板文件夹路径，base是模板包名， default是模板名，mailer是邮件theme文件夹，在mailer路径下，打开 customer/account/forgotpassword,里面是用户忘记密码发送的邮件， 其中subject\_开头的都是邮件的标题，body\_开头的都是邮件的内容 en结尾的代表英文语言的邮件，zh结尾的代表中文语言的邮件，打开相应文件 从en复制一份到对应的语言，然后修改里面的语言即可， 有一些语言，里面没有对应的文件，您新建即可。

添加或者修改邮件内容，举例：

如果您想修改 @fecshop/app/appfront/theme/base/default/mailer/customer/account/forgotpassword/body\_zh.php, 里面的内容，你可以在 @appfront/theme/terry/theme01/里面新建文件 mailer/customer/account/forgotpassword/body\_zh.php（如果存在，则不需要创建） 然后将上面的文件内容复制到新建的文件中，进行修改即可，这样就完成了邮件内容的添加或者重写。

通过以上四种方式的翻译，我们就完成了整个fecshop多语言翻译的闭环。

注意： 如果您不添加翻译，则默认就会使用英语的翻译。

关于邮件更详细的介绍，可以参看：[Fecshop 邮件](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/fecshop_email.php)

## 三：各个入口的语言配置

当我们在上面的步骤中设置了语言，以及各个语言对应的翻译文件后， 我们需要为各个入口设置对应的默认语言

### 设置各个域名下面的默认语言

打开 @app/config/fecshop\_local\_services/Store.php

<?php

return [

'store' => [

'class' => 'fecshop\services\Store',

'stores' => [

// store key：域名去掉http部分，作为key，这个必须这样定义。

'fecshop.appfront.fancyecommerce.com' => [

'language' => 'en\_US', // 语言简码需要在@common/config/fecshop\_local\_services/FecshopLang.php 中定义。

'languageName' => 'English', // 语言简码对应的文字名称，将会出现在语言切换列表中显示。

'localThemeDir' => '@appfront/theme/terry/theme01', // 设置当前store对应的模板路径。关于多模板的方面的知识，您可以参看fecshop多模板的知识。

'thirdThemeDir' => [], // 第三方模板路径，数组，可以多个路径

'currency' => 'USD', // 当前store的默认货币,这个货币简码，必须在货币配置中配置

/\*

\* 当设备满足什么条件的时候，进行跳转。

\* 这种方式不怎么高效，最好的方式是在nginx中配置。

\*/

...

设置：

language ：第一部分中， @common/config/fecshop\_local\_services/FecshopLang.php 中进行配置语言项， 这里填写的值，必须该配置文件中出现。

languageName 自定义语言label，这个会在前端显示出来具体的名字（因为简码，用户 是看不明白的），譬如在appfront端的顶部左侧显示的多语言切换列表，apphtml5端 首页底部显示的多语言切换列表

然后就可以访问查看了，当把该入口的各个域名对应的默认语言设置好后， 访问这个语言，就可以看到相应的语言了。

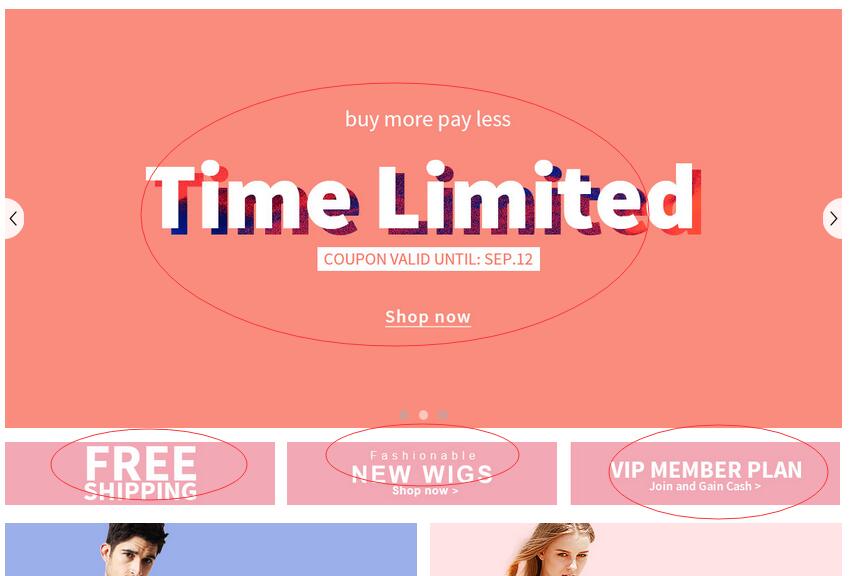
## 四：如何添加一门新语言

参看文档：<http://www.fecshop.com/topic/616>

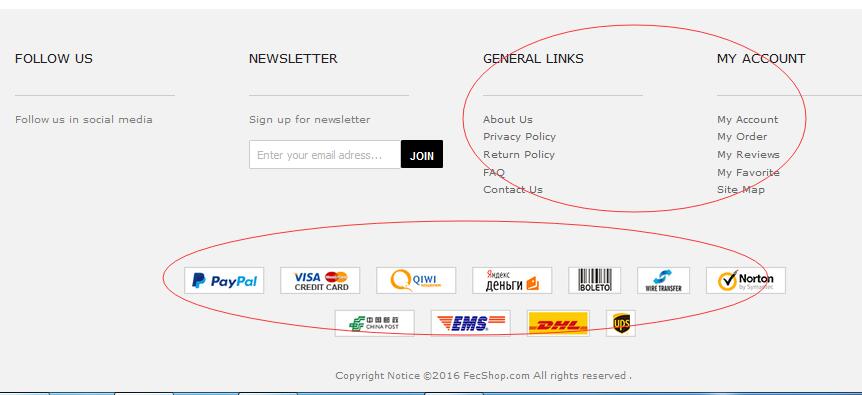
# Fecshop 静态块

fecshop 静态块是指一块静态的html段，可以在后台通过配置的方式进行修改， 这样，可以很方面的让工作人员进行修改。

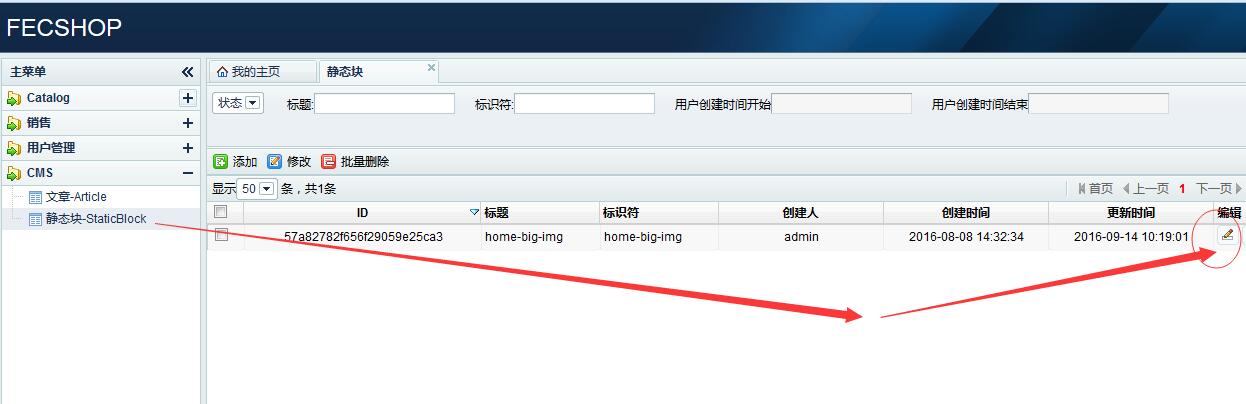
譬如首页的轮换大图，

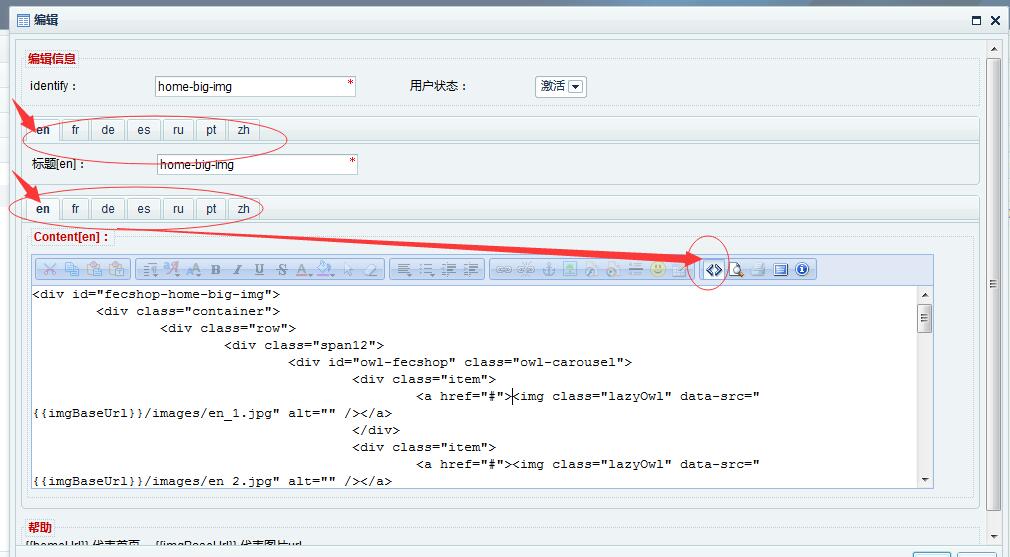


页面底部的文件条款等区块



您可以在后台进行编辑





在content（内容部分），{{homeUrl}} 代表首页。 {{imgBaseUrl}} 代表图片url。 其中，{{imgBaseUrl}}为 对应的文件路径为@appimage/app入口名/

### 使用方法

1.首先在后台新建一个static block，譬如key为home-big-img

2.在代码中调用

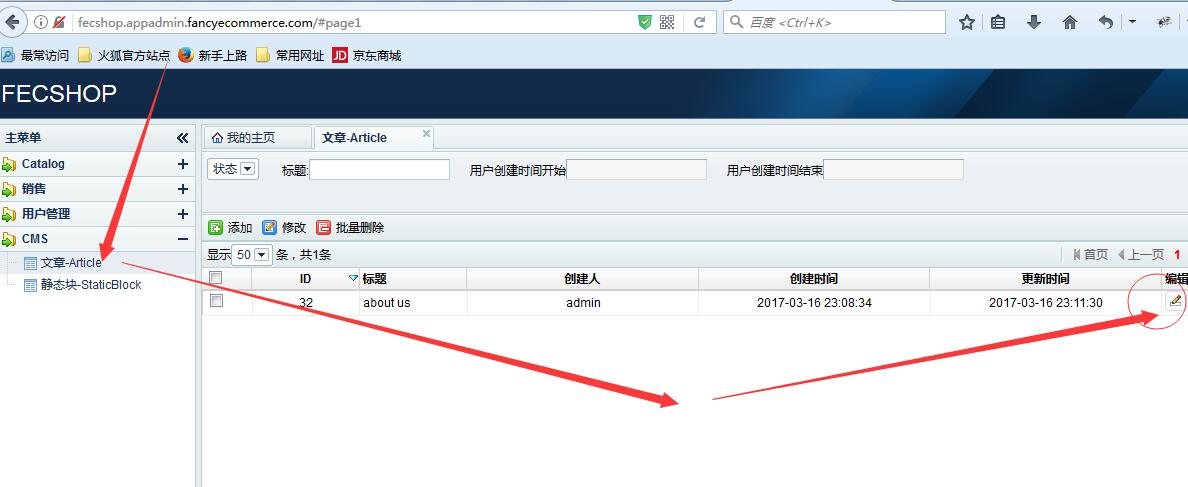
Yii::$service->cms->staticblock->getStoreContentByIdentify('home-big-img','appfront')

第一个参数为key，第二个参数为入口的名字，这个参数的作用 是为了生成url（域名部分）

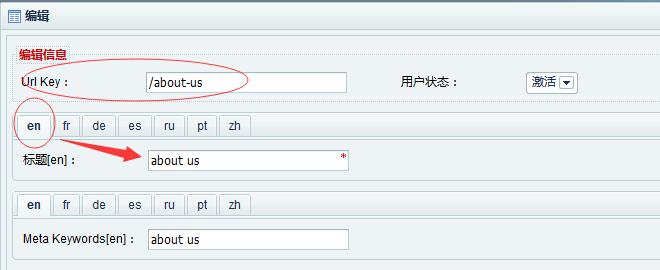
# Fecshop page页（文章）

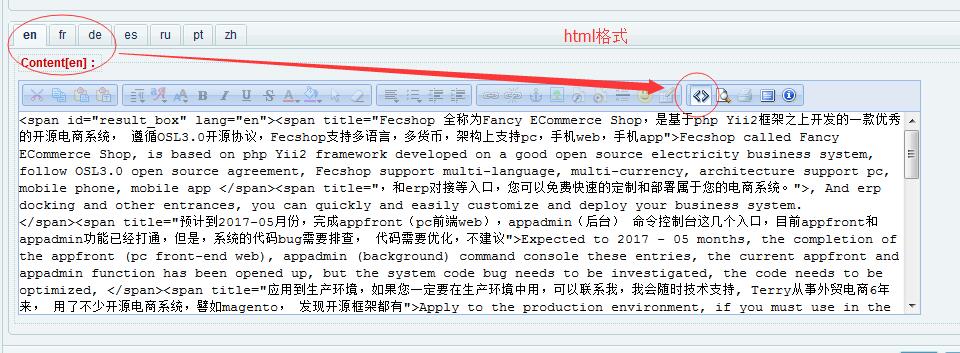
fecshop page,指的是一些单页面，一般用来写文字条款，譬如http://fecshop.appfront.fancyecommerce.com/index.php/about-us

您可以在后台编辑



弹出编辑框





然后保存，保存后，查看生成的url key就是url key，当然您也可以自定义url key。 用url key放到域名后面就可以访问了，譬如我新建的关于我们的 url key为 about-us， 那么我访问 <http://fecshop.appfront.fancyecommerce.com/index.php/about-us> 即可访问我新建的page页。

通过page，可以快速的新建一些文字页面。

# fecshop 首页

fecshop 首页，也就是home页，属于进入网站的主页面

### 1.配置

打开配置文件 @fecshop/app/appfront/config/appfront.php ，可以看到如下的代码：

// 首页对应的url key

'urlManager' => [

'rules' => [

'' => 'cms/home/index',

],

],

这是Yii2的rules配置，配置 '' => 'cms/home/index',， 这个配置代表的是首页地址对应 cms/home/index， 也就是cms模块下面的HomeController里面的actionIndex()方法， 也就是文件 @fecshop\app\appfront\modules\Cms\controllers\HomeController.php 里面的actionIndex()方法，

对于Yii2 controller的知识，可以参看：<http://www.yiichina.com/doc/guide/2.0/structure-controllers>

对于Yii2模块，可以参看：<http://www.yiichina.com/doc/guide/2.0/structure-modules>

url routing 可以参看：<http://www.yiichina.com/doc/guide/2.0/runtime-routing>

# Fecshop 404页

当一些url，找不到相应的controller，就会进入404页面，状态码为404，

404页面，使用的是Yii的 errorHandler组件，fecshop在配置文件 @fecshop/app/appfront/config/appfront.php中进行的配置。

'errorHandler' => [

'errorAction' => 'site/helper/error',

],

模板路径为：@fecshop/app/appfront/theme/base/front/site/helper/error.php

如果您想重写404页面，通过多模板重写的方式即可。

# Fecshop 小部件

fecshop小部件，不同于fecshop的小部件，fecshop的小部件是集数据库数据与view文件 生成的一个区块，譬如网站侧栏的热销产品，做成一个小部件，可以很方便的 在其他页面进行引入。当然，fecshop的head header（头部），footer（尾部）都是小部件实现的。

### 小部件结构：

小部件由两部分组成，动态数据提供部分 和 提供html代码的view部分，共同组合 ，显示出来小部件。

### 小部件配置：

@fecshop/config/services/Page.php里面进行小部件的配置，示例代码：

'widget' => [

'class' => 'fecshop\services\page\Widget',

'widgetConfig' => [

'head' => [

# 动态数据提供部分

'class' => 'fecshop\app\appfront\widgets\Head',

# 根据多模板的优先级，依次去模板找查找该文件，直到找到这个文件。

'view' => 'widgets/head.php',

# 缓存

'cache' => [

'enable' => false, # 是否开启

'timeout' => 4500, # 缓存过期时间

],

],

'header' => [

'class' => 'fecshop\app\appfront\widgets\Headers',

# 根据多模板的优先级，依次去模板找查找该文件，直到找到这个文件。

'view' => 'widgets/header.php',

'cache' => [

'enable' => false,

'timeout' => 4500,

],

],

'topsearch' => [

'view' => 'widgets/topsearch.php',

],

譬如header小部件，数据提供部分为:@fecshop\app\appfront\widgets\Headers.php, view部分为：@fecshop\app\appfront\theme\base\default\widgets\header.php。

其中，head header都是一个小部件，里面的class是动态数据提供部分， view是显示部分，cache是缓存，您可以通过该配置将缓存打开，以及设置缓存的超时时间。

小部件的view部分遵循fecshop的多模板机制，可以在二开theme路径下添加 相应文件实现view部分的重写，可以通过classMap机制进行动态数据提供class文件 的重写。

### 小部件的使用：

<?= Yii::$service->page->widget->render('header',$this); ?>

上面传的的$this，在view文件中对应的变量为：$parentThis，举例子可以参看head小部件

<?= Yii::$service->page->widget->render('head',$this); ?>

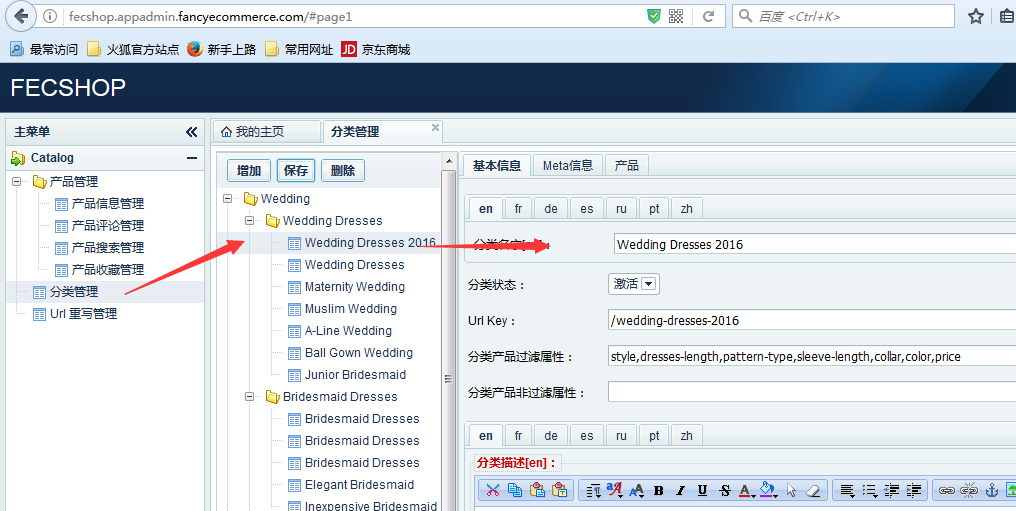
<?php $parentThis->head() ?>

$parentThis就是render()函数调用传入的$this变量。

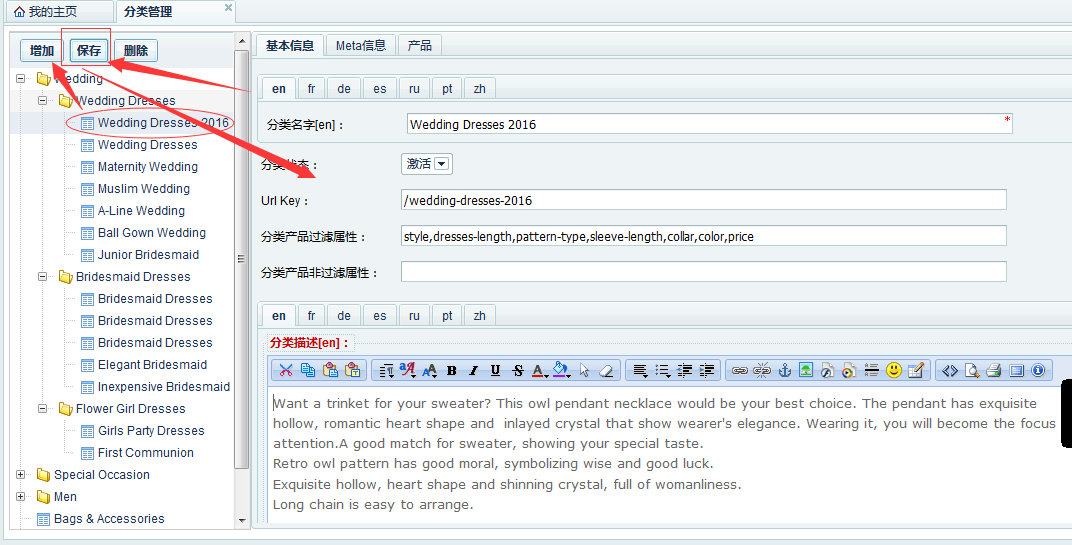
# Fecshop 分类

这里说的分类，就是在前台菜单导航栏里面的内容，包含home，产品分类，自定义分类等。

### 产品分类，自定义分类等

1.您可以通过下面的方式查看分类信息 

2.如果您想增加一个子分类，那么首先点击父分类，然后点击增加，在刷新后的右侧， 填写分类的信息，然后保存即可完成。



3.如果您想给分类添加产品可以按照下图所示，勾选产品。 

全部产品检索按钮：点击后，列表中可以查看所有的产品（已经勾选过的产品， 上面会有勾选的标志），您可以为这个分类勾选相应的产品

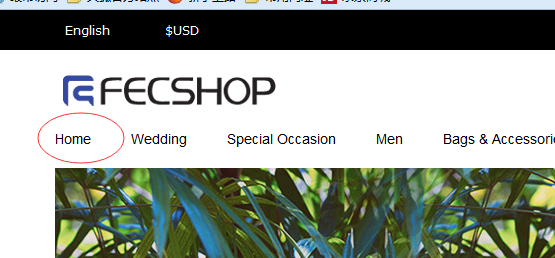
当前分类检索按钮：点击后，只显示当前分类已经勾选了的产品。

4.如果您新建了产品分类，但是没有勾选产品，那么在产品编辑的地方，也可以勾选产品。 勾选后，在分类部分，可以看到对应的产品。

5.新建分类后，在前端展示分类部分会显示新建的分类。

### HOME部分：

HOME部分，也就是在菜单，图片如下：



您可以通过配置的方式来配置是否显示home，配置的文件为： @appfront/config/fecshop\_local\_services/Page.php

'menu' => [

'displayHome' => [

'enable' => true, # 是否在菜单中显示home

'display'=> 'Home', # 显示对应的字符。

],

### 自定义菜单部分

对于fecshop，菜单的主体是产品分类菜单，其他的归结为自定义菜单。

自定义的配置的文件为： @appfront/config/fecshop\_local\_services/Page.php

/\*\*

\* 在菜单前面部分（产品分类菜单的前面部分）的自定义菜单。参考behindCustomMenu里面的格式

\*/

'frontCustomMenu' => [

],

/\*\*

\* 在菜单后面部分（产品分类菜单的前面部分）的自定义菜单

\*/

'behindCustomMenu' => [

[

'name' => 'custom menu', # 菜单名字

'urlPath' => '/my-custom-menu.html', # 菜单对应的url

'childMenu' => [ # 子菜单

[

'name' => 'my custom menu 2',

'urlPath' => '/my-custom-menu-2.html',

],

[

'name' => 'my custom menu 3',

'urlPath' => '/my-custom-menu-2.html',

'childMenu' => [

[

'name' => 'my custom menu 3',

'urlPath' => '/my-custom-menu-2.html',

],

[

'name' => 'my custom menu 3',

'urlPath' => '/my-custom-menu-2.html',

],

],

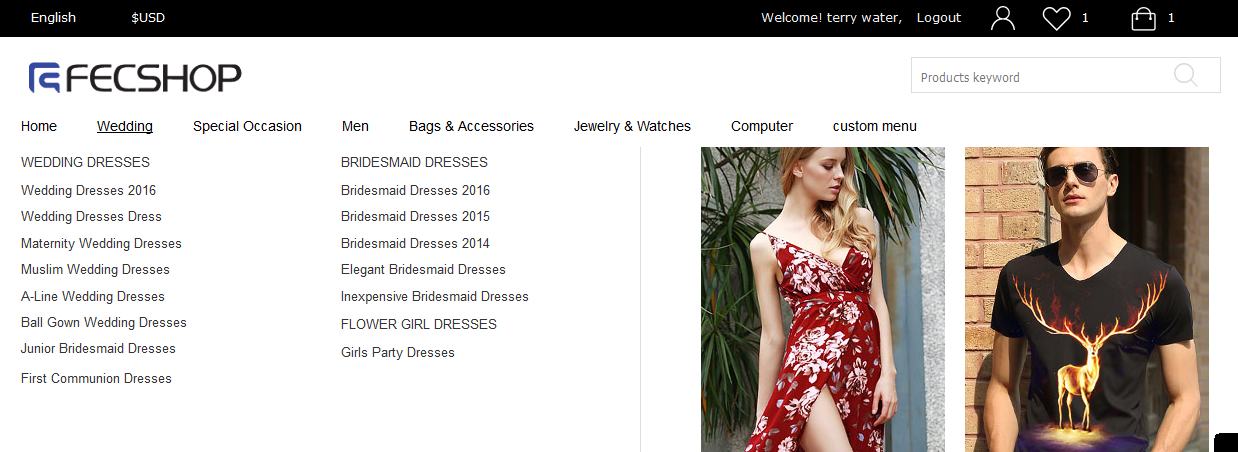
],

],

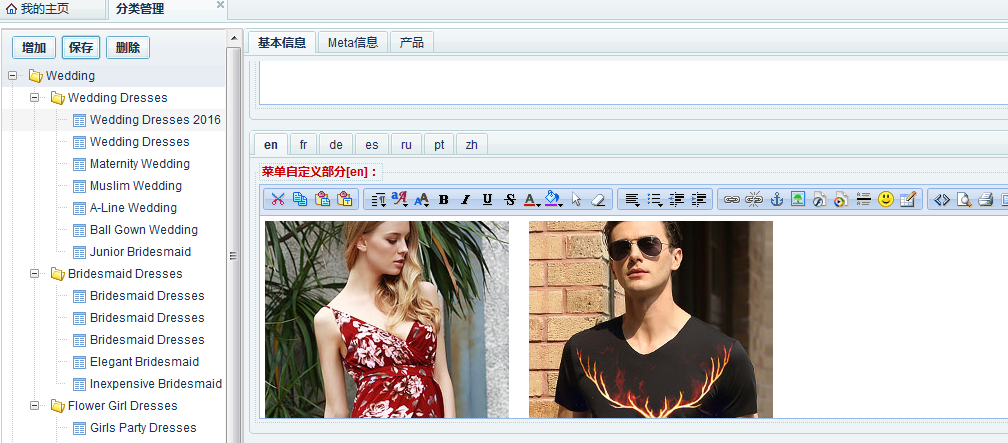
],

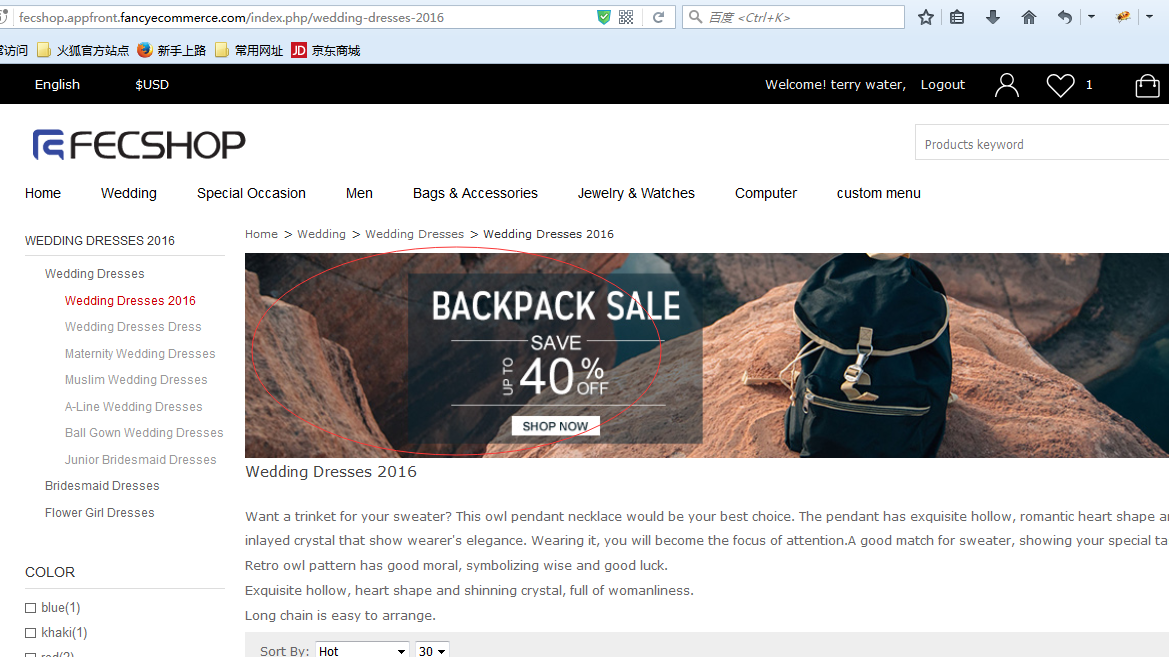
],

在上面自定义配置了菜单，就会在菜单导航栏中显示出来。

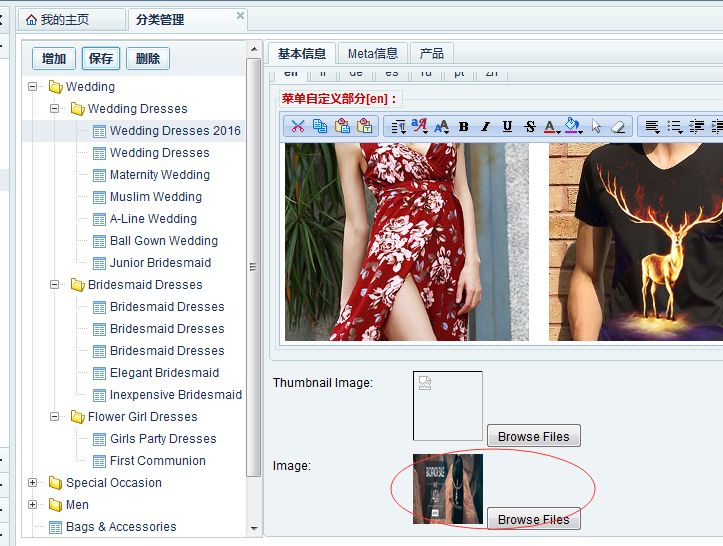


另外，需要说明一下分类编辑里面的内容，对于一级分类弹出部分右侧的图片内容 ，是在分类这里进行编辑的：

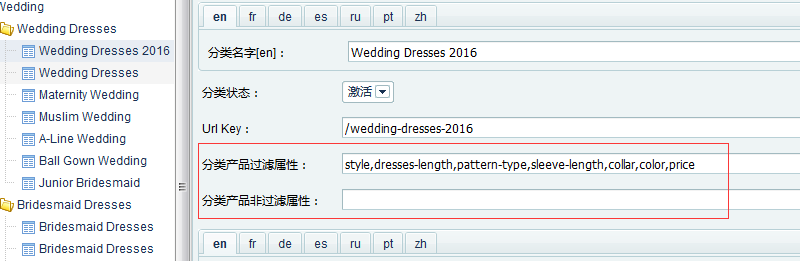


对于下图中，点击分类进如分类详细页面的图片，如图： 

是在这里编辑的，如图：



另外：

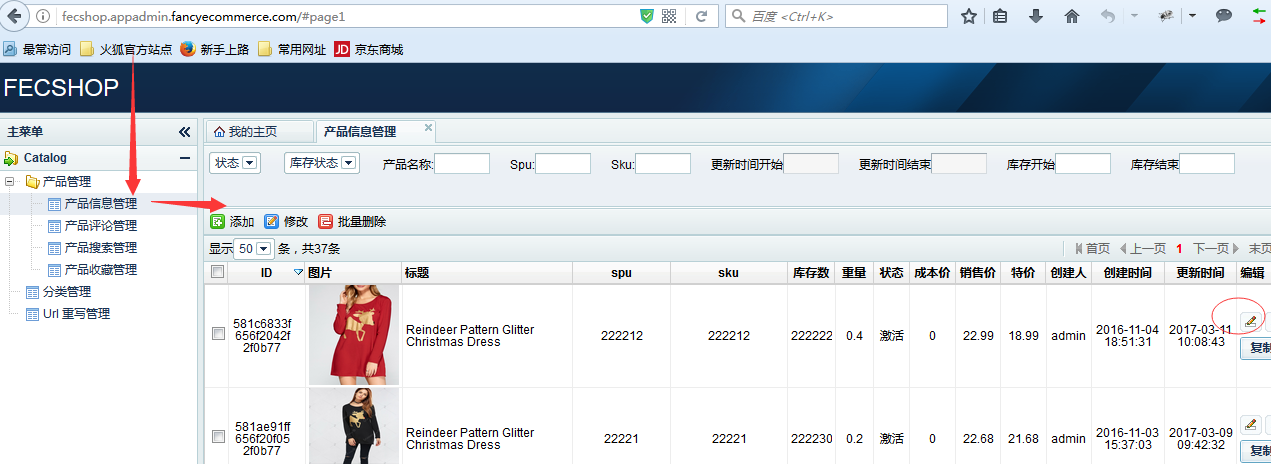


这里是分类中侧栏的产品过滤属性，这个在产品中详细解说。

# Fecshop 产品

作为电商系统，很多地方都是围绕产品展开的，譬如产品分类，产品搜索，产品评论， 产品收藏，购物车产品，订单产品等等。

### 后台产品编辑



### 产品属性组：

产品除了价格，名字，库存等一些公用的属性外， 产品根据不同的分类，会有一些独有的属性， 譬如电脑类产品会有cpu型号，主板型号等属性， 衣服类有颜色尺码等属性，在这里引入产品属性组的概念， 一类产品从属于一个属性组。

属性组的配置：

打开文件：@common/config/fecshop\_local\_services/Product.php文件 可以看到对属性组的配置

配置文件的内容太多，这么不贴出来的，您可以打开这个文件查看具体内容，里面有注释

属性组里面的属性有类别，目前有三个类：

1、**spu\_attr**

1.1 spu和sku的概念：sku是产品的唯一标示，最小库存单位，譬如一款鞋子为一个spu，这款鞋子的不同的颜色和不同的尺码为一个sku，

1.2 什么是spu属性（spu\_attr），当一款鞋子的不同的sku是颜色和尺码的不同，那么颜色和尺码就是这款鞋子的spu属性（spu\_attr）。

1.3 spu\_attr是用来在产品详细页面，将这个spu下面的所有sku串联起来，譬如页面：<http://fecshop.appfront.fancyecommerce.com/index.php/raglan-sleeves-letter-printed-crew-neck-sweatshirt-53386451-77774122> 访问这款衣服的每一个SKU，都会把其他的sku列出来，点击后，就进去其他sku的页面，您会发现url是变化的，这个是类似于京东的方式。 注意：

1.3.1 因为组合的复杂性，spu attr最多设置2个属性，不能超过2个(如果超过会出问题。spu属性会都不显示)，超过两个请使用customer\_option的方式。也就是类似于淘宝的方式。

1.3.2 这里的属性需要严格按照下面的格式进行配置，如果您想配置自己的spu属性，那么

1.3.3 通过链接：<https://fecshop.appfront.fancyecommerce.com/raglan-sleeves-letter-printed-crew-neck-sweatshirt-41214715>您可以看到，color属性是用图片来显示的， 在spu\_attr中，第一个属性默认用图片显示，譬如：

'spu\_attr' => [ # spu用来区分sku的属性，譬如下面的属性的不同，对应不同的sku，进而是不同的库存

# 第一个属性会被用户当做图片来显示。

'color' => [

'dbtype'=> 'String',

'label' =>'颜色',

'name' =>'color',

'sort\_order' => 1,

'display'=>[

'type' => 'select',

'data' => [

'one-color' => 'one-color',

'red' => 'red',

'white' => 'white',

'black' => 'black',

'blue' => 'blue',

'green' => 'green',

'yellow' => 'yellow',

'gray' => 'gray',

'khaki' => 'khaki',

'ivory' => 'ivory',

'beige' => 'beige',

'orange' => 'orange',

'cyan' => 'cyan',

'leopard' => 'leopard',

'camouflage' => 'camouflage',

'silver' => 'silver',

'pink' => 'pink',

'purple' => 'purple',

'brown' => 'brown',

'golden' => 'golden',

'multicolor' => 'multicolor',

'white & blue' => 'White & Blue',

'white & black' => 'White & Black',

]

],

//'require' => 0,

//'default' => 2,

],

# 第二个属性不会当做图片来显示

'size' => [

'dbtype'=> 'String',

'label' =>'尺码',

'name' =>'size',

'sort\_order' => 2,

'display' =>[

'type' =>'select',

'data' =>[

'one-size' => 'one-size',

'S' => 'S',

'M' => 'M',

'L' => 'L',

'XL' => 'XL',

'XXL' => 'XXL',

'XXXL' => 'XXXL',

]

],

//'require' => 0,

//'default' => 2,

],

],

color 将会被当做图片来显示出来。

2、**general\_attr（普通属性）**，可以是各种格式的值，譬如时间格式，email格式，下拉条选择值等。这些属性可以用于分类侧栏属性过滤。

示例配置：

'general\_attr' => [

# 这是input type='text' 的类型

'my\_remark' => [

'dbtype'=> 'String',

'label'=>'我的备注',

'name'=>'my\_remark',

'display'=>[

'type' => 'inputString', # 字符串格式的属性

],

'require' => 0,

],

# 这是input type='email' 的类型

'my\_email' =>[

'dbtype'=> 'String',

'label'=>'我的邮箱',

'name'=>'my\_email',

'require' => 0,

'display'=>[

'type' => 'inputEmail', # 字符串格式的属性（email格式验证）

],

],

# 这是input type='date' 的类型

'my\_date' => [

'label'=>'我的日期',

'name'=>'my\_date',

'display'=>[

'type' => 'inputDate', # 字符串格式的属性（Date格式验证）

],

],

# 这是<select> 的类型

'style' => [

'dbtype'=> 'String',

'label' =>'类型',

'name' =>'style',

'display' =>[

'type' =>'select', # 下拉条选择格式的属性

'data' =>[

'Casual' => 'Casual',

'Cute' => 'Cute',

'Sexy & Club'=> 'Sexy & Club',

'Bohemian' => 'Bohemian',

'Vintage ' => 'Vintage ',

'Brief' => 'Brief',

'Work' => 'Work',

'Novelty' => 'Novelty',

]

],

],

'dresses-length' => [

'dbtype'=> 'String',

'label' =>'裙长',

'name' =>'dresses-length',

'display' =>[

'type' =>'select', # 下拉条选择格式的属性

'data' =>[

'Mini' => 'Mini',

'Knee-Length' => 'Knee-Length',

'Mid-Calf'=> 'Mid-Calf',

'Ankle-Length' => 'Ankle-Length',

'Floor-Length ' => 'Floor-Length ',

]

],

],

'pattern-type' => [

'dbtype'=> 'String',

'label' =>'款式',

'name' =>'pattern-type', # 属性名字

'display' =>[

'type' =>'select', # 下拉条选择格式的属性

'data' =>[

'Animal' => 'Animal',

'Character' => 'Character',

'Floral'=> 'Floral',

'Geometric ' => 'Geometric ',

'Leopard '=> 'Leopard ',

'Letter'=> 'Letter',

'Paisley'=> 'Paisley',

'Patchwork'=> 'Patchwork',

'Polka Dot'=> 'Polka Dot',

'Print'=> 'Print',

'Striped'=> 'Striped',

]

],

],

'sleeve-length' => [

'dbtype'=> 'String',

'label' =>'袖长',

'name' =>'sleeve-length',

'display' =>[

'type' =>'select',

'data' =>[

'Sleeveless' => 'Sleeveless',

'Short-Sleeves' => 'Short Sleeves',

'Half-Sleeves'=> 'Half Sleeves',

'3-4-Length-Sleeves ' => '3/4 Length Sleeves ',

'Long-Sleeves '=> 'Long Sleeves ',

]

],

],

'collar' => [

'dbtype'=> 'String',

'label' =>'领口', # 后台显示的中文名（目前后台只有中文）

'name' =>'collar',

'display' =>[

'type' =>'select',

'data' =>[

'Round Neck' => 'Round Neck', # 下拉条里面对应的各个可以选择的值。

'V-Neck' => 'V-Neck',

'Hooded' => 'Hooded',

'Turn-down-Collar' => 'Turn-down Collar',

]

],

//'require' => 0,

//'default' => 2,

],

],

3、**custom\_options用户自定义属性**，显示方式方面有点和spu属性类似，spu属性显示的方式是京东的方式，点击每一个选项是url跳转的 用户自定义类似于淘宝的方式，选择各个颜色尺码，页面是不跳转的，各个颜色尺码有相应的图片，库存，价格，sku等。 您可以查看演示地址：http://fecshop.appfront.fancyecommerce.com/index.php/reindeer-pattern-glitter-christmas-dress-86519596 您可以给产品属性组添加的属性类型就上面几种，在后台编辑产品的时候，选择不同的属性组，就会加载相应的属性出来。

示例配置：

'custom\_options' => [

'my\_color' => [

'dbtype'=> 'String', #类型

'label' =>'My Color', # 显示的名字

'name' =>'color', # 在数据库中存在的列名

'showAsImg' => true, # （在前端展示部分）通过图片的方式展示属性。譬如；http://fecshop.appfront.fancyecommerce.com/index.php/reindeer-pattern-glitter-christmas-dress-86519596，

# 你会发现，该属性对应的显示方式不是值，而是产品的图片。

'require' => 1, # 1代表是必填选项，0代表选填

'display'=>[

'type' => 'select',

'data' => [

'red' => 'red',

'white' => 'white',

'black' => 'black',

'blue' => 'blue',

'green' => 'green',

'yellow' => 'yellow',

'gray' => 'gray',

'khaki' => 'khaki',

'ivory' => 'ivory',

'beige' => 'beige',

'orange' => 'orange',

'cyan' => 'cyan',

'leopard' => 'leopard',

'camouflage' => 'camouflage',

'silver' => 'silver',

'pink' => 'pink',

'purple' => 'purple',

'brown' => 'brown',

'golden' => 'golden',

'leopard' => 'leopard',

'multicolor' => 'multicolor',

'white & blue' => 'White & Blue',

'white & black' => 'White & Black',

]

],

],

'my\_size' => [

'dbtype'=> 'String',

'label' =>'My Size',

'name' =>'size',

'require' => 1,

'display' =>[

'type' =>'select',

'data' =>[

'S' => 'S',

'M' => 'M',

'L' => 'L',

'XL' => 'XL',

'XXL' => 'XXL',

'XXXL' => 'XXXL',

]

],

],

],

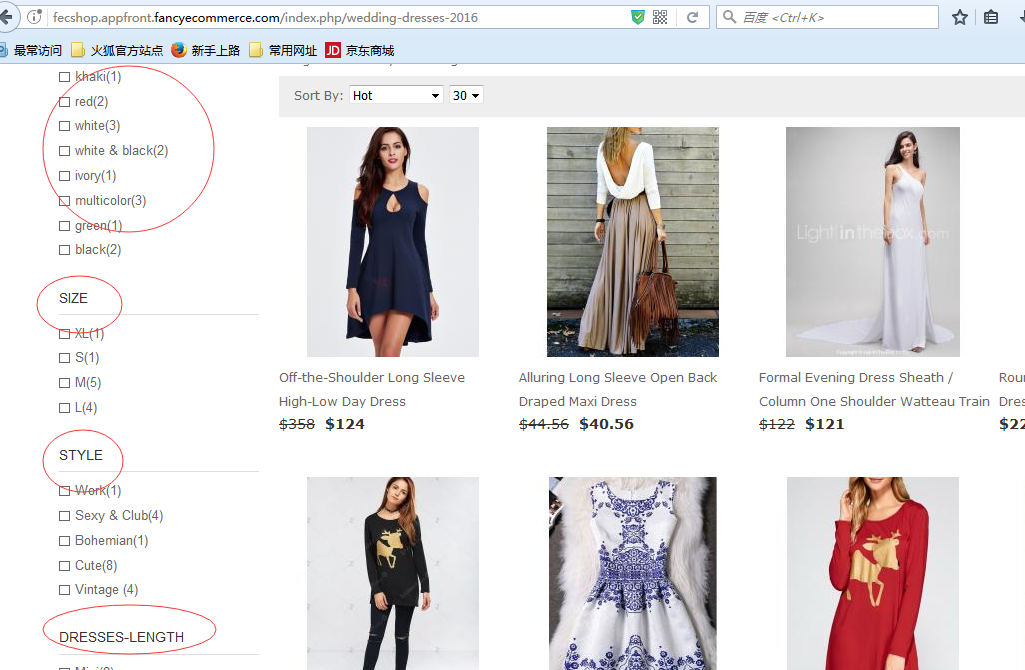
通过链接：<https://fecshop.appfront.fancyecommerce.com/reindeer-pattern-glitter-christmas-dress> 。您可以看到颜色是用图片实现的，如果您想用图片显示，那么 您需要设置'showAsImg' => true 即可。 只能有一个custom\_option属性设置为true，如果多个设置为true，只有一个 有效，其他的会被忽略。

上面是属性组里面的三种属性类型，分别是： spu\_attr general\_attr 和 custom\_options ，我们可以通过配置的方式添加多个属性组，每一个属性组包含上面 的三种属性，譬如衣服裙子一个属性组，电脑一个属性组，手机一个属性组 等等， 通过属性组配置的方式可以很方便的扩展产品独有的属性， 在配置文件中配置好产品属性组后， 在后台 编辑产品的时候选择相应的属性组即可添加相应的属性，对于属性组里面的属性，一旦编辑后 。尽量不要经常的改动。



### 分类页面产品属性过滤的设置

就是前端页面侧栏的产品过滤，如图：



如何设置呢？

1、全局设置：首先打开配置文件 @appfront/config/fecshop\_local\_modules/Catalog.php

配置

return [

'catalog' => [

'params'=> [

###############################

## category部分设置 ##

###############################

'category\_breadcrumbs' => false, # 是否显示分类的面包屑导航。

/\*\*

\* 注意：做侧栏分类产品过滤的属性，必须是select类型的，其他的类型请不要用，

\* 对于select类型，目前不支持多语言数据库存储，select类型的各个值是通过前端翻译文件来实现翻译的、

\* 对于color size 对应的保存值，只可以使用 '数字','字符','空格','&','-','\_' 这6类字符

\*/

'category\_filter\_attr' =>[

'color','size',

],

'category\_filter\_category' => true,

'category\_filter\_price' => true,

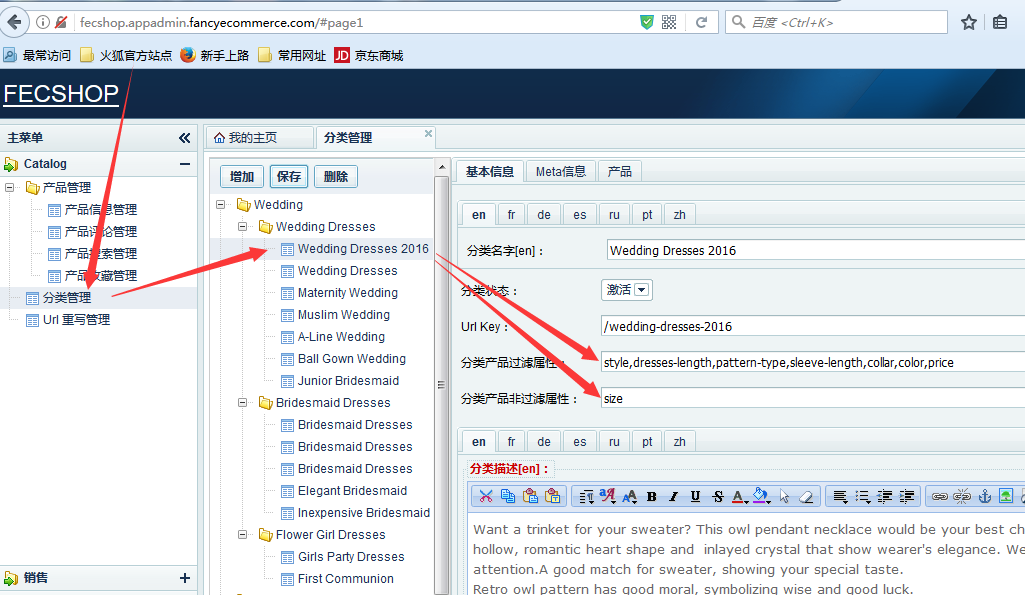
category\_filter\_attr 就是配置分类侧栏的产品属性过滤，目前设置了color和size ，这里设置后是一个公用的设置，也就是所有的分类就默认为这两个属性的过滤。

category\_filter\_category代表侧栏是否显示分类的过滤

category\_filter\_price代表侧栏是否显示价格的过滤。

2、个性化设置

在上面设置的全局设置的基础上，可以进行个性化设置，后台打开分类，如图：



**分类产品过滤属性：**是在全局设置的基础上添加一些新的产品属性，用于分类侧栏属性过滤

**分类产品非过滤属性：**是在全局设置的基础上删除一些产品属性，譬如下面的图中， 添加的是size，虽然在全局设置中添加了size，但是该分类添加了size，因此该 分类侧栏的属性过滤不会有size。

通过上面，我们可以很方便的添加产品在分类侧栏的属性过滤。

### 搜索页面产品属性过滤的设置

1.首先设置搜索语言，@common/config/fecshop\_local\_services/Search.php

1.1

return [

'search' => [

'filterAttr' => [

'color','size', # 在搜索页面侧栏的搜索过滤属性字段

],

'childService' => [

'mongoSearch' => [

'searchIndexConfig' => [ #设置用于全文搜索的产品属性以及权重，权重高的属性，排名靠前。

'name' => 10, # 产品name作为full search text的属性，权重为10

'description' => 5, # 产品description作为full search text的属性，权重为5

],

# more: https://docs.mongodb.com/manual/reference/text-search-languages/#text-search-languages

'searchLang' => [

'en' => 'english',

'fr' => 'french',

'de' => 'german',

'es' => 'spanish',

'ru' => 'russian',

'pt' => 'portuguese',

],

],

'xunSearch' => [

'fuzzy' => true, # 是否开启模糊查询

'synonyms' => true, #是否开启同义词翻译

'searchLang' => ['zh'],

],

],

]

];

1.1 filterAttr是添加在搜索页面侧栏用于过滤的产品属性。

1.2 searchIndexConfig是设置用于全文搜索的产品属性以及权重，权重高的属性，排名靠前。

1.3 searchLang是设置搜索的语言，全文搜索是需要根据语言切词的， 因此各个语言的搜索会略有差异，根据各个语言，生成对应的产品语言搜索表。

2.生成搜索表

首先需要根据上面的配置文件，设置您的所有的语言。也就是上面的searchLang。

每一个语言对应一个产品搜索表，这是因为mongodb的full text search，一个表只能有一个 语言选项，因此需要生成多个产品搜索表，生成的脚本为：

@fecshop/shell/search/fullSearchSync.sh，您进入@fecshop/shell/search下 执行sh fullSearchSync.sh即可。

### 设置产品的自定义属性

产品的自定义属性如图：



#### 1.产品的自定义有方式，京东模式和淘宝模式

1.1京东模式：类似于京东的产品，选择每一个选项后，页面会跳转到另外一个url页面， 也就是说：每一个选项就是一个产品（在数据库产品表中是一行数据）。

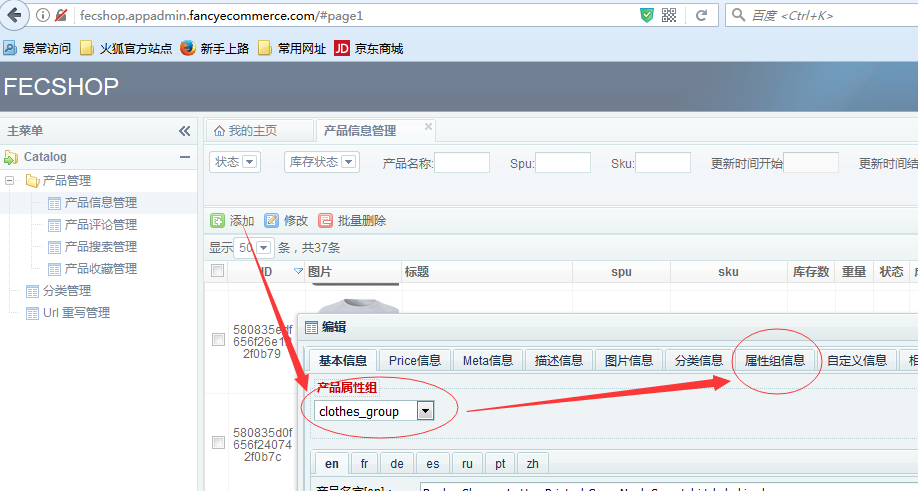
1.2淘宝模式：类似于淘宝商城的产品，选择一个选项后，产品页面不会跳转。

#### 2.下面详细说明：

2.1京东模式：这种模式，需要引入sku和spu的概念，关于这个概念在上面已经说明。 譬如页面（演示demo）：<http://fecshop.appfront.fancyecommerce.com/index.php/raglan-sleeves-letter-printed-crew-neck-sweatshirt-53386451-77774122>

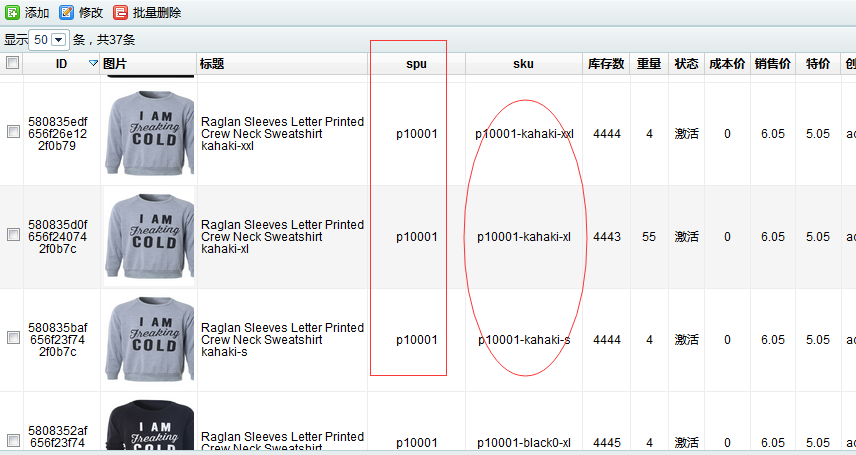
2.1.1配置：首先需要设置一个属性组，上面的内容已经说明，在配置文件中添加， 需要注意的spu属性不能超过两个（spu属性组合的复杂性）。

2.1.2新建产品，选择对应的属性组,如图



就可以查看到这个属性组中的属性，填写值即可。

填写sku和spu，需要注意的是，您新建的该款产品的spu必须相同，sku不同，一般用您的 spu+spu属性的值组合成sku，如图，我新建的spu对应的几个sku产品。



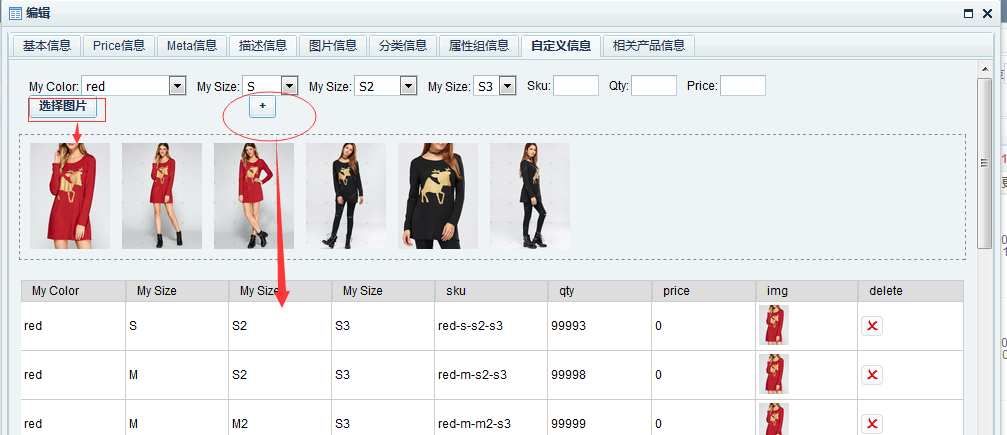
2.1.3 按照上面如图的方式配置完成后，基本就完成了配置，在前台就可以看到了。

2.2淘宝模式，演示demo：<http://fecshop.appfront.fancyecommerce.com/index.php/reindeer-pattern-glitter-christmas-dress-86519596>

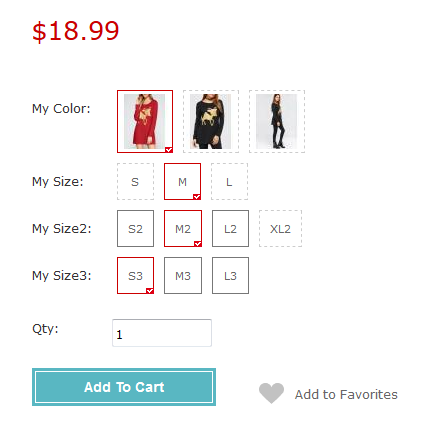
2.2.1配置：需要在属性组里面配置 custom\_options用户自定义属性，这个在上面已经 说明如何配置，您可以参考上面的说明中的文件中的例子进行修改。

2.2.2后台编辑

在属性组中添加了 custom\_options用户自定义属性，在新建产品的时候点击添加，弹出产品编辑的弹框， 选择相应的属性组，然后点击 “自定义信息”，如图：

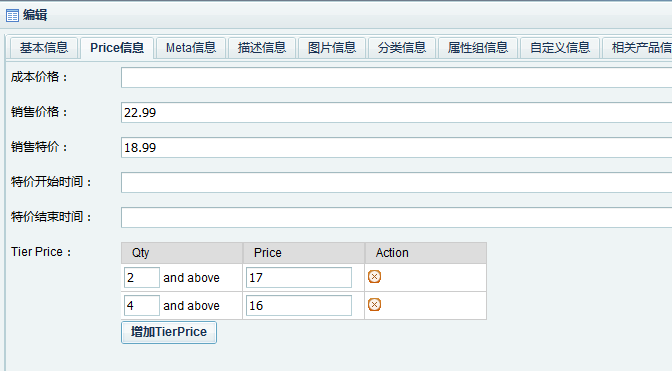


勾选，填写各个属性的值，以及选择图片，最后点击+号图标，就可以添加一行， 添加完了，保存，就可以到产品详细页面看了，譬如产品 http://fecshop.appfront.fancyecommerce.com/index.php/reindeer-pattern-glitter-christmas-dress-86519596



如果某些组合，您没有添加，就会看到某些选项是虚线标示，如果没有库存，也会虚线标示。 对于该产品，产品sku的库存会无效，产品自定义部分的库存才是相应选项的库存 ，当该产品被下单后，也是在该选项对应的qty部分减少产品的库存，而不是sku的库存（qty）

### 产品价格



成本价格：代表产品的成本，也就是采购在fecshop中，只是记录一下，没有其他用处， 当然，您可以通过脚本，生成您的出售价格。

销售价格：也就是产品在页面显示的价格

销售特价：产品的特价，当设置了产品的特价，而且特价时间没有过期，则特价是有效的， 当销售特价有效时，销售价格就会失效。

特价开始时间：产品销售特价开始的时间

特检结束时间：产品销售特价结束的时间

Tier Price： 批发价格

规定产品购买几个以上的价格。譬如购物2个以上17美元，购买4个以上16美元。

对于在产品分类页面显示的产品的几个都是一个产品的价格。产品中有一个最终价格， 是通过脚本计算出来的，这个最终价格用于在分类页面的价格排序以及价格过滤，该脚本一天跑一次。 文件为： @fecshop/shell/computeProductFinalPrice.sh， 可以在cron中配置该脚本一天跑一次。

### 产品导入和导出

目前后台还没有做这个功能。

# Fecshop 产品图片

fecshop 产品图片的实现思路，有很多参考了magento的图片生成的思路

### 1. 产品图片的存放路径

在配置文件 @fecshop/config/services/Product.php 中可以看到如下代码：

'image' => [

'class' => 'fecshop\services\product\Image',

'imageFloder' => 'media/catalog/product', # 产品图片存放路径。

//'allowImgType' => [ # 允许的图片类型

// 'image/jpeg',

// 'image/gif',

// 'image/png',

// 'image/jpg',

// 'image/pjpeg',

//],

'maxUploadMSize'=> 5, //MB # 图片最大尺寸

//'waterImg' => 'product\_water.jpg', # 水印图片

],

对于产品图片，是公用的，因此强制放到@appimage/common路径下面， 其中 imageFloder 的值代表的是相对路径，因此完整的图片存放路径为： @appimage/common/media/catalog/product 下面。

### 2. 产品图片的调用

原图的调用：

Yii::$service->image->getImgUrl('/2/01/20160905101021\_28071.jpg');

自定义尺寸的调用：

Yii::$service->product->image->getResize('/2/01/20160905101021\_28071.jpg',[230,230],false);

在自定义尺寸的调用的函数中 [230,230] 为width 和height的尺寸,详细参看函数：

/\*\*

\* $imgResize 可以为数组 [230,230] 代表生成的图片为230\*230，如果宽度或者高度不够，则会用白色填充

\* 如果 $imgResize设置为 230， 则宽度不变，高度按照原始图的比例计算出来。

\*/

protected function actionGetResize($imageVal, $imgResize, $isWatered = false)

{

$originImgPath = $this->getDir($imageVal);

if (!file\_exists($originImgPath)) {

$originImgPath = $this->getDir($this->defaultImg);

}

$waterImgPath = '';

if ($isWatered) {

$waterImgPath = $this->getDir('/'.$this->waterImg);

}

list($newPath, $newUrl) = $this->getProductNewPath($imageVal, $imgResize, $waterImgPath);

if (!file\_exists($newPath)) {

\fec\helpers\CImage::saveResizeMiddleWaterImg($originImgPath, $newPath, $imgResize, $waterImgPath);

}

return $newUrl;

}

图片按照尺寸调用的时候，php会去磁盘去查找该尺寸对应的产品图片 文件是否存在，如果不存在，php则会生成相应尺寸的图片存放到磁盘， 下次访问该图片，就直接调用生成后的图片。

上面的函数中,$isWatered代表是否打水印。

### 3.图片水印。

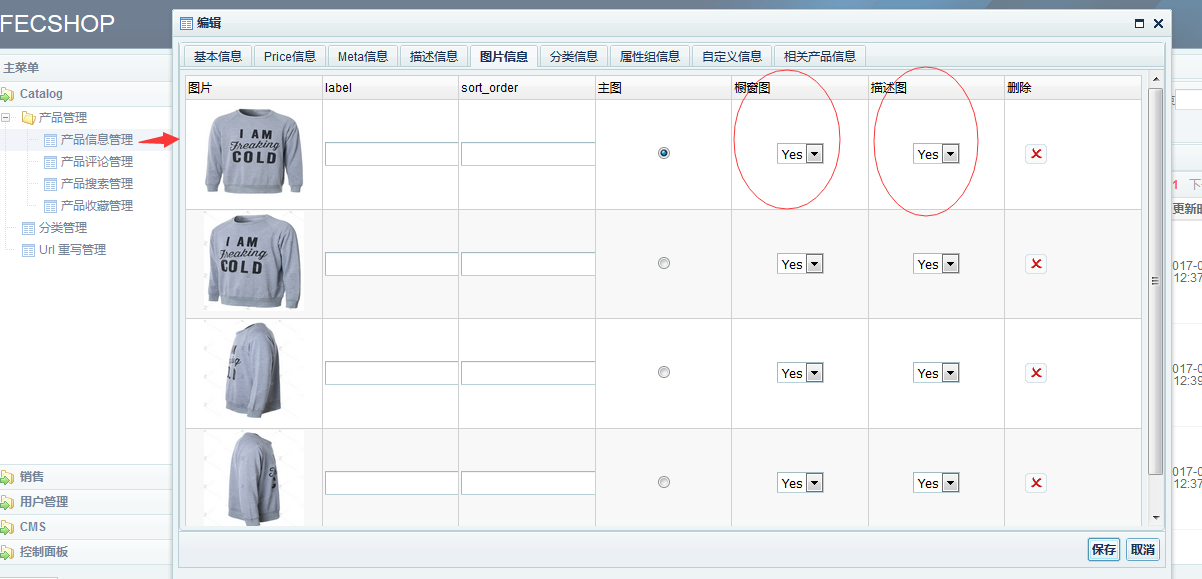
首先需要上传您的水印图。按照默认设置，水印图路径为 @appimage/common/media/catalog/product/product\_water.jpg ,您上传一张图覆盖这个水印图

然后在调用的时候， 自定义尺寸的调用：Yii::$service->product->image->getResize($product['image'],[230,230],true); ，第三个参数为true即可。

注意：目前原图是直接调用，如果想要打水印，您可以生成一个大尺寸的， 如果您不想让您的原图暴露，可以把程序里面所有的原图访问更改成 尺寸处理后的图片，然后nginx设置，封锁死原图访问url即可。

### 4.图片显示设置

在后台编辑的时候可以看到如下：

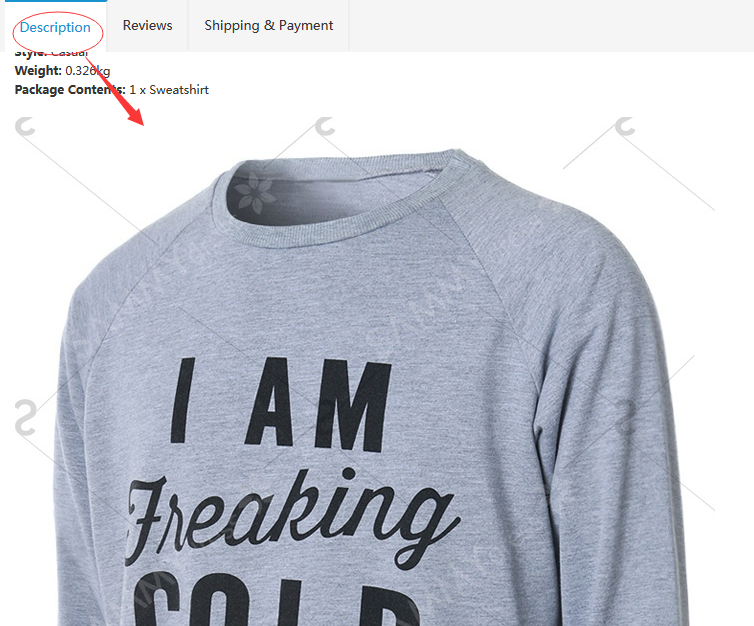


主图：代表产品的主图，在分类页面，购物车页面等，显示的都是产品的主图。

橱窗图：代表是的在产品详细页面顶部左侧显示的图片（如果是主图，无论是否设置成yes，都是橱窗图） 对于橱窗图的位置，参看图：



描述图：代表的是在产品详情页面，description描述下面显示的产品图



当一个产品设置成描述图，就会在产品描述部分下面显示出来。

# 产品库存

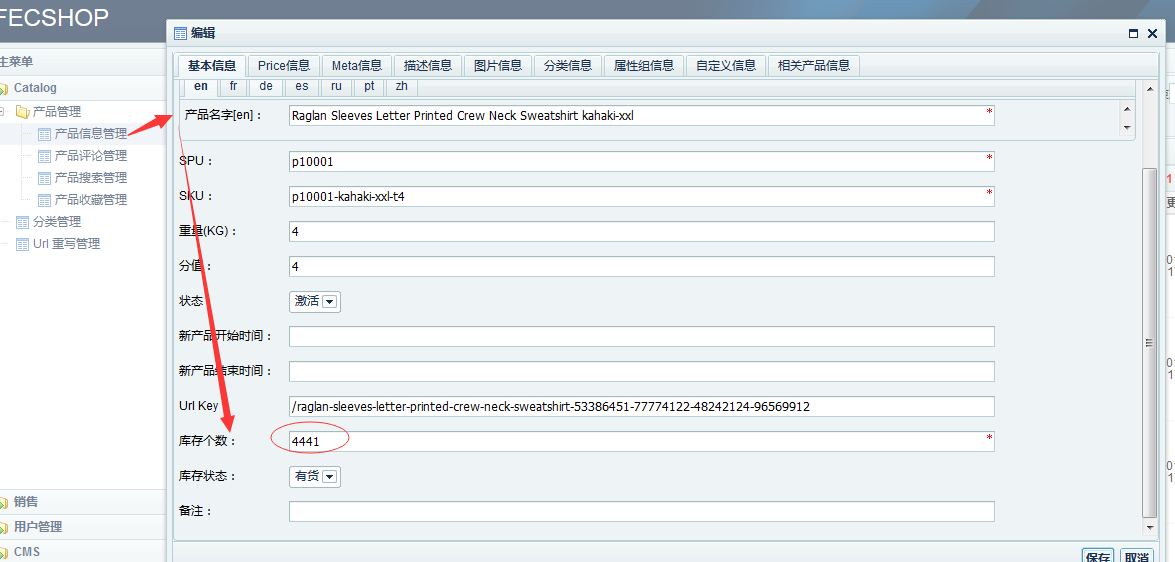
指的是产品的库存

### 1.产品库存类型

产品的类型分为两种，一种京东模式，一种淘宝模式，详情可以参看 ：[fecshop产品](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_product.html)

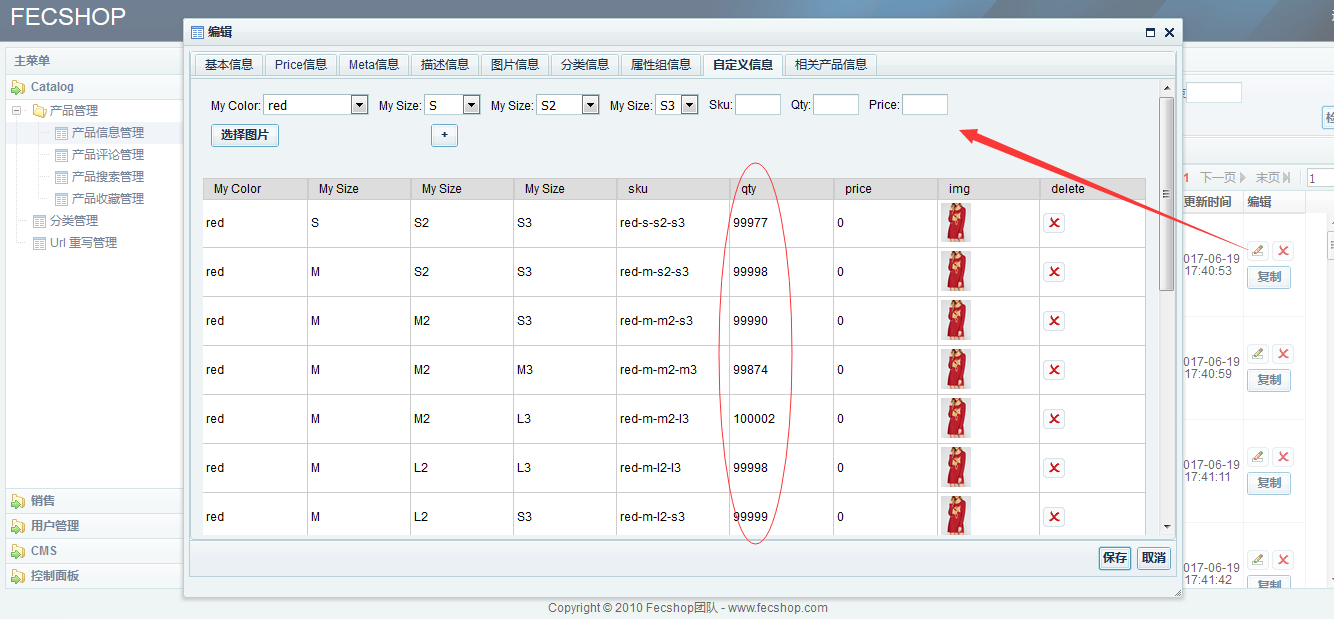
#### 京东模式

使用的是产品的qty字段，在后台编辑对应，如图：



#### 淘宝模式

对应的是custom\_option（自定义属性）部分的产品库存，在淘宝模式中， 产品的库存值（指的是上面截图的库存选项）将会被忽略，下面图中设置的库存 为有效库存。



对于淘宝和京东模式，库存方面只能二选一。

### 2.库存存储

产品表是mongodb表，产品库存已经从mongodb中抽出来，放到mysql中 ，因为产品库存操作具有多表事务性。 mongodb里面也会存储一个qty字段，但是这个存储的值无效，在 产品初始化的时候会使用mysql中的值覆盖掉。 库存在mysql中存储对应的表为：

淘宝模式库存：product\_custom\_option\_qty

京东模式库存：product\_flat\_qty

### 3.库存扣除的流程和安全性

关于支付流程，订单支付库存扣除的流程，具体流程参看[fecshop 支付](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_payment_method.html)

库存安全控制：a)高并发库存超卖控制， b)支付请求多次执行造成库存多次扣除的问题 c)未支付订单库存返还问题。

3.1 高并发库存超卖控制

为了防止超卖的控制，代码如下：@fecshop/service/product/Stock.php

// 开始扣除库存。

if (is\_array($items) && !empty($items)) {

foreach ($items as $k=>$item) {

$product\_id = $item['product\_id'];

$sale\_qty = (int)$item['qty'];

$product\_name = Yii::$service->store->getStoreAttrVal($item['product\_name'], 'name');

$custom\_option\_sku = $item['custom\_option\_sku'];

if ($product\_id && $sale\_qty) {

if(!$custom\_option\_sku){

// 应对高并发库存超卖的控制，更新后在查询产品的库存，如果库存小于则回滚。

$sql = 'update '.ProductFlatQty::tableName().' set qty = qty - :sale\_qty where product\_id = :product\_id';

$data = [

'sale\_qty' => $sale\_qty,

'product\_id'=> $product\_id,

];

$result = ProductFlatQty::getDb()->createCommand($sql,$data)->execute();

$productFlatQty = ProductFlatQty::find()->where([

'product\_id' => $product\_id

])->one();

if($productFlatQty['qty'] < 0){

Yii::$service->helper->errors->add('product: [ '.$product\_name.' ] is stock out ');

return false;

}

}else{

// 对于custom option（淘宝模式）的库存扣除

$sql = 'update '.ProductCustomOptionQty::tableName().' set qty = qty - :sale\_qty where product\_id = :product\_id and custom\_option\_sku = :custom\_option\_sku';

$data = [

'sale\_qty' => $sale\_qty,

'product\_id'=> $product\_id,

'custom\_option\_sku' => $custom\_option\_sku

];

$result = ProductCustomOptionQty::getDb()->createCommand($sql,$data)->execute();

$productCustomOptionQty = ProductCustomOptionQty::find()->where([

'product\_id' => $product\_id,

'custom\_option\_sku' => $custom\_option\_sku,

])->one();

if($productCustomOptionQty['qty'] < 0){

Yii::$service->helper->errors->add('product: [ '.$product\_name.' ] is stock out ');

return false;

}

}

}

}

return true;

}

3.2 支付请求多次执行造成库存多次扣除的问题

这个是在购物车页面点击paypal支付按钮进行支付的方式， 在跳转前，生成一个订单号和token对应，等跳转回来的时候，通过token 查找订单，然后更新订单支付，涉及到更新，并发高可能被执行多次， 因此需要进行version控制一下。

具体代码： @fecshop/service/Order.php

protected function checkOrderVersion($increment\_id){

# 更新订单版本号，防止被多次执行。

$sql = 'update '.MyOrder::tableName().' set version = version + 1 where increment\_id = :increment\_id';

$data = [

'increment\_id' => $increment\_id,

];

$result = MyOrder::getDb()->createCommand($sql,$data)->execute();

$MyOrder = MyOrder::find()->where([

'increment\_id' => $increment\_id,

])->one();

# 如果版本号不等于1，则回滚

if($MyOrder['version'] > 1){

Yii::$service->helper->errors->add('Your order has been paid');

return false;

}else if($MyOrder['version'] < 1){

Yii::$service->helper->errors->add('Your order is error');

return false;

}else{

return true;

}

}

3.3 未支付订单库存返还问题。

@fecshop/shell/order/returnPendingProductQtyStock.sh

# FecShop 产品评论

fecshop的产品评论，目前，只要是登录用户，就可以进行评论，和淘宝京东不同。前端添加 的评论，后台需要审核

是否只显示当前语言的评论？可以通过配置覆盖

@fecshop/config/services/Product.php 文件中的配置：

'review' => [

'class' => 'fecshop\services\product\Review',

'filterByLang' => false, // 是否通过语言进行评论过滤？

// true：代表用户购物过的产品才能评论，false：代表用户没有购买的产品也可以评论

'reviewOnlyOrderedProduct' => true,

// 订单创建后，多久内可以进行评论，

// 超过这个期限将不能评论产品（单位为月）, 当 reviewOnlyOrderedProduct 设置为true时有效。

'reviewMonth' => 6,

],

filterByLang: 改成true，只会显示当前语言下的评论。

reviewOnlyOrderedProduct: true代表用户购物过的产品才能评论，false代表用户没有购买的产品也可以评论

reviewMonth: 订单创建后x个月内可以进行评论，超过这个期限将不能评论产品（单位月）,当reviewOnlyOrderedProduct设置为true时有效。

# Fecshop 购物车

在产品页面中通过按钮”加入购物车”可以将产品加入购物车中，fecshop的购物车数据 是存储到数据库中的，session保存cart表的id，因此，当用户登录账号后，将产品加入购物车，换了其他的浏览器， 然后重新登录 账号后，购物车信息还是存在的。

购物车的配置信息：

@fecshop/config/services/Cart.php

<?php

return [

'cart' => [

'class' => 'fecshop\services\Cart',

# 子服务

'childService' => [

'quote' => [

'class' => 'fecshop\services\cart\Quote',

],

'quoteItem' => [

'class' => 'fecshop\services\cart\QuoteItem',

],

'info' => [

'class' => 'fecshop\services\cart\Info',

/\*\*

\* 单个sku加入购物车的最大个数。

\*/

'maxCountAddToCart' => 100,

# 上架状态产品加入购物车时，

# 如果addToCartCheckSkuQty设置为true，则需要检查产品qty是否>购买qty，

# 如果设置为false，则不需要，也就是说产品库存qty小于购买qty，也是可以加入购物车的。

'addToCartCheckSkuQty' => false,

],

'coupon' => [

'class' => 'fecshop\services\cart\Coupon',

],

],

],

];

单个sku加入购物车的最大个数：通过配置maxCountAddToCart

加入购物车是否检查库存：通过配置addToCartCheckSkuQty,一般是设置成false， 一般是在生成订单的时候进行检查，因为有一些人的订单可能未付款，超过一段时间 未付款的订单，会由后台定时脚本释放库存，因此，加入购物车的时候没有库存，可能待会就有了。

购物车数据是放到mysql的，因为涉及到多表事务。

### 购物车点击 Proceed to Pay 的设置

游客和登录用户都可以把产品加入购物车，但是，如果用户想进一步下单，我想强制 让用户先注册账户才能下单，也就是说，支持游客加入购物车，但是不允许游客下单，您可以 在文件： @fecshop/app/appfront(apphtml5 | appserver)/config/modules/Checkout.php

return [

/\*\*

\* checkout 模块的配置，您可以在@apphtml5/config/fecshop\_local\_modules/Checkout.php

\* 中进行配置，二开，或者重写该模块（在上面路径中如果文件不存在，自行新建配置文件。）

\*/

'checkout' => [

'class' => '\fecshop\app\apphtml5\modules\Checkout\Module',

/\*\*

\* 模块内部的params配置。

\*/

'params'=> [

'guestOrder' => true, // 是否支持游客下单

],

],

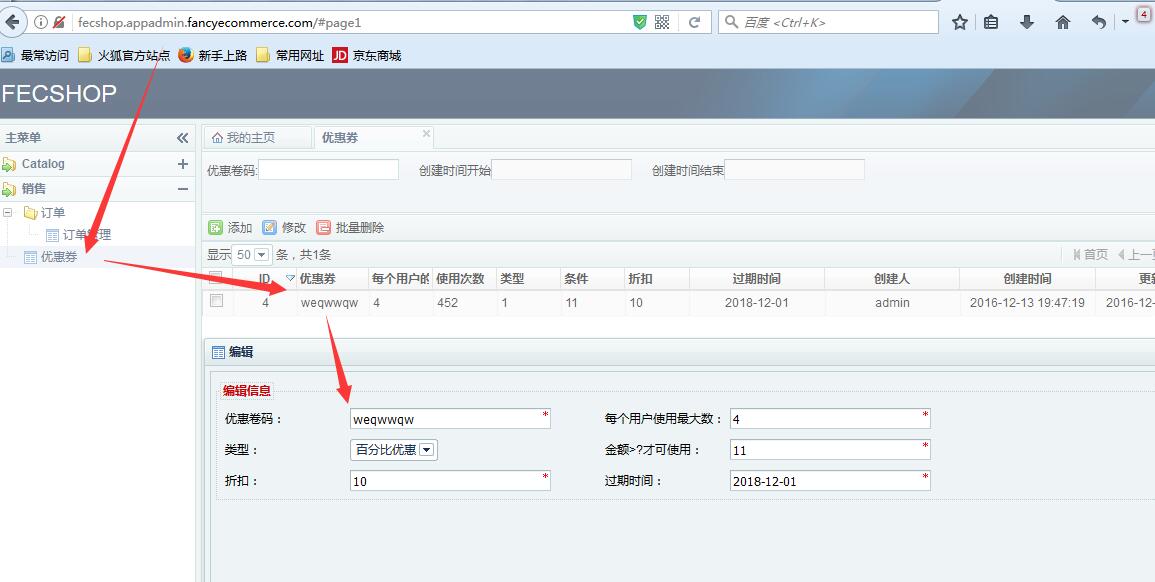
];

将guestOrder 设置成 false后，在购物车点击proceed to Pay ,就会跳转到账户登录页面

这里的设置，在入口 appfront , apphtml5, appserver 都有效

# Fecshop 优惠券

通过后台设置优惠券，用户可以使用优惠卷码进行折扣， 目前fecshop的优惠券功能比较简单。

优惠券后台的添加 

1.优惠卷码: 这个必须唯一

2.每个用户最大的使用次数：这个顾名思义，就是优惠券，每个用户最多使用多少次

3.类型：折扣的类型，是按照百分比的方式折扣，还是直接立减的方式折扣

4.金额>?才可使用：这个是使用优惠券的条件，当购物车金额满足多少的时候才可以使用优惠券

5.折扣，也就是减去的部分，如果折扣类型是百分比优惠， 那么填写10的意思是 减少产品总价的10%，如果是金额优惠，那么填写10的意思是 产品总价减少10。

6.过期时间，如果超过这个时间，优惠券则不可用。

后台设置后，就可以在前端使用了，目前优惠券没有做到和某个用户绑定，

现在的优惠券功能，更多的是为了给定量的客户发送邮件，刺激用户进行二次购买。

7.优惠券必须在用户登录状态下才能使用，在购物车页面进行使用的（必须满足优惠券 使用的条件，满足条件才可以使用）， 使用后，购物车表中会记录该优惠券，优惠券的使用次数会被+1，在购物车 中的优惠券是可以取消的， 当用户生成订单后，购物车中的信息会被清空，包括优惠券， 至此，优惠券使用完成，即使订单没有支付，优惠券也会被记录使用次数+1.

# Fecshop 货运方式

fecshop shipping method 的配置以及详细的说明

### 重量计算

1.3.2版本以后添加的体积重

1.产品的重量分为实际重量和体积重量两种。

2.计算运费的重量 = max(实际重量, 体积重量)，取两种的最大的哪一个作为产品最终重量 ，详细参看：<http://www.fecshop.com/topic/659>

3.体积重 = (长(c㎡) \* 宽(c㎡) \* 高(c㎡) ) / 换算系数

3.1fecshop默认的换算系数:采用 目前的国际标准值5000，如果您想要自己定义，在@common/config/fecshop\_local\_services/Shipping.php 中更改配置 volumeWeightCoefficient的值

3.2后台产品编辑功能，加入了长, 宽, 高, 体积重 4个属性，当您填写长, 宽, 高这三个属性后， 产品的体积重，将由上面的公式计算得到，填写到产品表中。

3.3购物车页面，显示产品的体积信息，重量信息，如果您不想显示，可以在模板中手动去除。

### Shipping Method 配置

货运方式的配置文件在 @fecshop/common/config/fecshop\_local\_services/Shipping.php

内容如下：

return [

'shipping' => [

// Shipping的运费，是表格的形式录入，shippingCsvDir是存放运费表格的文件路径。

'shippingCsvDir' => '@common/config/shipping',

'shippingConfig' => [

'free\_shipping'=> [ // 免运费

'label'=> 'Free shipping( 7-20 work days)',

'name' => 'HKBRAM',

'formula' => '0', // 这里填写公式，该值代表免邮。

// 国家限制，当国家限制不满足，则该物流不可用 （如果没有国家限制可以去掉）

'country' => [ // 这里填写(允许|不允许)使用的国家简码，如果您没有这方面的约束，请去掉，去掉后代表没有任何约束

'type' => 'allow', // allow代表只允许下面的国家使用该shipping，not\_allow代表不允许下面国家使用该shipping

'code' => [

'CN',

'US',

]

],

// 重量限制，当重量超出这个范围，该物流将不可用 （如果没有重量限制可以去掉）

'weight' => [

'min' => 0,

'max' => 100,

],

],

'middle\_shipping'=> [ // xxx shipping

'label'=> 'middle shipping( 6-15 work days)',

'name' => 'HKBRAM',

'formula' => '[weight] \* 0.5', // 这里填写公式

// 对于国家和重量限制，如果没有，则不用填写，如果有，参考上面的样式填写

],

'fast\_shipping'=> [

'label'=> 'Fast Shipping( 5-10 work days)',

'name' => 'HKDHL',

'formula' => 'csv', // 请将文件名字的命名写入，譬如： fast\_shipping.csv

'csv\_content' => '', // 这个由shipping动态从文件中获取内容

// 对于国家和重量限制，如果没有，则不用填写，如果有，参考上面的样式填写

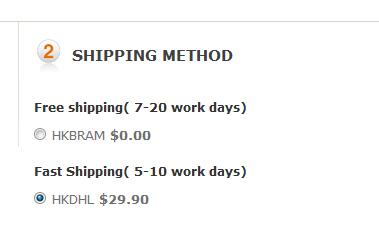
],

],

],

];

前端显示：



### 1.配置运费

在上面配置中，formula 代表运费计算数学公式。

1.1 formula == 'csv'

如果是该方式， 代表以csv文件的方式配置物流运费，运费的计算就依赖于csv表格的配置， shippingCsvDir配置了表格csv文件存放的路径@common/config/shipping。对于运费方式fast\_shipping cost填写的是csv，那么他的文件路径为@common/config/shipping/fast\_shipping.csv, 也就是fast\_shipping与.csv拼成文件名字，对于csv方式的详细，参看下面第4部分

1.2 formula 值为其他的时候，则代表数学公式

根据字符串数学公式，计算出来运费

### 2.对国家的限制

2.1通用限制，譬如：

'country' => [ // 这里填写(允许|不允许)使用的国家简码，如果您没有这方面的约束，请去掉，去掉后代表没有任何约束

'type' => 'allow', // allow代表只允许下面的国家使用该shipping，not\_allow代表不允许下面国家使用该shipping

'code' => [

'CN',

'US',

]

],

type ： 类型，可以填写 allow 和 not\_allow ，只能填写其中的一个值 ，详细参看上面的注释

2.2csv类型独有限制

对于csv类型的shipping，对于表格里面没有配置的国家和省市， 该shipping method不可用， 在csv表格中 \*,代表所有，详细查看第四部分， 因此，对于csv类型，除了上面的限制外，还可以在csv表格内部做限制， 也就是在csv表格中，把 \* \*部分的去掉，只填写可用的国家和省市， 那么只有填写的这些国家省市有效，其他的国家将会导致该货运方式不可用

### 3.重量限制

'weight' => [

'min' => 0,

'max' => 100,

],

3.1满足上面的重量范围时，则该货运方式有效。

3.2如果不加入这部分配置，则没有重量限制。

### 4.关于csv运费配置文件的说明：

shippingCsvDir: Shipping的运费，是表格的形式录入，shippingCsvDir是存放运费表格的文件路径。 对于运费方式fast\_shipping，打开表格@common/config/shipping/fast\_shipping.csv，你会发现下面的数据

Country,Region/State,"Zip/Postal Code","Weight (and above)","Shipping Price"

\*,\*,\*,0.0000,29.9000

\*,\*,\*,0.5100,31.9000

\*,\*,\*,1.0100,33.9000

\*,\*,\*,1.5100,35.9000

AU,\*,\*,0.0000,19.9000

AU,\*,\*,0.5100,22.9000

AU,\*,\*,1.0100,25.9000

AU,\*,\*,1.5100,28.9000

AU,\*,\*,2.0100,31.9000

DE,\*,\*,0.0000,119.9000

DE,\*,\*,0.5100,122.9000

DE,\*,\*,1.0100,125.9000

DE,\*,\*,1.5100,128.9000

DE,\*,\*,2.0100,131.9000

\*：代表所有的意思，

譬如国家（country）下面是，代表所有的国家，第二个代表该国家 对应所有的省市，第三个代表所有的zip编号，通过 ，可以先做一个通用的配置， 然后对某些国家做独有的运费设置，如果不这样，世界上几百个国家设置运费列表，麻烦死了， 只设置某些订单量比较大的国家即可。

如果不设置 \*,那么，只有相应的国家和省市有效，其他国家该shipping method不可用。

第四列为重量，第五列为运费。

譬如第1行的意思为0.0 - 0.51kg之间的重量，发送到任意 国家的运费为29.9（基础货币值）。

第2行的意思为0.5100 - 1.0100kg之间的重量，发送到任意 国家的运费为31.9000（基础货币值）。

通过\*将所有的国家进行了配置，然后，下面就是对某些国家的配置 ，譬如第五行，是对AU国家进行的单独配置。

后面就是对DE（德国）国家进行的配置。

通过上面您应该就明白了运费的配置。

# Fecshop 游客

Fecshop 游客 指的是未登录账户的用户部分。

1. 游客下单

购物车跳转到下单页面，判断用户是否登录，如果没有登录， 则跳转到登录页面的设置：

配置文件为：

./vendor/fancyecommerce/fecshop/app/appfront/config/modules/Checkout.php

./vendor/fancyecommerce/fecshop/app/apphtml5/config/modules/Checkout.php

./vendor/fancyecommerce/fecshop/app/appserver/config/modules/Checkout.php

/\*\*

\* 模块内部的params配置。

\*/

'params'=> [

'guestOrder' => true, // 是否支持游客下单

],

默认支持游客下单，如果想必须登录后才能下单， 那么把true改成false 即可。

您可以在本地路径更改配置，来覆盖掉Fecshop的默认配置

# Fecshop Facebook Google登录

Facebook Google 登录，指的是用户可以直接使用Facebook Google 账号登录 fecshop，而不需要注册账户

### 1、配置

在store的配置文件：@app/config/fecshop\_local\_services/Store.php 中可以看到，每一个store都可以独立配置相应的facebook 和google登录，配置代码如下：

// 第三方账号登录配置

'thirdLogin' => [

// facebook账号登录

'facebook' => [ //fb api配置 ，fb可以一个app设置pc和手机web两个域名

'facebook\_app\_id' => '108618299786621',

'facebook\_app\_secret' => '420b56da4f4664a4d1065a1d31e5ec73',

],

// google账号登录

'google' => [ //谷歌api visit https://code.google.com/apis/console to generate your google api

'CLIENT\_ID' => '380372364773-qdj1seag9bh2n0pgrhcv2r5uoc58ltp3.apps.googleusercontent.com',

'CLIENT\_SECRET' => 'ei8RaoCDoAlIeh1nHYm0rrwO',

],

],

上面的app\_id 和 secret是google 和facebook授权的， 这些信息，您需要到官方登录得到授权信息。

[facebook login 申请 app\_id 和 app\_secret](http://www.fecshop.com/topic/164)

[google login api 申请 CLIENT\_SECRET 和 CLIENT\_SECRET](http://blog.csdn.net/terry_water/article/details/55095209)

按照上面的文章的步骤设置好，然后将授权信息填写到相应的store中即可。

注意：每一个store都需要填写，除非你不想该store支持Facebook Google 登录

# Fecshop 国家

fecshop 国家的配置

国家的设置，对于下拉条选择的方式，目前只支持国家，省/直辖市 ， 还是无法到city的，

打开文件：@fecshop/services/helper/Country.php

国家一般是不会变动的，如果您想重写可以重写函数 getAllCountryArray()

对于省，不是每一个国家都编辑了的，只有部分国家的省进行了编辑 打开上面的文件查看函数：getCountryStateArr()， 譬如美国的省的配置：

US是上面 getAllCountryArray()函数返回数组中的key，也就是国家简码

AL 是Alabama这个省（州）的简码，Alabama是省的名字， 对于省的简码，可以随便写，只要在数组中唯一就行了。

'US' => [

'AL' => 'Alabama',

'AK' => 'Alaska',

'AS' => 'American Samoa',

'AZ' => 'Arizona',

'AR' => 'Arkansas',

'AF' => 'Armed Forces Africa',

'AA' => 'Armed Forces Americas',

'AC' => 'Armed Forces Canada',

'AE' => 'Armed Forces Europe',

'AM' => 'Armed Forces Middle East',

'AP' => 'Armed Forces Pacific',

'CA' => 'California',

'CO' => 'Colorado',

'CT' => 'Connecticut',

'DE' => 'Delaware',

'DC' => 'District of Columbia',

'FM' => 'Federated States Of Micronesia',

'FL' => 'Florida',

'GA' => 'Georgia',

'GU' => 'Guam',

'HI' => 'Hawaii',

'ID' => 'Idaho',

'IL' => 'Illinois',

'IN' => 'Indiana',

'IA' => 'Iowa',

'KS' => 'Kansas',

'KY' => 'Kentucky',

'LA' => 'Louisiana',

'ME' => 'Maine',

'MH' => 'Marshall Islands',

'MD' => 'Maryland',

'MA' => 'Massachusetts',

'MI' => 'Michigan',

'MN' => 'Minnesota',

'MS' => 'Mississippi',

'MO' => 'Missouri',

'MT' => 'Montana',

'NE' => 'Nebraska',

'NV' => 'Nevada',

'NH' => 'New Hampshire',

'NJ' => 'New Jersey',

'NM' => 'New Mexico',

'NY' => 'New York',

'NC' => 'North Carolina',

'ND' => 'North Dakota',

'MP' => 'Northern Mariana Islands',

'OH' => 'Ohio',

'OK' => 'Oklahoma',

'OR' => 'Oregon',

'PW' => 'Palau',

'PA' => 'Pennsylvania',

'PR' => 'Puerto Rico',

'RI' => 'Rhode Island',

'SC' => 'South Carolina',

'SD' => 'South Dakota',

'TN' => 'Tennessee',

'TX' => 'Texas',

'UT' => 'Utah',

'VT' => 'Vermont',

'VI' => 'Virgin Islands',

'VA' => 'Virginia',

'WA' => 'Washington',

'WV' => 'West Virginia',

'WI' => 'Wisconsin',

'WY' => 'Wyoming',

],

譬如中国的配置：

'CN' => [

'BJ' => '北京市',

'SH' => '上海市',

'TJ' => '天津市',

'CQ' => '重庆市',

'HEB' => '河北省',

'SAX' => '山西省',

'LN' => '辽宁省',

'JL' => '吉林省',

'HLJ' => '黑龙江省',

'JS' => '江苏省',

'ZJ' => '浙江省',

'AH' => '安徽省',

'FJ' => '福建省',

'JX' => '江西省',

'SD' => '山东省',

'HEN' => '河南省',

'HUB' => '湖北省',

'HUN' => '湖南省',

'GD' => '广东省',

'HN' => '海南省',

'SC' => '四川省',

'HZ' => '贵州省',

'YN' => '云南省',

'SNX' => '陕西省',

'GS' => '甘肃省',

'QH' => '青海省',

'TW' => '台湾省',

'GX' => '广西壮族自治区',

'NMG' => '内蒙古自治区',

'XZ' => '西藏自治区',

'NX' => '宁夏回族自治区',

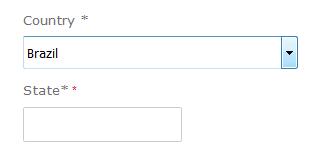
'XJ' => '新疆维吾尔自治区',

'XG' => '香港特别行政区',

],

如果您专做的国家市场，在这里没有省市，您可以重写这个函数，然后 把相应的省市信息添加到数组中。

如果在 getCountryStateArr() 返回的数组中不存在的国家， 在下单页面，就需要用户在输入框中填写国家，



如果在getCountryStateArr()返回的数组中存在相应的国家和省的数据 那么在下单页面，省部分是下拉条选择。



您可以重写 getCountryStateArr()方法，把您想要专做市场的国家对应的省 填写进去

# Fecshop 支付

对支付平台的配置信息。

### Payment支付配置文件

Payment支付配置文件：@common/config/fecshop\_local\_services/Payment.php

详细的代码可以参看：<https://github.com/fecshop/yii2_fecshop_app_advanced/blob/master/common/config/fecshop_local_services/Payment.php>

### 设置https方式

参看文章：[以https方式的paypal支付](http://www.fecshop.com/topic/381)

### 设置paypal支付

注意：paypal支付部分一定要看完文档

1.配置

因为paypal有两种，购物车点击支付按钮的快捷支付，和标准支付，因此 在两个地方进行设置

'paypal\_standard' => [

// 订单生成后，跳转到支付开始页面的url

'start\_url' => '@homeUrl/payment/paypal/standard/start',

// 下面是沙盒地址，线上地址为：https://api-3t.paypal.com/nvp，

// 这个url的作用用于 Yii::$service->payment->paypal->PPHttpPost5 ，发起一些api请求

// 譬如获取token，获取paypal存储的address（购物车快捷支付），发起扣款请求

'nvp\_url' => 'https://api-3t.sandbox.paypal.com/nvp',

// 下面是沙盒地址，线上地址为：https://www.paypal.com/cgi-bin/webscr

// 获取token后，通过这个url跳转到paypal的url地址，另外paypal的IPN消息的合法性认证，也是使用的这个url

// 也就是 Yii::$service->payment->paypal->getVerifyUrl()

'webscr\_url' => 'https://www.sandbox.paypal.com/cgi-bin/webscr',

'account' => 'zqy234api1-facilitator\_api1.126.com',

'password' => 'HF4TNTTXUD6YQREH',

'signature'=> 'An5ns1Kso7MWUdW4ErQKJJJ4qi4-ANB-xrkMmTHpTszFaUx2v4EHqknV',

'label'=> 'PayPal Express Payments',

// 跳转到paypal确认后，跳转到fecshop的url

'return\_url' => '@homeUrl/payment/paypal/standard/review',

// 取消支付后，返回fecshop的url

'cancel\_url' => '@homeUrl/payment/paypal/standard/cancel',

// 支付成功后，fecshop跳转的url

'success\_redirect\_url' => '@homeUrl/payment/success',

// paypal发送IPN，fecshop用于接收IPN消息的地址。

'ipn\_url' => '@homeUrl/payment/paypal/standard/ipn',

],

和

'paypal\_express' => [

// 下面是沙盒地址，线上地址为：https://api-3t.paypal.com/nvp，

// 这个url的作用用于 Yii::$service->payment->paypal->PPHttpPost5 ，发起一些api请求

// 譬如获取token，获取paypal存储的address（购物车快捷支付），发起扣款请求

'nvp\_url' => 'https://api-3t.sandbox.paypal.com/nvp',

// 下面是沙盒地址，线上地址为：https://www.paypal.com/cgi-bin/webscr

// 获取token后，通过这个url跳转到paypal的url地址，另外paypal的IPN消息的合法性认证，也是使用的这个url

// 也就是 Yii::$service->payment->paypal->getVerifyUrl()

'webscr\_url' => 'https://www.sandbox.paypal.com/cgi-bin/webscr',

'account' => 'zqy234api1-facilitator\_api1.126.com',

'password' => 'HF4TNTTXUD6YQREH',

'signature'=> 'An5ns1Kso7MWUdW4ErQKJJJ4qi4-ANB-xrkMmTHpTszFaUx2v4EHqknV',

'label'=> 'PayPal Express Payments',

// 跳转到paypal确认后，跳转到fecshop的url

'return\_url' => '@homeUrl/payment/paypal/express/review',

// 取消支付后，返回fecshop的url

'cancel\_url' => '@homeUrl/payment/paypal/express/cancel',

// 支付成功后，fecshop跳转的url

'success\_redirect\_url' => '@homeUrl/payment/success',

// paypal发送IPN，fecshop用于接收IPN消息的地址。

'ipn\_url' => '@homeUrl/payment/paypal/express/ipn',

],

**更改线上地址**：您需要将上面的nvp\_url 和api\_url的url中的sanbox去掉，，譬如 'nvp\_url' => 'https://api-3t.sandbox.paypal.com/nvp', 改成 'nvp\_url' => 'https://api-3t.paypal.com/nvp',

**更改线上的账户**：在paypal账户获取 account password signature ,注意 account password 是您的paypal的api申请到的，而不是登录paypal官网的账户密码， 具体的申请步骤参看文章： <http://www.fecshop.com/topic/374> ，申请到了填写上去即可，注意上面配置里面的paypal\_express和paypal\_standard 都要填写

2.**登录paypal，开启IPN**

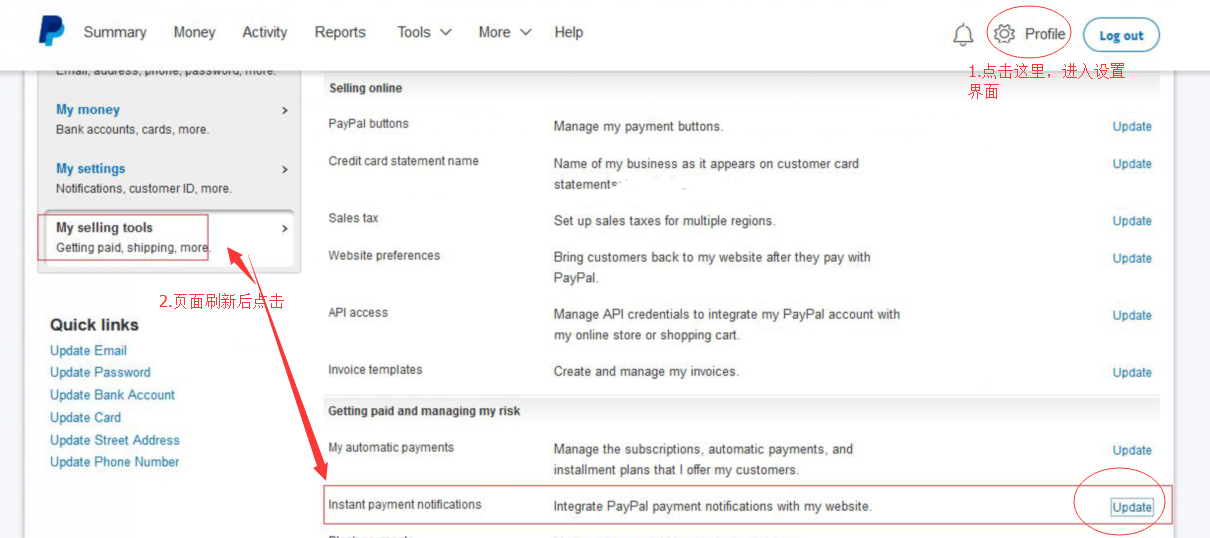
一定要登录paypal官网 www.paypal.com，设置开启IPN，否则将无法接收paypal的IPN消息。

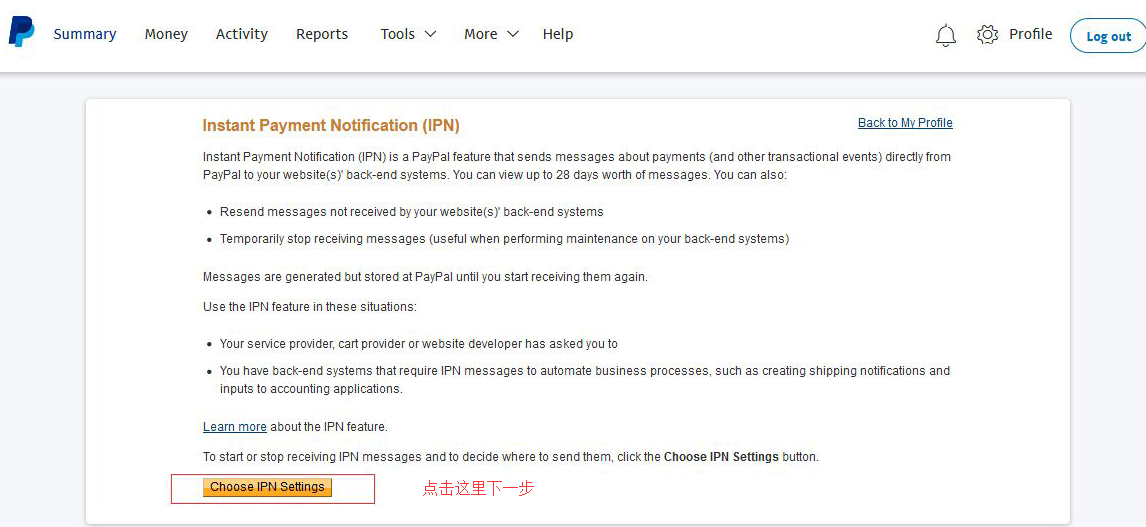
**一定要登录paypal官网 www.paypal.com，设置开启IPN，否则将无法接收paypal的IPN消息。**

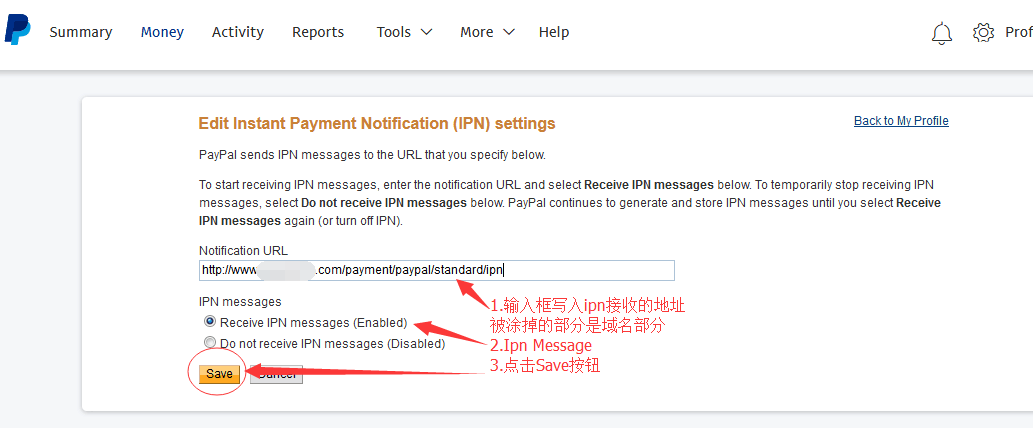
**一定要登录paypal官网 www.paypal.com，设置开启IPN，否则将无法接收paypal的IPN消息。**

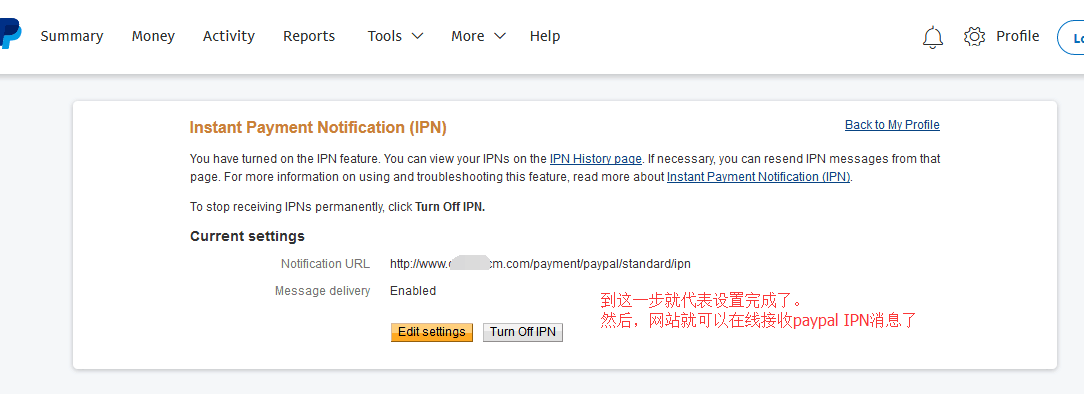
**一定要登录paypal官网 www.paypal.com，设置开启IPN，否则将无法接收paypal的IPN消息。**

重要的事情说三遍，下面是详细的设置步骤。









设置完成后，就可以接收IPN消息了。

为什么要设置接收IPN呢？可以参看一下这个文章：<http://www.fecshop.com/topic/457>

### 设置微信支付

因为微信库包sdk的配置是const常量，因此比较纠结，最后 把配置文件放到了 @common/config/payment/wxpay/lib/WxPay.Config.php文件中 ,更改里面的配置即可。

目前微信支付只支持pc端，也就是appfront端， 因为微信内部打开的网站需要公众号支付，我没有这个玩意（微信比较恶心， 没有沙盒环境，所以我没法做）， 微信外部的浏览器打开的网站支付，需要用html5支付的方式， 这种支付方式又很难申请，另外，这两种支付不相通， 公众号支付无法在外部浏览器使用，html5支付无法在微信内部打开的网页使用， 很无语。

### 支付宝支付

@common/config/fecshop\_local\_services/Payment.php

'alipay' => [

// 沙盒环境在：https://openhome.alipay.com/platform/appDaily.htm 这里获取。

// 沙盒环境 - 商家appId 在这里获取 https://openhome.alipay.com/platform/appDaily.htm?tab=info

'appId' => '2016080500172713',

// 沙盒环境 - 商家UID ： https://openhome.alipay.com/platform/appDaily.htm?tab=account

'sellerId' => '2088102170055546',

// 应用私钥，可以在这里通过工具生成：https://docs.open.alipay.com/291/105971/

'rsaPrivateKey' => '',

// 支付宝公钥，注意，这里不是应用私钥，需要把应用公钥提交后获取的支付宝公钥

// 对于沙盒账户的步骤可以参看：http://blog.csdn.net/terry\_water/article/details/75258175

'alipayrsaPublicKey' => 'MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAt5egD1BQCK5fCQXKsgWh+VFj9zanV9cdwVmM/MOQ/zrwMBHMIRO0IdJMft351iXtyACKVX+noK1qzkiVOdg3MxLjbGoMDKR+/1PDxoxtWSVUJBywoYHH/Dh7TCi5GWGasOlXV4qWi0e5Yfa2x/Wi0cxqx76aY5izXEyabHAvWgTWNv121ZRNhl4qcuoWZYiMIQpTst6hEhRn/isUMgdtLRQ1a06q+qOkLmJ99vq8cqbfduAdOuhzbZNWqLV76CSc0meurlVtDoIn5kVAZdzjNTA2rlqSCgs/OZxaL8s/qrIynhLoB6U6i0fj4RsIsbrvoSnrPWo98rsM0RrlU8fpdwIDAQAB', //'MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEApIw+Hsk65Z+mieDsEiTkhtf7ZNBgks83DLUDb1yh2d/HDB0s9zHFzsgQGny0kUTM0fJ43h7WydyUG9Kuv4fxD5iVfM2xkUYW5bvfTXVaj5LLj8rTKL+nnFybzzM5rewqh2u1Gzd7BbpOnhMn4Y+7JyyaWXsnRFBxIrmRAqQJVlVUG4RclLHfplFkMVcEMzoRda2UV54oQDMg8ZxignCqxgIKr7bpwpgdpdqZArHtmyEjhQfIblCLDjVk0rKxGsaz+ATYVt3eQozdyNEuKFRhy0VGmwmdQYhQFbge7SS6bVqXZHsq2fNZ6hMJ2XNOZajFm5jXMksnaX85PzdJ58HFewIDAQAB',

// 下面是沙盒地址， 正式环境请改为：https://openapi.alipay.com/gateway.do

'gatewayUrl' => 'https://openapi.alipaydev.com/gateway.do',

],

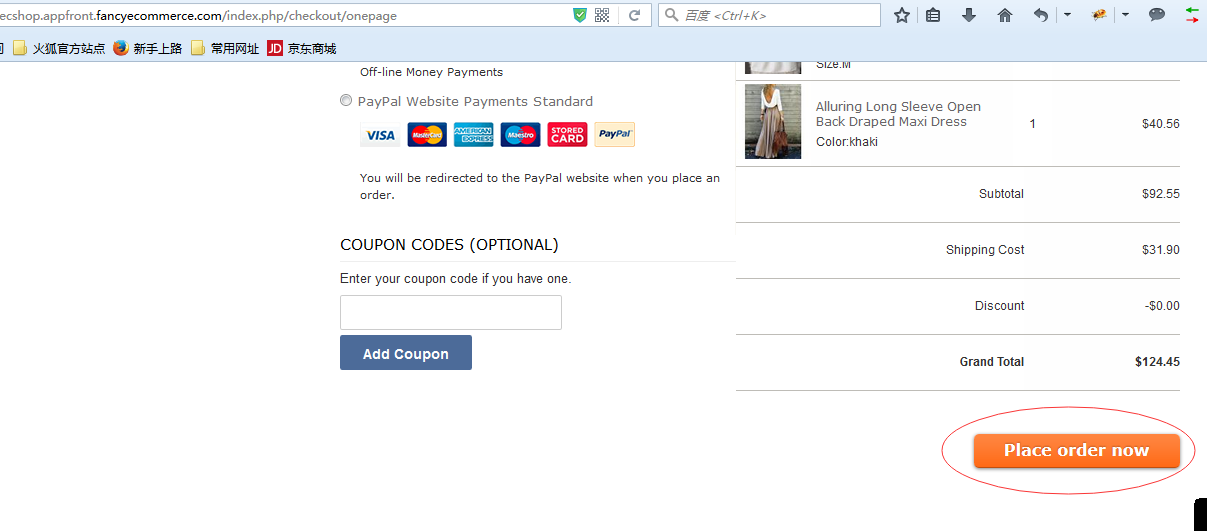
设置上面的相应信息即可。每一行都有注释，详细看即可。

### 开发新支付 - 配置

当您想要开发一个新的支付方式的时候，您需要做一个跳转到第三方支付平台的准备页面（payment start url） 和一个支付成功返回的页面url。

下面是详细的步骤，我这里也有一个支付宝开发的详细步骤，供参考，地址为：<http://www.fecshop.com/topic/143>

#### 1.1 支付跳转前的工作



在支付页面，填写好支付信息后，点击支付按钮，fecshop会先进行一系列的处理， 最终生成订单，将订单编号保存到session中，然后跳转到当前支付配置对应的 start\_url,譬如上面paypal的start\_url为@homeUrl/payment/paypal/standard/start（@homeUrl是首页url），在这个url中，需要将订单中的字段值取出，组合成支付平台想要的 数据格式，发送给第三方支付平台，并进行跳转到第三方支付平台。

#### 1.2 支付成功跳转后的工作

支付成功后，用户通过点击，或者由第三方支付平台自动，跳转到网站中，也就是上面配置的 success\_redirect\_url,该url对应的内容显示用户支付成功后的信息

#### 1.3 在第三方支付平台取消订单

如果用户在支付平台点击取消订单，那么就会跳转到fecshop的取消订单的链接， 也就是cancel\_url对应的链接

#### 1.4 IPN消息url

当用户支付成功后，支付平台会给fecshop发送一个支付成功的消息，fecshop接收到消息 后会把接收到的参数传递给paypal并进行询问是否是paypal发送的，当paypal反馈是， fecshop会把订单支付状态改成processing

### 开发新支付 - 功能实现

上面填写的4个url，您可以扩展一个新模块实现url对应的功能即可。 二开支付，知道原理后，还是蛮容易的。

paypal支付详细说明：

paypal有两种支付方式，一种是购物车页面点击支付按钮的方式，一种是 填写完成货运地址等信息后，在支付的方式。

paypal 购物车按钮流程

1.购物车点击paypal支付按钮，进入start部分：

1.1 检查产品库存，如果库存不足，则返回购物车页面

1.2 生成订单，订单只有order\_id,increment\_id,payment\_token几个字段填写值，其他为空

1.2 获取token，跳转到paypal express部分

2.进入paypal网站，确认后返回网站

2.1 用户登录后点击continue，跳回网站return url，在该页面api获取用户paypal保存的货运地址，自动填写货运地址

3.提交

3.1 用户修改货运地址等编辑信息。

3.2 用户点击提交按钮，发起支付请求。

3.3 服务器端检验用户提交的货运地址等信息是否正确

3.4 如果用户使用的是新地址，则插入用户的货运地址。

3.5 生成订单，不清空购物车，但是扣除库存。（生成订单的时候，是通过1.1的token进行查询出来的，而不是new的新的）

3.6 如果api请求发生报错，那么返回到return页面（譬如货运地址填写错误等），显示报错信息，让用户重新编辑信息提交，重新发起支付请求。

3.7 如果发生报错多次，token就会失效，paypal会拒绝请求。返回一个token超出最大使用次数。

3.8 api发起支付请求，支付成功后，把支付信息写入到订单中，清空购物车。

3.9 最后跳转到支付成功页面。支付成功页面一般加入以下监控的js，接收支付成功信息。

注意：ipn部分目前没有做任何处理，只是打印了一下log， 如果用户在paypal提出申诉，要求退款等，paypal会给网站发送一个消息， 网站这边不会同步订单状态为“取消”，而是让erp那边做处理。

paypal 填写货运地址在点击支付的流程。

1. 填写表单：

1.1 用户在购物车页面填写地址信息，货运方式，支付方式

1.2 用户提交

1.3 服务端接收信息生成订单，生成订单扣除产品库存，但是不清空购物车，

然后根据配置的startUrl，跳转到相应的链接

1.4 生成订单的时候，如果存在错误，跳转到checkout/onepage页面，显示报错信息

1. Start部分：

2.1 提交相应订单信息获取token

2.2 更新token到订单中

2.3 跳转到paypal支付网站，登录后，点击pay now返回网站（购物车点击过去，显示的按钮是continue）

1. return部分：

3.1 通过返回的token，查询订单，得到订单信息

3.2 根据订单信息，拼装成api url，api发送下单请求。

3.3 如果下单失败，跳转到checkout/onepage页面，显示报错信息

3.4 如果下单成功，清空购物车

3.5 跳转到支付成功页面。

如果您想开发一个新支付，可以参考下支付宝支付的整个步骤： [Fecshop 支付宝支付开发思路 和 详细的文件结构](http://www.fecshop.com/topic/143)   
，这个文章写得比较详细，可以了解整个流程。

# Fecshop 订单

fecshop 订单指的是在fecshop下单后的订单信息。

### 订单配置

#### 配置文件

为：@common/config/fecshop\_local\_services/Order.php，详细如下：

<?php

/\*\*

\* FecShop file.

\*

\* @link http://www.fecshop.com/

\*

\* @copyright Copyright (c) 2016 FecShop Software LLC

\* @license http://www.fecshop.com/license/

\*/

return [

'order' => [

'increment\_id' => '1100000000', // 订单的格式。

'requiredAddressAttr' => [ // 必填的订单字段。

'first\_name',

'last\_name',

'email',

'telephone',

'street1',

'country',

'city',

'state',

'zip',

],

//处理多少分钟后，支付状态为pending的订单，归还库存。

'minuteBeforeThatReturnPendingStock' => 600,

// 脚本一次性处理多少个pending订单。

'orderCountThatReturnPendingStock' => 30,

// 订单状态配置

// !!!!!!下面是订单支付状态!!!!!!! ,各个订单状态查看相应注释

// 等待付款状态

'payment\_status\_pending' => 'payment\_pending',

// 付款处理中，(支付处理中，因为信用卡有预售，因此需要等IPN消息来确认是否支付成功)

'payment\_status\_processing' => 'payment\_processing',

// 收款成功（支付状态已确认，代表已经收到钱了）

'payment\_status\_confirmed' => 'payment\_confirmed',

// 欺诈【当paypal的返回金额和网站金额不一致【以及货币类型】的情况，就会判定该状态】

'payment\_status\_suspected\_fraud' => 'payment\_suspected\_fraud',

// 订单支付已取消【用户进入paypal点击取消订单返回网站，或者payment\_pending订单超过xx时间未支付被脚本取消，或者客服后台取消】

'payment\_status\_canceled' => 'payment\_canceled',

// 订单审核中（订单收款成功后，进入erp，需要客服审核，才能开始发货流程，或者可能存在某些问题，被客服暂时挂起）

'status\_holded' => 'holded',

// 订单备货处理中，从成功收款进入erp并客服审核成功后，进入备货流程 到 发货前的状态

'status\_processing' => 'processing',

// 订单已发货【订单包裹被物流公司收取后】

'status\_dispatched' => 'dispatched',

// 订单已退款【已收款订单因为某些原因进行退款，譬如：仓库无货，用户收到货后发现破损退款等】

'status\_refunded' => 'refunded',

// 订单已完成，【用户收到货物xx时间后，未发起纠纷争端，订单状态标记为已完成】

'status\_complete' => 'completed',

],

];

### 订单状态的大致流程：

1.生成订单后，订单等待支付，订单的状态为 payment\_status\_pending（等待付款状态）

2当跳转到第三方支付，譬如paypal，如果用户没有使用paypal的余额，而是使用了paypal 绑定的信用卡支付，支付完成后跳转到网站，订单的状态为payment\_status\_processing（付款处理中）， 原因是paypal需要到信用卡发卡行扣款，在扣款没有完成之前，在paypal的后台可以查看这个订单， 但是，并未收到款，需要等待paypal去信用卡扣款成功， 当paypal扣款成功后，会以IPN消息的方式，给fecshop的ipn接收API发送一个post请求，告诉fecshop订单支付已收款， 然后fecshop会把订单状态改成payment\_status\_confirmed（收款成功）

3.如果用户使用的是paypal余额付款，那么付款直接成功，订单状态将直接被改为payment\_status\_confirmed（收款成功）

4.当paypal返回的订单信息和网站订单金额和货币类型不一致，就会被fecshop判定为欺诈订单， 订单类型更改为： payment\_status\_suspected\_fraud （欺诈）

5.如果用户跳转到paypal网站支付，如果用户没有支付，而是点击了取消链接，那么就会跳转回fecshop，订单将会被 取消，释放库存，订单状态更改为 payment\_status\_canceled (订单支付已取消)

6.payment\_status\_pending（等待付款状态）的订单，有一些用户跳转到支付平台后， 然后就关掉了页面，那么这个订单就会一直处于等待付款状态，占用库存，因此，我们需要后台脚本， 将一段时间内没有支付的payment\_status\_pending订单取消掉， 订单状态更改为 payment\_status\_canceled 并释放库存。

7.已付款订单的处理，一般都是在erp系统中处理， 因此通过脚本将订单推动到erp，推送成功后订单状态变成status\_holded(订单审核中)

8.erp客服审核订单，审核成功后，进入备货流程，订单状态改成status\_processing（订单备货处理中）

9.备货完成，订单发货， 订单状态改成 status\_dispatched（订单已发货）

10.订单发货，买家收到货物后，发现有问题或者其他原因要求退货，商家允许，达成一致后， 订单状态改成 status\_refunded 订单已退款，

11.如果买家没有纠纷，那么过一段时间后，订单状态为 status\_complete (订单已完成)

对于erp处理过程中会有很多的状态，可以做一个关系对应表，把这些状态做一下关系对应。

### 其他配置说明

increment\_id ：为订单编号格式

requiredAddressAttr： 为下单界面必填的字段

minuteBeforeThatReturnPendingStock： 这个是下面的后台脚本 （释放未付款订单库存的脚本）所用到的参数，将pending（未支付的订单） 的库存释放掉，这里的单位是分钟，如果您的库存为零库存（ 零库存指的是，如果没有库存 可以通过采购部门采购，相当于您不需要考虑库存），可以不需要跑这个脚本。

orderCountThatReturnPendingStock： 这个后台脚本一次性处理多少个pending订单。

**注意**：通过脚本将pending的库存返还给产品后，订单的状态将会变成取消状态， 订单取消状态，是无法进行支付的，因此，minuteBeforeThatReturnPendingStock 尽量设置的大一些，我设置的默认为10个小时，对于零库存商城 （也就是产品库存为0没有关系，可以继续卖，然后采购部门去采购，这属于零库存模式）， 这种模式可以批量将产品的所有库存设置的非常大，下面的这个脚本也不需要跑。 ，下面的这个脚本就是根据上面设置的参数来处理pending状态订单，释放产品库存的脚本。

### 释放未付款订单库存的脚本

详细参看：[order:pending订单取消脚本](http://www.fecshop.com/doc/fecshop-guide/develop/cn-1.0/guide-fecshop-console-pending-cancel.html)

### 订单支付状态的进一步验证

如果您感觉还是不放心，订单传递到erp进行发货处理的时候，加入一层付款成功验证， 譬如paypal，您可以去官方网站下载付款成功的订单，也就是csv表格 ，然后通过导入的方式，二次验证订单支付状态，这样是最稳妥的方式， 另外还需要验证一下货币和金额。

### 下单后购物车的清空：

游客用户下单后，购物车是不清空的，支付成功后返回网站再清空购物车产品

登录用户下单后，购物车直接清空，用户可以在账户中心的我的订单中查看未支付订单，重新下单。

# Fecshop SiteMap

sitemap是为了方便搜索引擎蜘蛛爬取而生成的xml文件。

### 配置文件

配置文件为：@console/config/fecshop\_local\_services/Sitemap.php,您可以通过配置的方式 对该配置文件进行重写。

return [

'sitemap' => [

'class' => 'fecshop\services\Sitemap',

'sitemapConfig' => [

/\*\*

\* 对于下面的设置，您可能感觉很啰嗦，域名作为store的key，在store service中已经设置，

\* 为什么需要在这里重新搞一套呢？ 这样做是为了更加的灵活

\*

\*/

# appfront入口

'appfront' => [

# store的key(域名)，

'fecshop.appfront.fancyecommerce.com' => [

'https' => true, # false代表使用http，true代表使用https

'sitemapDir' => '@appfront/web/sitemap.xml', # sitemap存放的地址

'showScriptName'=> false, # 是否显示index.php ，譬如http://www.fecshop.com/index.php/xxxxxx,当nginx没有设置重写，这里需要设置为true,这样url中会存在index.php，否则会404

# 这个设置对seo来说，设置为false最合适，也就是隐藏 url中index.php ，这种设置需要开启nginx的url重写

],

# store的key(域名)

'fecshop.appfront.fancyecommerce.com/fr' => [

'https' => true, # false代表使用http，true代表使用https

'sitemapDir' => '@appfront/web/fr/sitemap.xml', # sitemap存放的地址

'showScriptName'=> false,

],

'fecshop.appfront.es.fancyecommerce.com' => [

'https' => true, # false代表使用http，true代表使用https

'sitemapDir' => '@appfront/web/sitemap\_es.xml',

'showScriptName'=> false,

],

'fecshop.appfront.fancyecommerce.com/cn' => [

'https' => true, # false代表使用http，true代表使用https

'sitemapDir' => '@appfront/web/cn/sitemap.xml',

'showScriptName'=> false,

],

]

],

],

];

如果您的nginx重写没有开启，那么showScriptName需要设置成true，原因参看上面 配置里面的说明。

上面配置了4个store，(fecshop.appfront.fancyecommerce.com) 里面的各个配置项，在配置后面都有说明。

1.上面的store，就是您在各个入口下面的store，每一个store用不同的 域名或者后缀作为首页url，您可以这是pc入口，以及手机web入口，因为他们的 首页url不同，语言不同，进而内容不同，因此他们都需要有自己的sitemap文件。 每一个store配置里面都有sitemap的文件路径地址。

2.生成了sitemap.xml，我们需要通过首页域名+sitemap.xml作为访问路径。 如果在地址下面直接有对应名字的文件，则可以直接访问，但是名字是其他的名字，就需要nginx 设置了。譬如 fecshop.appfront.es.fancyecommerce.com生成的sitemap文件名字为 sitemap\_es.xml，我们需要设置一下，让url fecshop.appfront.es.fancyecommerce.com/sitemap.xml可以访问 到sitemap\_es.xml

location ~ /sitemap.xml

{

if ($host ~ .\*appfront.es.fancyecommerce.com) {

rewrite ^/sitemap\.xml /sitemap\_es.xml last;

}

}

可以参看我的nginx的配置如下：

server {

listen 80 ;

server\_name fecshop.appfront.fancyecommerce.com fecshop.appfront.es.fancyecommerce.com;

root /www/web/develop/fecshop/appfront/web;

server\_tokens off;

include none.conf;

index index.php index.html index.htm;

access\_log /www/web\_logs/access.log wwwlogs;

error\_log /www/web\_logs/error.log notice;

location ~ \.php$ {

fastcgi\_pass 127.0.0.1:9000;

fastcgi\_index index.php;

include fcgi.conf;

}

location ~ /sitemap.xml

{

if ($host ~ .\*appfront.es.fancyecommerce.com) {

rewrite ^/sitemap\.xml /sitemap\_es.xml last;

}

}

location /fr/ {

index index.php;

if (!-e $request\_filename){

rewrite . /fr/index.php last;

}

}

location /es/ {

index index.php;

if (!-e $request\_filename){

rewrite . /es/index.php last;

}

}

location /cn/ {

index index.php;

if (!-e $request\_filename){

rewrite . /cn/index.php last;

}

}

location /de/ {

index index.php;

if (!-e $request\_filename){

rewrite . /de/index.php last;

}

}

location ~ .\*\.(gif|jpg|jpeg|png|bmp|swf)$ {

expires 30d;

}

location ~ .\*\.(js|css)?$ {

expires 12h;

}

}

通过上面的nginx配置

location ~ /sitemap.xml

{

if ($host ~ .\*appfront.es.fancyecommerce.com) {

rewrite ^/sitemap\.xml /sitemap\_es.xml last;

}

}

fecshop.appfront.es.fancyecommerce.com/sitemap.xml 访问的文件 不是 /www/web/develop/fecshop/appfront/web/sitemap.xml 而是 /www/web/develop/fecshop/appfront/web/sitemap\_es.xml 文件。

对于访问url中路径和名字与实际文件一致的，不需要在nginx做更改配置。

3.生成xml

通过上面，我们配置好了各个store的参数，我们就可以生成sitemap文件了

执行的shell文件为：@fecshop/shell/sitemapGeneral.sh

执行完了，就会在配置相应的地方生成对应的sitemap文件。

# Fecshop Robots

就是web目录下的robots.txt，譬如 @appfront/web/robots.txt文件 ,具体的语法请自行查询。默认是允许所有搜索引擎爬取的。

# Fecshop Url重写

fecshop url 重写，属于自定义的方式重写，定义的重写会被保存到mongodb中。

### 1.去掉index.php

对于url: <https://fecshop.appfront.fancyecommerce.com/index.php/special-occasion> 为了seo的考虑，我们希望把index.php/去掉，下面是设置方法。

找到文件 @appfront/config/fecshop\_local\_services/Url.php ,内容如下：

return [

'url' => [

'showScriptName'=> false, # if is show index.php in url. if set false ,you must config nginx rewrite

],

];

showScriptName：true代表url中显示index.php ,false代表url中不显示index.php

当设置了false，也就是不显示index.php，需要在nginx设置重写 ，具体可以搜索这个(用bing.com搜索就不错，不需要翻墙，也可以 切换英文搜索)。

### 2. 重写原理

2.1 在数据库的产品或分类保存的时候，会有一个唯一的url\_key字符串，和真实的yii2的 url key对应，譬如/xxxxxxx 对应 /catalog/product/index?id=xxxx

2.2当一个url访问的时候，会到数据库(mongodb)中查询，该url是否在数据库中存在，如果存在， 则会使用对应的真实的yii2的url路径，譬如上面的/catalog/product/index

2.3 执行相应的模块

关于重写的原理详细参看：[yii2 Url 自定义 伪静态url](http://www.fancyecommerce.com/2016/05/18/yii2-url-%E8%87%AA%E5%AE%9A%E4%B9%89-%E4%BC%AA%E9%9D%99%E6%80%81url/)

### 3. 重写的url类型

3.1 page页面

3.2 category分类页面

3.3 product产品页面

在上述页面保存的时候，如果填写url\_key就会使用填写的url\_key，如果不填写 就会使用名字生成。

如果生成的url在数据库中存在，那么会在后面加入一组随机数字，如果随机数字还存在， 那么就会使用另外一组随机数字，直到唯一为止。

Url自定义（Url 重写）是为了seo，让网页中的关键字，标题，在url中也出现。

### 4.fecshop实现重写的文件

文件为：@fecshop/yii/web/Request.php ，重写了@yii/web/Request的一部分方法实现的。 具体的实现方法，你可以参看文件@fecshop/yii/web/Request.php的内容

# Fecshop 邮件

当用户注册邮箱，下单，忘记密码等，都会给用户发送一封邮件，邮件部分 支持多语言。

### SMTP配置文件为：

@common/config/fecshop\_local\_services/Email.php

配置如下：

return [

'email' => [

'mailerConfig' => [

# 默认通用配置

'default' => [

'class' => 'yii\swiftmailer\Mailer',

'transport' => [

'class' => 'Swift\_SmtpTransport',

'host' => 'smtp.qq.com', #SMTP Host

'username' => '372716335@qq.com', #SMTP 账号

'password' => 'wffmbummgnhhcbbj', #SMTP 密码

'port' => '587', #SMTP 端口

'encryption' => 'tls',

],

'messageConfig'=>[

'charset'=>'UTF-8',

],

],

],

],

];

对于上面配置中的的 mailerConfig，除了default，您还可以继续添加其他的SMTP 配置，

您只要配置完上面的文件，就把smtp配置好了。

### 邮件选择smtp配置

fecshop的配置文件为：@fecshop/config/services/Email.php

在邮件模板里面可以指定使用上面配置文件中的mailerConfig里面的 数组key，譬如default。

在customer子服务中就可以设置，譬如：

'login' => [

'enable' => true,

# 邮件内容的动态数据提供部分

'widget' => 'fecshop\services\email\widgets\customer\account\login\Body',

# 邮件内容的view部分

'viewPath' => '@fecshop/services/email/views/customer/account/login',

# 如果不定义 mailerConfig，则会使用email service里面的默认配置

'mailerConfig' => 'default',

],

### 公用配置

对于邮件里面会出现一些store方面的公用配置,在配置文件中可以看到

//公用配置

'mailerInfo' => [

#在邮件中显示的Store的名字

'storeName' => 'FecShop',

# 在邮件中显示的电话

'phone' => 'xxxxxxxxxx',

# 在邮件中显示的联系邮箱地址。

'contacts' => [

'emailAddress' => '2358269014@qq.com',

],

],

修改成您自己的信息即可.

**注意**：不要在 vendor/fancyecommerce下面修改任何文件， 对配置的更改，您应该在您的配置目录下添加， 譬如您可以在 @appfront/config/fecshop\_local\_services/Email.php 下面 添加配置，在系统初始化的时候，高优先级的配置会覆盖fecshop的配置。

### 邮件模板配置介绍，以及邮件多语言原理

在文件 @fecshop/config/services/Email.php中对 子服务 customer 里面，有很多邮件模板的配置

viewPath 就是邮件模板html部分 ， widget是动态数据提供部分。

譬如:

'customer' => [

'class' => 'fecshop\services\email\Customer',

# 各个邮件的模板部分：

'emailTheme' => [

# 注册账户发送的邮件的模板配置

'register' => [

'enable' => true,

# 邮件内容的动态数据提供部分

'widget' => 'fecshop\services\email\widgets\customer\account\register\Body',

# 邮件内容的view部分

'viewPath' => '@fecshop/services/email/views/customer/account/register',

/\*\*

\* 1.默认是default，譬如下面的 'mailerConfig' => 'default',你可以不填写，因为默认就是default

\* 2.您可以使用上面email服务的配置项mailerConfig中的设置的各个项，譬如填写default 或者 login等。

\* 3.您还可以直接填写数组的配置（完整配置），譬如：

\* 'register' => [

\* 'class' => 'yii\swiftmailer\Mailer',

\* 'transport' => [

\* 'class' => 'Swift\_SmtpTransport',

\* 'host' => 'smtp.qq.com',

\* 'username' => '372716335@qq.com',

\* 'password' => 'wffmbummgnhhcbbj',

\* 'port' => '587',

\* 'encryption' => 'tls',

\* ],

\* 'messageConfig'=>[

\* 'charset'=>'UTF-8',

\* ],

\*

\* ],

\*/

'mailerConfig' => 'default',

],

# 登录用户发送邮件的模板的设置。

'login' => [

'enable' => true,

# 邮件内容的动态数据提供部分

'widget' => 'fecshop\services\email\widgets\customer\account\login\Body',

# 邮件内容的view部分

'viewPath' => '@fecshop/services/email/views/customer/account/login',

# 如果不定义 mailerConfig，则会使用email service里面的默认配置

'mailerConfig' => 'default',

],

# 忘记密码发送邮件的模板的设置

'forgotPassword' => [

'enable' => true,

'widget' => 'fecshop\services\email\widgets\customer\account\forgotpassword\Body',

# 邮件内容的view部分

'viewPath' => '@fecshop/services/email/views/customer/account/forgotpassword',

#忘记密码邮件发送后的超时时间。

'passwordResetTokenExpire' => 86400, # 3600\*24\*1, # 一天

# 如果不定义 mailerConfig，则会使用email service里面的默认配置

# 通过邮箱找回密码，发送的resetToken过期的秒数

'mailerConfig' => 'default',

],

# 联系我们发送的邮件模板

'contacts' => [

'enable' => true,

# 联系我们的邮箱地址

# widget 邮件动态数据提供部分。

'widget' => 'fecshop\services\email\widgets\customer\contacts\Body',

# 邮件内容的view部分

'viewPath' => '@fecshop/services/email/views/customer/contacts',

'address' => '2358269014@qq.com',

# 如果不定义 mailerConfig，则会使用email service里面的默认配置

//'mailerConfig' => 'default',

],

# 订阅newsletter后发送的邮件模板。

'newsletter' => [

# 订阅邮件成功后，是否发送邮件给用户

'enable' => true,

# widget 邮件动态数据提供部分。

'widget' => 'fecshop\services\email\widgets\customer\newsletter\Body',

# 邮件内容的view部分

'viewPath' => '@fecshop/services/email/views/customer/newsletter',

# 如果不定义 mailerConfig，则会使用email service里面的默认配置

'mailerConfig' => 'default',

],

],

],

对于邮件模板 ， widget （动态数据） 和 viewPath（静态文件） 功能形成了模板内容。

widget 对应的是动态数据的php对象，譬如： 'widget' => 'fecshop\services\email\widgets\customer\newsletter\Body' 对应的是@fecshop\services\email\widgets\customer\newsletter\Body.php文件

view是html部分的路径，在该路径下面需要有subject（邮件标题）和body（邮件内容）两个部分，然后加上语言， 譬如：@fecshop/services/email/views/customer/newsletter下面有 subject\_en.php和body\_en.php两个文件，代表英文语言的邮件标题和邮件内容， 你可以添加subject\_fr.php和body\_fr.php两个文件，代表 法文状态下的邮件标题和邮件内容。

如果您想要得到法文的邮件，但是没有subject\_fr.php 和 body\_fr.php 文件，那么，系统会使用默认语言的邮件，也就是subject\_en.php 和body\_en.php

### 邮件重写

如果您想重写邮件的内容，那么您在配置中重新指定viewPath和widget 的值，在路径中重新写subject和body文件即可。

fecshop的配置文件为：@fecshop/config/services/Email.php

# 注册账户发送的邮件的模板配置

'register' => [

'enable' => true,

# 邮件内容的动态数据提供部分

'widget' => 'fecshop\services\email\widgets\customer\account\register\Body',

# 邮件内容的view部分

'viewPath' => '@fecshop/services/email/views/customer/account/register',

'mailerConfig' => 'default',

],

譬如重写登录发送邮件的模板:

您可以新建一个widget 继承fecshop\services\email\widgets\customer\account\login\Body, 来实现widget的重写。

对于view文件，可以新建一个viewPath，然后把 @fecshop/services/email/views/customer/account/login 里面的所有文件复制 到本地的viewPath中，然后重写即可。

# Fecshop Cron 脚本

需要在线上设置的定时脚本。脚本地址在@fecshop/shell路径下面

1.每天一次脚本：

computeProductFinalPrice.sh

2.时间间隔：将pending订单（创建时间）未支付超过30分钟的订单取消，并归还产品库存。

returnPendingProductQtyStock.sh

3.使用工具脚本

urlRewrite.sh url重写的脚本，这个脚本在url 重写不准确或者其他情况下需要 手动跑一次，用来纠正url 重写的数据问题

fullSearchSync.sh这个脚本是产品搜索生成mongodb产品相应语言的full Text Search表 的脚本。

# Fecshop验证码

在注册账户，登录，找回密码等表单提交的地方需要验证码， 您可以通过配置的方式决定是否开启验证码。

配置文件为：@fecshop/app/appfront/config/modules/Customer.php

打开后通过注释就可以找到各个配置项.

return [

'customer' => [

'class' => '\fecshop\app\appfront\modules\Customer\Module',

'params'=> [

'register' => [

# 账号注册成功后，是否自动登录

'successAutoLogin' => true,

# 注册登录成功后，跳转的url

'loginSuccessRedirectUrlKey' => 'customer/account',

# 注册页面的验证码是否开启

'registerPageCaptcha' => true,

],

'login' => [

# 在登录页面 customer/account/login 页面登录成功后跳转的urlkey，

'loginPageSuccessRedirectUrlKey' => 'customer/account',

# 在其他页面的弹框方式登录的账号成功后，的页面跳转，如果是false，则代表返回原来的页面。

'otherPageSuccessRedirectUrlKey' => false ,

# 登录页面的验证码是否开启

'loginPageCaptcha' => false,

# 邮件信息，登录账号后是否发送邮件

],

'forgotPassword' => [

# 忘记密码页面的验证码是否开启

'forgotCaptcha' => true,

],

'leftMenu' => [

'Account Dashboard' => 'customer/account',

'Account Information' => 'customer/editaccount',

'Address Book' => 'customer/address',

'My Orders' => 'customer/order',

'My Product Reviews' => 'customer/productreview',

'My Favorite' => 'customer/productfavorite',

],

'contacts' => [

# 联系我们页面的验证码是否开启

'contactsCaptcha' => true,

],

'newsletterSubscribe' => [

],

],

],

];

您可以在appfront通过添加配置覆盖的方式，重新设置是否开启验证码。

# FecShop 缓存

fecshop 有整页缓存和局部缓存

### 缓存刷新

进入appadmin后台,刷新缓存，具体如图：



因为各个入口的redisCache都是独立配置的database，因此，后台刷新缓存，需要加载redis cache的配置， 因此，需要在配置中指定具体的配置文件

打开: @appadmin/config/fecshop\_local\_services/Fecadmin.php 可以看到配置：

'fecadmin' => [

'params' => [

/\*\*

\* Fecshop缓存是基于redis，下面是各个入口redis配置所在的文件路径

\* 1.`commonConfig`是公用部分

\* 2.app开头的key，指的是各个入口的redis所在的配置文件

\* 这个配置的作用，是为了在后台清空各个入口的全部缓存，因此需要加载相应的redis的配置

\*/

'cacheRedisConfigFile' => [

'commonConfig' => '@common/config/main-local.php',

'appAdmin' => '@appadmin/config/main-local.php',

'appApi' => '@appapi/config/main-local.php',

'appFront' => '@appfront/config/main-local.php',

'appHtml5' => '@apphtml5/config/main-local.php',

'appServer' => '@appserver/config/main-local.php',

],

],

],

cacheRedisConfigFile: 数组的作用是，指定各个入口的redisCache配置所在文件，其中， commonConfig 是公用redisCache配置。

关于这个功能的具体实现，您可以参看文件：@fecshop\app\appadmin\modules\Fecadmin\block\cache\Index.php

### Appfront AppHtml5 入口使用缓存

实现原理为yii2的full page cache

配置：

@appfront/config/fecshop\_local\_services/Cache.php

具体如下：

return [

'cache' => [

/\*\*

\* cache 总开关，设置false后，无论cacheConfig中的enable

\* 是否为true，都会全部关闭掉cache。

\*/

'enable'=> true,

/\*\*

\* 各个页面cache的配置

\*/

'cacheConfig' => [

# 分类页面

'category' => [

'enable' => true, # 是否开启分类页面的缓存

'timeout' => 3600, # 则设置缓存的过期时间，这里设置为秒

'disableUrlParam' => 'fecshop', # 如果开启缓存，在url加入什么参数后，系统不读取缓存，这个选项是为了方便在不刷新缓存的情况下，查看无缓存的页面是什么样子。

# url出现的这些参数的值，将参与cache唯一key的生成。

'cacheUrlParam' => [

# 分页，排序，等参数

'p','dir','sort','numPerPage',

# 侧栏属性过滤等参数

'price','size','color',

'style','dresses-length','pattern-type',

'collar','xinghao','cpu'

],

],

# 产品页面

'product' => [

'enable' => true, # 是否开启产品页面的缓存

'timeout' => 3600, #则设置缓存的过期时间，这里设置为秒

'disableUrlParam' => 'fecshop', # 如果开启缓存，在url加入什么参数后，系统不读取缓存，这个选项是为了方便在不刷新缓存的情况下，查看无缓存的页面是什么样子。

],

# 首页页面

'home' => [

'enable' => true, # 是否开启首页页面的缓存

'timeout' => 3600, # 则设置缓存的过期时间，这里设置为秒

'disableUrlParam' => 'fecshop', # 如果开启缓存，在url加入什么参数后，系统不读取缓存，这个选项是为了方便在不刷新缓存的情况下，查看无缓存的页面是什么样子。

],

# Article（page）页面

'article' => [

'enable' => true, # 是否开启Article页面的缓存

'timeout' => 3600, # 则设置缓存的过期时间，这里设置为秒

'disableUrlParam' => 'fecshop',# 如果开启缓存，在url加入什么参数后，系统不读取缓存，这个选项是为了方便在不刷新缓存的情况下，查看无缓存的页面是什么样子。

],

],

],

];

详细看里面的注释代表的具体含义。

开启缓存后，页面的数据将从cache中读取，页面的动态数据部分，会 通过ajax的方式把页面的动态数据加载过来。 通过整页缓存，页面的加载就会非常的快。

### Appfront AppHtml5 入口使用局部缓存

局部缓存，相当于一个区块， 譬如fecshop pc端web的menu部分，头部header部分，尾部footer部分等，都可以 通过局部缓存的方式，将其缓存起来， 在购物车，下单，账户中心等内容全部是动态内容，不能做全页缓存的页面， 可以通过局部缓存的方式缓存某些非动态内容的区块， fecshop是通过 page - widget 服务实现的。

首先参看配置文件：

@appfront/config/fecshop\_local\_services/Page.php , 代码如下：

'widget' => [

'widgetConfig' => [

'head' => [

# 动态数据提供部分

'class' => 'fecshop\app\appfront\widgets\Head',

# 根据多模板的优先级，依次去模板找查找该文件，直到找到这个文件。

'view' => 'widgets/head.php',

# 缓存

'cache' => [

'enable' => false, # 是否开启

'timeout' => 4500, # 缓存过期时间

],

],

'header' => [

'class' => 'fecshop\app\appfront\widgets\Headers',

# 根据多模板的优先级，依次去模板找查找该文件，直到找到这个文件。

'view' => 'widgets/header.php',

'cache' => [

'enable' => false,

'timeout' => 4500,

],

],

'topsearch' => [

'view' => 'widgets/topsearch.php',

],

'menu' => [

'class' => 'fecshop\app\appfront\widgets\Menu',

# 根据多模板的优先级，依次去模板找查找该文件，直到找到这个文件。

'view' => 'widgets/menu.php',

'cache' => [

'enable' => false,

//'timeout' => 4500,

],

],

'footer' => [

'class' => 'fecshop\app\appfront\widgets\Footer',

# 根据多模板的优先级，依次去模板找查找该文件，直到找到这个文件。

'view' => 'widgets/footer.php',

'cache' => [

'enable' => false,

//'timeout' => 4500,

],

],

'scroll' => [

#'class' => 'fecshop\app\appfront\modules\Cms\block\widgets\Scroll',

# 根据多模板的优先级，依次去模板找查找该文件，直到找到这个文件。

'view' => 'widgets/scroll.php',

],

'breadcrumbs' => [

'view' => 'widgets/breadcrumbs.php',

],

'flashmessage' => [

'view' => 'widgets/flashmessage.php',

],

]

],

上面是各个区块的配置， 各个区块的调用方式为：

<?= Yii::$service->page->widget->render('menu',$this); ?>

如果上面的配置设置了cache 为开启，并且class对应的动态数据提供对象， 存在函数 getCacheKey ,则可以使用缓存

譬如：head的配置如下，下面定义了cache是否开启，如果设置为true，则 代表开启缓存， timeout代表缓存的过期时间。

'head' => [

# 动态数据提供部分

'class' => 'fecshop\app\appfront\widgets\Head',

# 根据多模板的优先级，依次去模板找查找该文件，直到找到这个文件。

'view' => 'widgets/head.php',

# 缓存

'cache' => [

'enable' => false, # 是否开启

'timeout' => 4500, # 缓存过期时间

],

],

上面配置中的class对应的Head.php文件内容为：

<?php

namespace fecshop\app\appfront\widgets;

use Yii;

use fecshop\interfaces\block\BlockCache;

class Head implements BlockCache

{

public function getLastData()

{

return [

];

}

public function getCacheKey(){

$store = Yii::$service->store->currentStore;

$moduleId = Yii::$app->controller->module->id;

$controllerId = Yii::$app->controller->id;

$actionId = Yii::$app->controller->action->id;

$urlPathKey = $moduleId.'\_'.$controllerId.'\_'.$actionId;

return self::BLOCK\_CACHE\_PREFIX.'\_'.$store.'\_'.$urlPathKey;

}

}

getCacheKey() 返回的是cache的唯一key。

通过上面的配置可以看出，head， header， footer， menu 都是可以设置局部缓存的。

### 关于Yii2 cache

更多的参看：[Yii2 cache](http://www.yiichina.com/doc/guide/2.0/caching-overview)

# 搜索

对于搜索，目前使用了2种方式，对于外文，譬如法语，德语，西班牙语 都是用的mongodb的full Search ，但是mongodb的搜索不支持中文， 目前中文部分用的是xunsearch，对于中小网站都没啥问题， 后面会扩展用elasticSearch(目前还没有做)来做搜索工作。

### 安装xunsearch

[http://www.fancyecommerce.com/2016/09/24/xunsearch-安装，使用/](http://www.fancyecommerce.com/2016/09/24/xunsearch-%e5%ae%89%e8%a3%85%ef%bc%8c%e4%bd%bf%e7%94%a8/)

其他参考资料：

<http://www.xunsearch.com/>

<https://getyii.com/topic/61>

<http://www.yiichina.com/code/661>

### 搜索配置部分：

配置文件为：

fecshop的配置文件路径为： @fecshop/config/services/Search.php

用户可以在@common/config/fecshop\_local\_services/Search.php 这里进行重写配置。

完整的配置如下：

return [

'search' => [

'class' => 'fecshop\services\Search',

'filterAttr' => [

'color','size', # 在搜索页面侧栏的搜索过滤属性字段

],

'childService' => [

'mongoSearch' => [

'class' => 'fecshop\services\search\MongoSearch',

'searchIndexConfig' => [

'name' => 10,

'description' => 5,

],

#more : https://docs.mongodb.com/manual/reference/text-search-languages/#text-search-languages

'searchLang' => [

'en' => 'english',

'fr' => 'french',

'de' => 'german',

'es' => 'spanish',

'ru' => 'russian',

'pt' => 'portuguese',

],

],

'xunSearch' => [

'class' => 'fecshop\services\search\XunSearch',

'fuzzy' => true, # 是否开启模糊查询

'synonyms' => true, #是否开启同义词翻译

'searchLang' => [

'zh' => 'chinese',

],

],

],

]

];

1.侧栏属性过滤

filterAttr : 在搜索页面侧栏显示的过滤属性，选择后在侧栏会显示该属性的过滤选项信息

2.mongodb配置 也就是配置数组里面 mongoSearch下面的配置。

searchIndexConfig为配置搜索的权重比， 上面的配置代表为：name的权重为10，description的权重为5.

searchLang为语言设置，因为搜索要涉及到分词，因此要和语言挂钩的， 对于mongodb的搜索，有一个语言的文字对应表，网址为： <https://docs.mongodb.com/manual/reference/text-search-languages/#text-search-languages> ，您需要根据上面语言的名字，在这里配置各个语言的参数。譬如：法语配置，fr为语言二位简码， 在上面网址中法语的名字为：french, 因此在上面进行配置：'fr' => 'french',

3.国产搜索软件xunSearch

xunSearch作者写了一个yii2的扩展，用于中文搜索， 因为mongodb是老外的玩意，不支持中文分词，因此，我用了比较小巧 的xunSearch，（当然专业的搜索工具还是elasticSearch）， 关于语言的配置和上面大致一样，不过这里只设置中文就好了，如果您是 外贸网站，那么根本不需要使用xunsearch啦。

xunSearch配置组件：组件的配置文件：fecshop/config/components/XunSearch.php， 打开这个文件，你会发现：'iniDirectory' => '@fecshop/config/xunsearch', 也就是active record的配置，有点类似elasticSearch的mapping， 也就是数据model的配置文件的文件夹路径，我们进入@fecshop/config/xunsearch， 发现下面有search.ini文件，我们打开@fecshop/config/xunsearch/search.ini 文件，这里是对search的一些配置， 具体语法可以参看<http://www.xunsearch.com/>查阅，这里不做讲解。  当然，如果您不想扩展fecshop的xunSearch搜索，你可以不需要明白这些，直接用就好。

对于@fecshop/config/xunsearch/search.ini中的配置，我们需要了解的是 project.name = search ， 这个和fecshop\models\xunsearch\Search.php里面的projectName()返回的名字一致即可。

### 数据同步到搜索工具中

通过上面，我们大致就设置完了，然后我们需要跑初始化脚本， 也就是将数据同步到我们的搜索工具中，需要跑的脚本为： 首先shell进入路径 @fecshop/shell/search ， 然后执行 sh fullSearchSync.sh 即可，跑完后，mongodb的搜索和xunSearch的搜索都把数据同步过去。

[root@iZ942k2d5ezZ search]# pwd

/www/web/develop/fecshop/vendor/fancyecommerce/fecshop/shell/search

[root@iZ942k2d5ezZ search]# sh fullSearchSync.sh

There are 37 products to process

There are 1 pages to process

##############ALL BEGINING###############

Page 1 done

begin delete Mongodb Search Date

##############ALL COMPLETE###############

[root@iZ942k2d5ezZ search]#

### 去除某个搜索

如果您想去除某个搜索，譬如您是做跨境出口电商的，不想做中文，那么 可以去除xunsearch，您只需要在配置中加入一个'enableService' => false就可以了， 譬如：

return [

'search' => [

'class' => 'fecshop\services\Search',

'filterAttr' => [

'color','size', # 在搜索页面侧栏的搜索过滤属性字段

],

'childService' => [

'mongoSearch' => [

'class' => 'fecshop\services\search\MongoSearch',

'searchIndexConfig' => [

'name' => 10,

'description' => 5,

],

#more : https://docs.mongodb.com/manual/reference/text-search-languages/#text-search-languages

'searchLang' => [

'en' => 'english',

'fr' => 'french',

'de' => 'german',

'es' => 'spanish',

'ru' => 'russian',

'pt' => 'portuguese',

],

],

'xunSearch' => [

'class' => 'fecshop\services\search\XunSearch',

'enableService' => false,

'fuzzy' => true, # 是否开启模糊查询

'synonyms' => true, #是否开启同义词翻译

'searchLang' => [

'zh' => 'chinese',

],

],

],

]

];

加入 'enableService' => false, 后，xunsearch将可不用。

因此，您可以打开文件：@common\config\fecshop\_local\_services\Search.php 加入 'enableService' => false, 即可。

'xunSearch' => [

'fuzzy' => true, // 是否开启模糊查询

'enableService' => false,

'synonyms' => true, //是否开启同义词翻译

'searchLang' => [

'zh' => 'chinese',

],

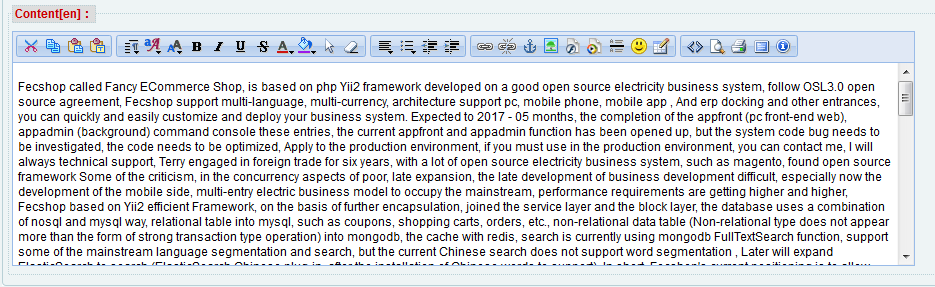
],

### 使用

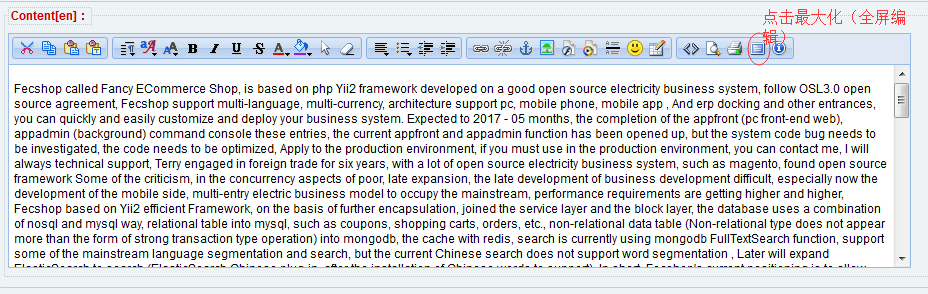
在前台就可以使用搜索功能进行搜索了。

# Fecshop 后台编辑器

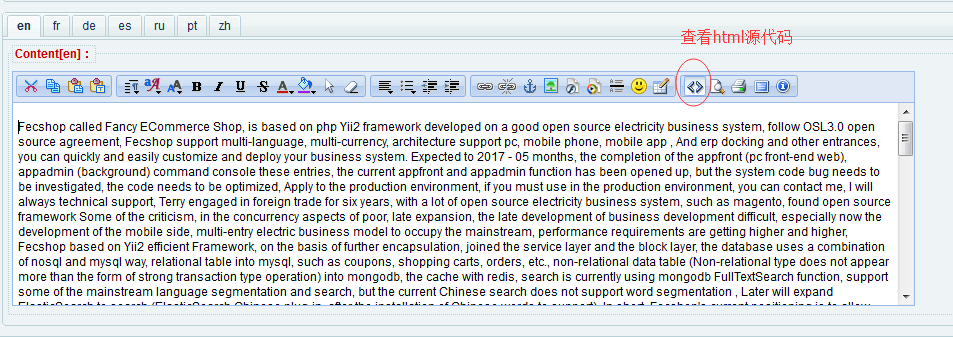
Fecshop 后台编辑器 就是在后台编辑栏里面的编辑器，如图



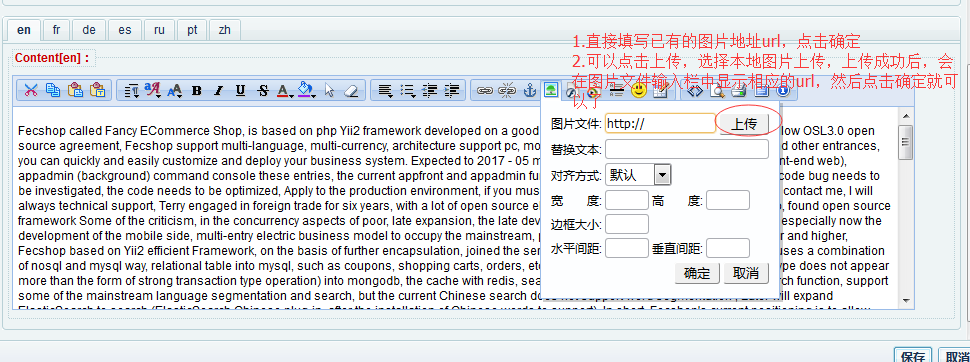
点击编辑栏最大化，如图:



查看html源码



上传图片



上传图片后的效果



# Fecshop Event

Fecshop 事件

fecshop的事件，没有使用yii2的事件机制，而是自己做了一个比较 简单的类似事件机制。

### 事件使用

目前只有四个事件：

return [

'event' => [

'class' => 'fecshop\services\Event',

# 事件配置表，这是Fecshop默认存在的事件。您可以在配置中添加您的事件函数。

'eventList' => [

# 加入购物车before

'event\_add\_to\_cart\_before' => [],

# 加入购物车after

'event\_add\_to\_cart\_after' => [],

# 生成订单before

'event\_generate\_order\_before' => [],

# 生成订单after

'event\_generate\_order\_after' => [],

]

],

];

添加事件举例：首先在配置中加入如下:

return [

'event' => [

'eventList' => [

# 加入购物车前

'event\_add\_to\_cart\_before' => [

['appfront\event\CartTest1','beforeAdd1'],

['appfront\event\CartTest2','beforeAdd2'],

],

]

],

];

然后新建 文件appfront\event\CartTest1.php 并新建

class CartTest1

{

public static function beforeAdd1($cartInfo){

//var\_dump($cartInfo);

}

}

即可，在函数里面写您想要做的事情了

由于fecshop采用service的方式，感觉event的用处不大， 但还是得有这个机制，就这样做了4个事件。

# fecshop 配置加速

fecshop配置加速的将fecshop的N个配置文件合并成一个配置文件， 进而减少了使用merge合并数组耗费的时间， 在并发不高的网站不需要做这个配置加速，影响不大，并发量高 的网站推荐使用

### 1.开启方法

1.1合并生成单配置文件

在web（@app/web/）目录下会发现文件index-merge-config.php ， 访问执行：www.domain.com/index-merge-config.php ，就会 把所有的配置合并并写入 @app/merge\_config.php中

1.2开启使用单配置文件

打开文件 @app/web/index.php 找到代码行 $use\_merge\_config\_file = false; 改成true，即可。

### 2.注意

如果您的配置文件进行了更新，更新后，您需要重新执行一下 www.domain.com/index-merge-config.php ， 否则@app/merge\_config.php中的配置还是原来的， 造成配置没有生效。

### 3.帮助

关于单文件配置的原理可以参看文章：[yii2 配置加速 – N个配置文件生成一个配置文件](http://www.fancyecommerce.com/2017/04/10/yii2-%e9%85%8d%e7%bd%ae%e5%8a%a0%e9%80%9f-n%e4%b8%aa%e9%85%8d%e7%bd%ae%e6%96%87%e4%bb%b6%e7%94%9f%e6%88%90%e4%b8%80%e4%b8%aa%e9%85%8d%e7%bd%ae%e6%96%87%e4%bb%b6/)

# Fecshop Log日志

1.日志配置 @app/config/main.php 打开后，可以看到log组件的配置如下：

'log' =>[

# 追踪级别

# 消息跟踪级别

# 在开发的时候，通常希望看到每个日志消息来自哪里。这个是能够被实现的，通过配置 log 组件的 yii\log\Dispatcher::traceLevel 属性， 就像下面这样：

'traceLevel' => 3,

# 通过 yii\log\Logger 对象，日志消息被保存在一个数组里。为了这个数组的内存消耗， 当数组积累了一定数量的日志消息，日志对象每次都将刷新被记录的消息到 log targets 中。 你可以通过配置 log 组件的 yii\log\Dispatcher::flushInterval 属性来自定义数量

'flushInterval' => 1,

'targets' => [

'file' =>[

//'levels' => ['trace'],

'categories' => ['fecshop\_debug'],

'class' => 'yii\log\FileTarget',

# 当 yii\log\Logger 对象刷新日志消息到 log targets 的时候，它们并 不能立即获取导出的消息。相反，消息导出仅仅在一个日志目标累积了一定数量的过滤消息的时候才会发生。你可以通过配置 个别的 log targets 的 yii\log\Target::exportInterval 属性来 自定义这个数量，就像下面这样：

'exportInterval' => 1,

# 输出文件

'logFile' => '@apphtml5/runtime/fecshop\_logs/fecshop\_debug.log',

# 你可以通过配置 yii\log\Target::prefix 的属性来自定义格式，这个属性是一个PHP可调用体返回的自定义消息前缀

'prefix' => function ($message) {

return $message;

},

# 除了消息前缀以外，日志目标也可以追加一些上下文信息到每组日志消息中。 默认情况下，这些全局的PHP变量的值被包含在：$\_GET, $\_POST, $\_FILES, $\_COOKIE,$\_SESSION 和 $\_SERVER 中。 你可以通过配置 yii\log\Target::logVars 属性适应这个行为，这个属性是你想要通过日志目标包含的全局变量名称。 举个例子，下面的日志目标配置指明了只有 $\_SERVER 变量的值将被追加到日志消息中。

# 你可以将 logVars 配置成一个空数组来完全禁止上下文信息包含。或者假如你想要实现你自己提供上下文信息的方式， 你可以重写 yii\log\Target::getContextMessage() 方法。

'logVars' => [],

],

],

],

如果要开启log，可以把注释去掉，查看 'logFile' => '@apphtml5/runtime/fecshop\_logs/fecshop\_debug.log', 对应的文件是否存在，如果不存在，需要新建文件 fecshop\_debug.log ，并设置可写即可完成配置。

2.日志的使用

\Yii::info($post\_log,'fecshop\_debug');

其中第二个参数 fecshop\_debug 对应上面的配置 'categories' => ['fecshop\_debug'],

您可以定义多个 targets 每个targets的categories 不同，然后在输出的时候做区分 就可以把日志写入到不同的log文件里面了

[Yii2 Log的特性](http://www.fancyecommerce.com/2017/05/18/yii2-log-%E7%89%B9%E6%80%A7/)

通过上面的特性，可以知道log在使用过程中的一些情况，log的配置以及log 文件如果存在问题，调用log的函数\Yii::info($post\_log,'fecshop\_debug') 不会卡住，会继续执行。

fecshop 代码里面会加入一些log的代码，在线上环境中可以把 fecshop\_debug 对应的配置去掉 就不会输出log，您可以自定义一些log的配置，然后做自定义log的输出。

关于Yii Log更多的资料可以参阅： [Yii Log](http://www.yiichina.com/doc/guide/2.0/runtime-logging)

# Fecshop 事务

fecshop事务指的是数据库层面的事务 在执行过程中要么一起成功要么一起失败

### Fecshop事务介绍

fecshop中的事务用的比较少，只有一些非常需要事务的地方才用 ，大致在3处使用了事务，加入购物车，优惠券，生成订单 ，而且只是部分使用，而不是全部，因为有一些失败了并没有太大的影响。

1.譬如生成订单的时候需要清空购物车，即使生成了订单，没有清空购物车 但这又没有太大的影响，因此事务应用的比较少，

2.譬如有一些公司对库存不敏感，没有库存了，可以去采购，因此相当于0库存 只有上下架状态，库存个数不影响销售，这种情况下生成订单 不去扣库存，同样没有太多的影响， 因此，您可以根据您们自己的业务类型进行自己勾画业务。

3.services里面没有添加任何事务，事务的使用是在controller和 block层， services层如果您一定要在里面使用，那么您二开的时候一定要 规避事务嵌套，一个事务没有结束又嵌套另外一个事务会出问题。

4.mysql事务操作中，读过的数据，会被上读锁，事务未结束前， 读的数据是不允许做更新删除操作，因此，事务操作，where条件一定 要做到索引搜索，不然innodb会进行锁表， 譬如update user set sex = 'men' where name = 'xxxx' ，如果使用innodb， user表的name字段没有索引，那么update操作会锁表， 如果name字段有索引，那么只会锁相应的行，这个了解innodb的原理就会明白

5.对于mysql的操作，一般解决 锁表的问题 和 索引的问题，基本可以解决90%的问题，基本可以解决90% 的问题，当然锁的问题除了事务类操作，也有一部分是索引导致的，可以通过 索引，将表锁搞成行锁 ，因此，在更新操作，删除操作的过程中，一定要注意，不要锁表， 而应该锁行。

# Fecshop 消息MQ

这里的MQ，以及例子为rabbitMQ 的使用，当然，您可以使用其他的Mq， zhuravljov/yii2-queue同样支持，[详情参看地址](https://github.com/zhuravljov/yii2-queue/blob/master/docs/guide/README.md#queue-drivers)

### 1.环境

queue的支持，需要安装rabbitMq和php amqp扩展

1.1 [安装 RabbitMQ – centos 6](http://www.fancyecommerce.com/2017/06/02/%e5%ae%89%e8%a3%85-rabbitmq-centos-6/)

1.2 [rabbitmq php 安装amqp扩展](http://www.fancyecommerce.com/2017/06/02/rabbitmq-php-%e5%ae%89%e8%a3%85amqp%e6%89%a9%e5%b1%95/)

1.3 安装 [Yii2 php-amqplib](https://github.com/zhuravljov/yii2-queue)扩展 ，您在根目录的composer.json 中require部分添加代码 "php-amqplib/php-amqplib": "2.6.\*" 和 "php-amqplib/php-amqplib": "2.6.\*" 即可。 即可

"require": {

"php": ">=5.4.0",

"yiisoft/yii2": ">=2.0.6",

"yiisoft/yii2-bootstrap": "\*",

"yiisoft/yii2-swiftmailer": "\*",

"yiisoft/yii2-apidoc": "~2.0.0",

"fancyecommerce/fecshop": ">=1.1.3.1",

"php-amqplib/php-amqplib": "2.6.\*",

"zhuravljov/yii2-queue": "\*"

},

然后执行 composer update ,安装。

### 2.Mq的设置

在配置文件@fecshop\app\console\config\console.php

//配置RabbitMq 部分

'bootstrap' => [

'queue', // The component registers own console commands

],

'components' => [

'queue' => [

//'class' => \zhuravljov\yii\queue\amqp\Queue::class,

'class' => 'zhuravljov\yii\queue\amqp\Queue',

'host' => 'localhost',

'port' => 5672,

'user' => 'mqadmin',

'password' => 'mqadmin20177',

'queueName' => 'queue',

],

],

上面的用户名和密码就是在上面安装rabbitMq时设置的用户名和密码。

### 3.使用

使用是非常简单的，Yii2的queue组件非常棒。

（注意：下面的方式仅限于MQ的生产方和消费方都是fecshop（Yii2）， 因为传递的是对象，如果消费方不是fecshop，那么，消费方是没有传递的对象的定义，因此是不行的）

3.1 定义一个执行类，您想要做xx事情，就在下面execute($queue)方法中，譬如：

<?php

namespace fecshop\app\console\modules\Amqp\block;

use yii\base\Object;

class PushTest extends Object implements \zhuravljov\yii\queue\Job

{

public $name;

public $age;

# 这里是具体要做的事情。

public function execute($queue)

{

echo 'name:'.$this->name.'###'.'age:'.$this->age;

}

}

这是一个对象，里面有相应的值，这个对象必须实现execute($queue) 方法。

里面的内容，您可以做输出，也可以做调用。总之，您想要做的事情都在 execute($queue)方法中

3.2 生产方

<?php

/\*\*

\* FecShop file.

\*

\* @link http://www.fecshop.com/

\* @copyright Copyright (c) 2016 FecShop Software LLC

\* @license http://www.fecshop.com/license/

\*/

namespace fecshop\app\console\modules\Amqp\controllers;

use Yii;

use yii\console\Controller;

use fecshop\app\console\modules\Amqp\block\PushTest;

/\*\*

\* @author Terry Zhao <2358269014@qq.com>

\* @since 1.0

\*/

class TestController extends Controller

{

public function actionTest()

{

Yii::$app->queue->push(new PushTest([

'name' => 'terry',

'age' => 31,

]));

}

}

生产方的new PushTest() 就是上面我们定义的类。

3.3 执行

您可能纳闷，为什么不需要写消费方的代码？ 这里是不需要的，这个Yii2扩展做的非常棒，解耦的非常好， 因为生产方传递的是一个对象，消费方直接执行这个对象的execute()方法。 消费方就是下面的queue/listen

生产端命令，这个就是上面写的controller

./yii amqp/test/test

消费端命令

./yii queue/listen

在生产端执行后，就可以看到输出了

name:terry###age:31

### 4.应答机制

应答机制：当消费方接收解析，执行的时候，有可能会失败，这样就有疏漏， 为了防止这种情况，我们可以用应答机制， 消费方在连接初始化的时候，告诉需要开启应答，消费方获取消息后， MQ不会删除消息，等消费方接受消息执行成功后，应答机制告知MQ， MQ才会删除消息，这样，就避免消息执行失败。

默认是开启应答的

4.1消费方消费成功后，告诉RabbitMq，RabbitMq认为这个消息消费成功，才会删除消息。

4.2消费方执行时间很长，RabbitMq是没有超时时间的，因此不会把这个消息给其他的消费方消费

4.3如果程序卡住，kill进程，连接断开，MQ会认为消费失败，把消息发送给其他的消费者消费。

4.4上面的PushTest->execute($queue)，如果里面添加了exit()退出代码，这样会直接停掉，导致没有应答,MQ会重新发送给其他消费方。 ，因此这个函数里面是不能写入exit之类的代码的。

监听部分的代码分析：vendor\zhuravljov\yii2-queue\src\drivers\amqp\Queue.php

下面是listen部分的代码

/\*\*

\* Listens amqp-queue and runs new jobs.

\*/

public function listen()

{

$this->open();

$callback = function(AMQPMessage $message) {

# 此处调用执行。

# 执行$this->handleMessage，会调用PushTest->execute($queue)，

# 执行完成后，返回true

if ($this->handleMessage(null, $message->body)) {

# 下面是应答代码，执行后，告诉MQ,消费完成

# 注意，Rabbit是没有超时概念的，如果上面函数一直执行，没有完成，那么这个未完成的消费不会分给其他的消费方

# 消费完成，进行应答

$message->delivery\_info['channel']->basic\_ack($message->delivery\_info['delivery\_tag']);

}

};

# 多个消费方负载方面的设置。

$this->channel->basic\_qos(null, 1, null);

# 第三个参数 no\_ack 的 值为false代表，需要应答，true代表不需要应答。

$this->channel->basic\_consume($this->queueName, '', false, false, false, false, $callback);

while(count($this->channel->callbacks)) {

$this->channel->wait();

}

}

fecshop有一个测试的例子，你可以在两个终端执行：

生产方：./yii amqp/test/test

消费方：./yii queue/listen

### 5.功能使用

对于fecshop，目前还没有做MQ部分的功能，有一些地方是可以用的 ，譬如发邮件部分，用MQ发送邮件可以节省很多响应时间， 目前只是接纳到框架中，如果您有需求，可以使用MQ扩展或者重构。

### 6.和其他系统的MQ对接

有时候需要和其他的系统进行对接，譬如从erp MQ里面对接产品数据， MQ传递的是一个序列化，或者json格式的数据，那么，上面的方式就不好用了 ，我们可以用下面的方式，来写和外部系统MQ的对接。

6.1 生产方：

<?php

/\*\*

\* FecShop file.

\*

\* @link http://www.fecshop.com/

\* @copyright Copyright (c) 2016 FecShop Software LLC

\* @license http://www.fecshop.com/license/

\*/

namespace fecshop\app\console\modules\Amqp\controllers;

use Yii;

use yii\console\Controller;

use fecshop\app\console\modules\Amqp\block\PushTest;

/\*\*

\* @author Terry Zhao <2358269014@qq.com>

\* @since 1.0

\*/

class TestController extends Controller

{

protected $\_numPerPage = 50;

/\*\*

\* 得到当前的时间。

\*/

public function actionTest()

{

# 这个是对象的方式，消息的传递和接收都是fecshop的时候使用

//Yii::$app->queue->push(new PushTest([

// 'name' => 'terry',

// 'age' => 31,

//]));

# 这是一种比较随便的方式，发送的数组会以序列化的方式发送过去

# 由于传递的不是对象，接收方需要进行处理。

Yii::$app->queue->push([

'name' => 'water',

'age' => 331,

]);

}

}

通过上面的push方法，传递一个序列化的数组给MQ。

6.2 消费方

如果其他的系统生产数据给MQ（序列化数组格式），我们消费，

定义queue组件：

'components' => [

'queue' => [

//'class' => \zhuravljov\yii\queue\amqp\Queue::class,

//'class' => 'zhuravljov\yii\queue\amqp\Queue',

'class' => 'fecshop\app\console\modules\Amqp\block\Queue',

'host' => 'localhost',

'port' => 5672,

'user' => 'mqadmin',

'password' => 'mqadmin20177',

'queueName' => 'queue',

],

],

实现上面的class

<?php

namespace fecshop\app\console\modules\Amqp\block;

use yii\base\Object;

class Queue extends \zhuravljov\yii\queue\amqp\Queue

{

/\*\*

\* @inheritdoc

\*/

protected function handleMessage($id, $message)

{

$d = unserialize($message);

# 这里填写执行代码

// do some thing ...

\Yii::info('222','fecshop\_debug');

return true;

}

/\*

protected function handleMessage($id, $message)

{

if ($this->messageHandler) {

return call\_user\_func($this->messageHandler, $id, $message);

} else {

return parent::handleMessage($id, $message);

}

}

\*/

}

我们在新的组件中重写 handleMessage 方法，然后在里面处理接收的数据即可。

<?php

/\*\*

\* FecShop file.

\*

\* @link http://www.fecshop.com/

\* @copyright Copyright (c) 2016 FecShop Software LLC

\* @license http://www.fecshop.com/license/

\*/

namespace fecshop\app\console\modules\Amqp\controllers;

use Yii;

use yii\console\Controller;

use fecshop\app\console\modules\Amqp\block\PushTest;

/\*\*

\* @author Terry Zhao <2358269014@qq.com>

\* @since 1.0

\* 这是一个测试RabbitMq 的一个例子。这里作为消息生产方。

\* 你可以通过执行 ./yii amqp/test/test 来生产数据。

\*/

class TestController extends Controller

{

protected $\_numPerPage = 50;

/\*\*

\* 测试

\*/

public function actionTest()

{

// 这个是对象的方式，消息的传递和接收都是fecshop的时候使用

// Yii::$app->queue->push(new PushTest([

// 'name' => 'terry',

// 'age' => 31,

// ]));

// 这是一种比较随便的方式，发送的数组会以序列化的方式发送过去

// 传递的给MQ的个数格式为序列化数组。

Yii::$app->queue->push([

'name' => 'water',

'age' => 331,

]);

}

public function actionListen()

{

Yii::$app->queue->listen();

}

}

然后执行

生产： ./yii amqp/test/test

消费： ./yii amqp/test/listen

如果有多个MQ接收消息， 可以定义多个组件

'components' => [

'queue1' => [

],

'queue2' => [

],

'queue3' => [

],

],

每一个指定一个class即可。

# Fecshop Assets

assets 管理的是css js 以及css中出现的img等静态资源

### 原理

这个可以参看帖子：<http://www.fecshop.com/topic/432>

### 设置

对于后台appadmin，因为不会涉及到jscss的重构等，因此使用的是Yii2原声的asset。

而apphtml5 appfront涉及到多模板机制，因此fecshop进行了封装， 使用的是page asset services，打开文件

@appfront/config/fecshop\_local\_services/Page.php

就可以看到配置

'asset' => [

'class' => 'fecshop\services\page\Asset',

// 在js后面加一个v参数，修改js后，更改v参数，否则，浏览器会使用缓存。

// /assets/dbdba3fa/js/js.js?v=2

'jsVersion' => 1,

'cssVersion' => 1,

/\*\*

\* @var string the root directory string the published asset files.

\* 设置: js和css的发布路径

\* 譬如设置为：'@appimage/assets'，也可以将 @appimage 换成绝对路径

\*/

'basePath' => '@webroot/assets',

/\*\*

\* @var string the base URL through which the published asset files can be accessed.

\* 设置: js和css的URL路径

\* 可以将 @web 换成域名 ， 譬如 `http:://www/fecshop.com/assets`

\* 这样就可以将js和css文件使用独立的域名了【把域名对应的地址对应到$basePath】。

\*/

'baseUrl' => '@web/assets',

// 是否每次访问都强制复制css js img等文件到发布地址，true代表每次访问都发布

// 一般开发环境用true，线上用false。当线上更新jscss文件，可以清空assets发布路径下的文件的方式来更新

'forceCopy' => true,

],

如果你想对js css使用独立的域名， 那么，您可以设置如下：

'basePath' => '@appimage/assets', //如果没有这个路径请新建该路径

'baseUrl' => 'http://yourdomain.com/assets', // 将这个域名解析到@appimage

如果您想要使用cdn，那么，您可以把 @appimage/assets文件上传到cdn ，然后更改baseUrl做指向即可.

因为fecshop的assets是基于yii2的asssets，因此，您可以了解一下yii2 assets：<http://www.yiichina.com/doc/guide/2.0/structure-assets>

# Fecshop Error Handler

Error Handler指的是Yii2的错误捕捉机制，fecshop利用了 Yii2的error Handle机制来记录系统的异常。

### Fecshop重新处理errorHandler的原因

**本地开发环境：**

在开发环境中，web界面的报错信息直接报出在访问界面，一点问题也没有，对于api类型 ，在开发的过程中也可以直接报出，同样没有安全问题。

**线上环境:**

1.如果报错直接输出，会暴露网站的很多信息，这样是非常不安全的， 而如果使用yii2默认的PROD的errorHandler处理方式，只显示一句 An internal server error occurred.，用户反馈报错给开发者后， 开发者在日志中看到一堆的报错，不知道那个报错是这个用户的报错， 因此不利于开发者快速的定位问题。

2.比较好的方式，在报错的时候，系统在写入日志的时候，返回一个错误编码，

{

"code":500,

"error\_no":"5a2113f5bfb7ae617546c872"

}

3.用户访问页面报错后，将错误编码反馈给开发者，开发者 通过错误编码，到日志中找到该用户的报错信息，这样可以更方便的处理用户的 的错误。

4.另外，我们通过日志的方式，可以记录网站在运行期间出现的所有错误url， 有了这个信息，我们就可以依次排查各个错误的原因，进而将 网站出现的问题都清楚掉，让网站更加稳健。

### Error Handler

打开配置文件 @fecshop/app/app入口/config/appserver.php

app入口 代指:appfront, appserver, apphtml5, appapi]

你可以看到errorHandler的配置，譬如：@fecshop/app/appfront/config/appserver.php

// product 生产环境，errorHandler使用 AppfrontErrorHandler

if (YII\_ENV\_PROD) {

$config['components']['errorHandler']['class'] = 'fecshop\components\AppfrontErrorHandler';

}

上面指的是，当环境为PROD时（在入口index.php中设置YII\_ENV）， errorHandler 将使用配置中的class，下面我们看一下这个class

打开文件@fecshop/components/AppfrontErrorHandler.php

<?php

namespace fecshop\components;

use Yii;

use yii\web\ErrorHandler;

use yii\base\Exception;

/\*\*

\* 异常捕获器

\*/

class AppfrontErrorHandler extends ErrorHandler

{

/\*\*

\* [renderException description]

\* @property $exception | Object 异常数据对象

\*

\*/

public function renderException($exception)

{

//echo 1;exit;

// 获取异常数据 , 404页面不做收集

$code = $exception->statusCode ?: 500;

if ($code != 404) {

method\_exists($exception,'getMessage') ? $message = $exception->getMessage() : $message = '';

method\_exists($exception,'getName') ? $name = $exception->getName() : $name = '';

method\_exists($exception,'getFile') ? $file = $exception->getFile() : $file = '';

method\_exists($exception,'getLine') ? $line = $exception->getLine() : $line = '';

method\_exists($exception,'getTraceAsString') ? $traceString = $exception->getTraceAsString() : $traceString = '';

$time = time();

$ip = Yii::$app->request->userIP;

$url = Yii::$service->url->getCurrentUrl();

$req\_info = $this->getRequestInfo();

$reponse = Yii::$app->response;

Yii::$app->response->format = $reponse::FORMAT\_JSON;

if (YII\_ENV\_PROD) {

$errorKey = $this->saveProdException($code, $message, $file, $line, $time, $ip, $name, $traceString, $url, $req\_info);

Yii::$app->response->data = [

'code' => $code,

'error\_no' => $errorKey,

];

Yii::$app->response->send();

Yii::$app->end();

} else {

$time = date('Y-m-d H:i:s', $time);

$exceptionInfo = [

'code' => $code,

'message' => $message,

'file' => $file,

'line' => $line,

'time' => $time,

'ip' => $ip,

'name' => $name,

'traceString' => $traceString,

];

Yii::$app->response->data = $exceptionInfo;

Yii::$app->response->send();

Yii::$app->end();

}

} else {

parent::renderException($exception);

}

}

public function getRequestInfo(){

$request = Yii::$app->request;

$ajax = 0;

$request\_type = '';

$request\_data = [];

$header\_accept = '';

$header\_user\_agent = '';

if ($request->isAjax) {

$ajax = 1;

}

if ($request->isGet) {

$request\_type = 'get';

$request\_data = $request->get();

}

if ($request->isPost) {

$request\_type = 'post';

$request\_data = $request->post();

}

// $headers is an object of yii\web\HeaderCollection

$headers = $request->getHeaders();

$headers\_arr = [];

if (is\_object($headers) or is\_array($headers)) {

foreach ($headers as $k=>$v) {

$headers\_arr[$k] = $v;

}

}

$userHost = Yii::$app->request->userHost;

$userIP = Yii::$app->request->userIP;

return [

'ajax' => $ajax,

'request\_type' => $request\_type,

'request\_data' => $request\_data,

'headers\_data' => $headers\_arr,

'userHost' => $userHost,

'userIP' => $userIP,

];

}

public function saveProdException($code, $message, $file, $line, $created\_at, $ip, $name, $trace\_string, $url, $req\_info){

return Yii::$service->helper->errorHandler->saveByErrorHandler(

$code, $message, $file, $line, $created\_at,

$ip, $name, $trace\_string, $url, $req\_info

);

}

/\*\*

\* 这块代码目前没有编写，您也可以用 Sentry（错误日志收集框架） 来收集错误日志。

\* $option = [

\* 'fromMail' => 'xxx@xxx.com',

\* 'subject' => 'fecshop报错 Code:' . $exceptionInfo['code'],

\* 'htmlBody' => '异常' . '(' . YII\_ENV . ')' . $exceptionInfo['code'] . ':' . $exceptionInfo['message'] . '<br />文件：' . $exceptionInfo['file'] . ':' . $exceptionInfo['line'] . '<br /> 时间：' . $exceptionInfo['time'] . '<br />请求ip：' . $exceptionInfo['ip'],

\* ];

\*

\*

\*if (!$result) {

\* throw new Exception("Exception mail send faild!", 1);

\*}

\*/

}

通过上面的代码可以看到，错误写到了mongodb表：error\_handler\_log 中， 该表数据中的\_id，就是反馈给用户的 "error\_no":"5a2113f5bfb7ae617546c872"

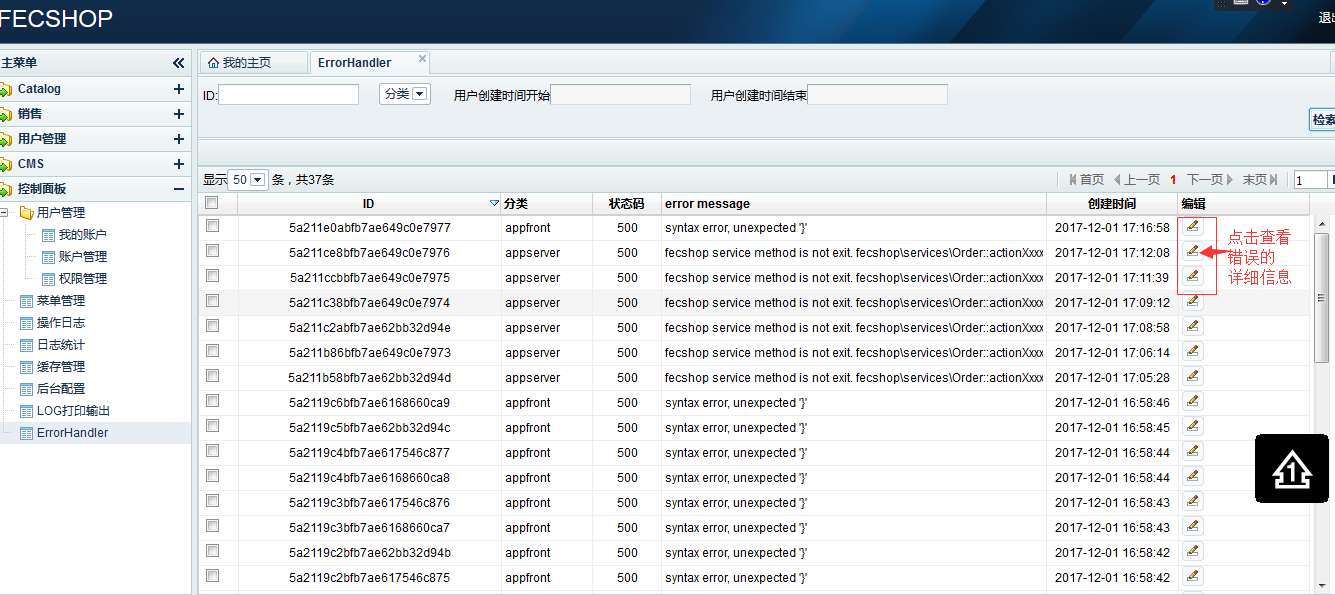
{

"code":500,

"error\_no":"5a2113f5bfb7ae617546c872"

}

保存的error信息，可以在后台 控制面板-->ErrorHanele 中查看



将 error\_no 粘贴到ID处，即可进行查询相应的错误信息。

### 总结：

1.如果是开发环境，使用的是Yii2默认的error处理方式，直接报错输出

2.如果是PROD生产环境，appfront apphtml5这种web类的入口， 404错误还是用的Yii2默认的error处理方式， 因此，会出现一个404的WEB页面，告诉用户该页面不存在。 对于appapi，appserver接口类入口的404错误 ，譬如：appapi，appserver 404错误 将使用 @fecshop/components/AppserverErrorHandler.php 来处理。

3.如果是生产环境，其他的错误将由fecshop的errorHandle处理

3.1 apphtml5 appfront的errorHandler使用的是 @fecshop/components/AppfrontErrorHandler.php

3.2 appserver appapi的errorHandler使用的是 @fecshop/components/AppserverErrorHandler.php

4.如果使用了fecshop的errorhandler，只返回code和errorKey给用户，其他的信息通过函数 saveProdException()保存到mongodb中。

5.用户将errorKey的值发给开发者，开发者可以通过 errorKey 在mongodb表 error\_handler\_log中查询到具体的错误

6.这样做的好处是，让线上环境将具体的错误隐藏起来，还可以通过反馈给用户的 错误编号找到相应的错误，让开发者快速定位当前用户的问题。

7.将开发环境设置成生产环境的方式：

在@app/web/index.php中，将

defined('YII\_DEBUG') or define('YII\_DEBUG', true);

defined('YII\_ENV') or define('YII\_ENV', 'dev');

设置成：

defined('YII\_DEBUG') or define('YII\_DEBUG', false);

defined('YII\_ENV') or define('YII\_ENV', 'prod');

即可切换到生产环境

8.mongodb表error\_handler\_log中保存的错误信息数据示例:

{

"\_id": ObjectId("5a211e0abfb7ae649c0e7977"),

"category": "appfront",

"code": NumberInt(500),

"message": "syntax error, unexpected '}'",

"file": "/www/web/develop/fecshop/vendor/fancyecommerce/fecshop/app/appfront/modules/Customer/controllers/TestController.php",

"line": NumberInt(31),

"created\_at": NumberInt(1512119818),

"ip": "113.116.124.39",

"name": "",

"url": "//fecshop.appfront.fancyecommerce.com/customer/test?fd=3232&323=232",

"request\_info": {

"ajax": NumberInt(0),

"request\_type": "get",

"request\_data": {

"fd": "3232",

"323": "232"

},

"headers\_data": {

"cache-control": [

"max-age=0"

],

"upgrade-insecure-requests": [

"1"

],

"connection": [

"keep-alive"

],

"cookie": [

"Hm\_lvt\_9cfdb8cb93410e80005df709d26cd1c6=1511760319,1511796767,1511797236,1511838720; \_ga=GA1.2.395760026.1496285492; PHPSESSID=mqk98mldqgqficiksoqb69h8ui; \_csrf=d59b458176e323d701f64264dbb8fe770ba8e30c26a799002cd5aa4af50db072a%3A2%3A%7Bi%3A0%3Bs%3A5%3A%22\_csrf%22%3Bi%3A1%3Bs%3A32%3A%22%89%C6%E9zkg%16%28%0A%2A%E4u%83%F3%FD%8B%80A%87%89%99%FC%24%EE%09Y%5B2%FA%82%94B%22%3B%7D"

],

"accept-encoding": [

"gzip, deflate, br"

],

"accept-language": [

"zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2"

],

"accept": [

"text/html,application/xhtml+xml,application/xml;q=0.9,\*/\*;q=0.8"

],

"user-agent": [

"Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:57.0) Gecko/20100101 Firefox/57.0"

],

"host": [

"fecshop.appfront.fancyecommerce.com"

]

},

"userHost": null,

"userIP": "113.116.124.39"

},

"trace\_string": "#0 /www/web/develop/fecshop/vendor/composer/ClassLoader.php(301): Composer\\Autoload\\includeFile('/www/web/develo...')\n#1 [internal function]: Composer\\Autoload\\ClassLoader->loadClass('fecshop\\\\app\\\\app...')\n#2 [internal function]: spl\_autoload\_call('fecshop\\\\app\\\\app...')\n#3 /www/web/develop/fecshop/vendor/yiisoft/yii2/base/Module.php(633): class\_exists('fecshop\\\\app\\\\app...')\n#4 /www/web/develop/fecshop/vendor/yiisoft/yii2/base/Module.php(591): yii\\base\\Module->createControllerByID('test')\n#5 /www/web/develop/fecshop/vendor/yiisoft/yii2/base/Module.php(583): yii\\base\\Module->createController('')\n#6 /www/web/develop/fecshop/vendor/yiisoft/yii2/base/Module.php(517): yii\\base\\Module->createController('test')\n#7 /www/web/develop/fecshop/vendor/yiisoft/yii2/web/Application.php(102): yii\\base\\Module->runAction('customer/test', Array)\n#8 /www/web/develop/fecshop/vendor/yiisoft/yii2/base/Application.php(380): yii\\web\\Application->handleRequest(Object(fecshop\\yii\\web\\Request))\n#9 /www/web/develop/fecshop/appfront/web/index.php(86): yii\\base\\Application->run()\n#10 {main}"

}

9.重写errorHandle，按照自己的方式处理

您可以在本地配置中添加配置，来覆盖fecshop默认的配置。

$config['components']['errorHandler']['class'] = '您自己的errorHandler处理class';

# Fecshop breadcrumbs

Fecshop breadcrumbs指的是面包屑导航，也就是在分类， 产品页面的菜单栏下面，显示当前的路径，譬如： Home > women dress > summer lady beautiful red dress

### 配置：

配置文件是：

@fecshop/config/services/Page.php

1.您可以在 @app/config/fecshop\_local\_services/Page.php 添加 配置，覆盖掉fecshop默认的配置.

2.homeName: 默认是Home，如果为空，则在面包屑导航中不显示。

ifAddHomeUrl:代表首页是否添加链接，false代表是文本，ture代表添加 上首页的链接

### 使用

1.面包屑导航使用，您可以通过下面的代码添加项到breadcrumbs中

Yii::$service->page->breadcrumbs->addItems($items)

源码：您可以在@fecshop\services\page\Breadcrumbs.php里面找到 找到看到如下代码：

/\*\*

\* property $items|Array. add $items to $this->\_items.

\* $items format example. 将各个部分的链接加入到面包屑导航中

\* $items = ['name'=>'fashion handbag','url'=>'http://www.xxx.com'];.

\*/

protected function actionAddItems($items)

{

if ($this->active) {

$this->\_items[] = $items;

}

}

2.将添加到breadcumbs中的子项调用出来

Yii::$service->page->breadcrumbs->getItems();

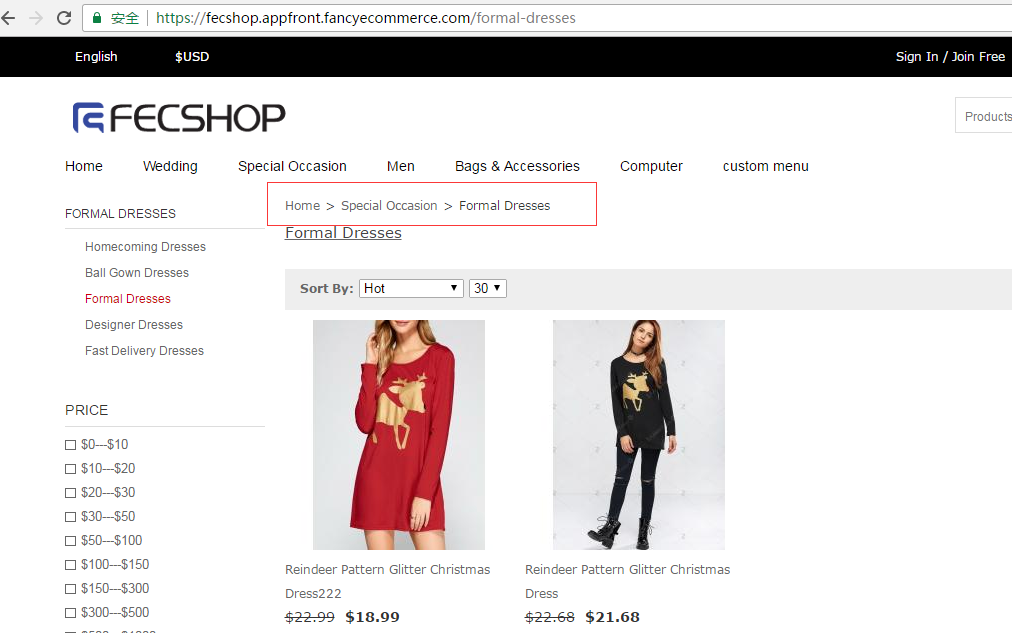
### 例子分析：

1.@appfront-category-breadcrumbs 部分：

如果在appfront 入口，您想打开分类页面部分的breadcrumbs ，您可以在@appfront/config/fecshop\_local\_modules/Catalog.php 设置

'category\_breadcrumbs' => true,

设置完，就可以看到分类页面的面包屑导航出来了



2.产品页面没有实现，您可以在 fecshop\app\appfront\modules\Catalog\block\product\Index.php 中自己实现，或者，过段时间我写一下这个。

面包屑导航对seo有 一定效果，不过也会弄的页面比较丑， 另外，手机web端，面包屑导航已经不要了，取而代之的就是一个回退箭头。

看自己取舍了。

# 上线前的配置

当您配置好线上环境，网站可以访问， 产品，分类等都已经上传，那么就可以开始上线了， 上线前需要进行一系列的线上配置。

### 1.邮箱配置

1.1smtp配置

您需要设置您的邮箱smtp信息，比较牛的就是sendgrid了，当然 也有一些其他的邮件商，如果订单量少，可以用 google等免费邮箱的smtp服务，不过发送量有限制

1.2邮件模板的配置

如果您想个性化您的邮件模板，那么您需要修改您的模板内容

邮件配置的详细参看 [fecshop 邮件设置](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_email.html)

### 2.facebook google登录

首先需要申请相应的key ： [facebook login 申请 app\_id 和 app\_secret](http://blog.csdn.net/terry_water/article/details/55095721) ， [google login api 申请 CLIENT\_SECRET 和 CLIENT\_SECRET](http://blog.csdn.net/terry_water/article/details/55095209)

申请到了就是配置， 您可以在@appfront/config/fecshop\_local\_services/Store.php中配置

### 3.货运信息

配置各个货运方式，以及相应的运费，详细参看： [Fecshop 货运方式](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_shipping_method.html)

#### 4.支付信息

配置paypal的信息，详细参看：[Fecshop 支付方式](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_payment_method.html)

需要注意的是：paypal的地址和账号一定需要是线上的账户，而不是沙盒url和沙盒账户， 这个一定要看好。

### 5.设置cron

具体参看[fecshop cron](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_cron_script.html)

### 6.是否开启注册登录等验证码

具体参看[Fecshop 验证码](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_yzm.html)

### 7.设置语言和货币 以及相应的文件翻译

一般是一个域名一个语言，[Fecshop 多语言](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_mutil_lang.html)

[Fecshop 货币](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_currency.html)

### 8.编辑各个page页和static block

[Fecshop page页](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_page.html)

[Fecshop 静态块](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_static_block.html)

### 9.缓存开启

[Fecshop 缓存](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_cache.html)

### 10. https vls加密方式

google 对https的网站会有加分，建议网站使用https的方式 ，nginx配置加密证书的配置可以参考： [centos 下安装 Let’s Encrypt 永久免费 SSL 证书](http://www.fancyecommerce.com/2017/04/07/centos-%e4%b8%8b%e5%ae%89%e8%a3%85-lets-encrypt-%e6%b0%b8%e4%b9%85%e5%85%8d%e8%b4%b9-ssl-%e8%af%81%e4%b9%a6/)

### 11. 配置加速

这个非必须，网站流量如果比较大，才建议使用 ，如果想开启，可以参考：[fecshop 配置加速](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_config_speed.html)

### 12. 设置

将logo图片换成您自己的logo，在

### 13.最后

最后多测试，多看，多观察哪里有问题，新站上线经常会出问题

# Fecshop 安全问题

### 1.查看端口开放：

netstat -tln

譬如我的检测为：

[root@ip-172-31-30-214 ~]# netstat -tln

Active Internet connections (only servers)

Proto Recv-Q Send-Q Local Address Foreign Address State

tcp 0 0 127.0.0.1:9000 0.0.0.0:\* LISTEN

tcp 0 0 0.0.0.0:25672 0.0.0.0:\* LISTEN

tcp 0 0 127.0.0.1:27017 0.0.0.0:\* LISTEN

tcp 0 0 172.31.30.214:3690 0.0.0.0:\* LISTEN

tcp 0 0 127.0.0.1:6379 0.0.0.0:\* LISTEN

tcp 0 0 0.0.0.0:80 0.0.0.0:\* LISTEN

tcp 0 0 0.0.0.0:4369 0.0.0.0:\* LISTEN

tcp 0 0 0.0.0.0:22 0.0.0.0:\* LISTEN

tcp 0 0 0.0.0.0:15672 0.0.0.0:\* LISTEN

tcp 0 0 127.0.0.1:25 0.0.0.0:\* LISTEN

tcp 0 0 :::5672 :::\* LISTEN

tcp 0 0 :::3306 :::\* LISTEN

tcp 0 0 :::4369 :::\* LISTEN

tcp 0 0 :::22 :::\* LISTEN

tcp 0 0 ::1:25 :::\* LISTEN

对于外网ip开放（也就是172.31.30.214）的端口要注意安全，密码一定要很长 提高安全性， 对于mongodb（默认端口27017）和redis（默认端口6379）绑定的ip为内网127.0.0.1， 如果开放端口为外网IP，那么，mongodb和redis必须设置密码，否则将会被入侵。

### 2.mongodb的安全设置

mongodb的安装可以参考：[http://www.fancyecommerce.com/2016/05/03/yii2-mongodb的安装和配置-mongo/](http://www.fancyecommerce.com/2016/05/03/yii2-mongodb%E7%9A%84%E5%AE%89%E8%A3%85%E5%92%8C%E9%85%8D%E7%BD%AE-mongo/)

#### 1.1 设置端口和ip

安全ip连接设置：vim /etc/mongod.conf

net:

port: 27017

bindIp: 127.0.0.1

port代表开放的访问端口，bindIp这个是一个安全ip配置，允许那些ip连接mongodb，如果设置成bind\_ip = 0.0.0.0，代表任意ip，建议只写连接的ip，如果是本机，就填写127.0.0.1即可。

这个是一个安全设置，线上系统一定要注意，iptables设置，只允许部分ip访问该端口，其他的ip不允许访问该port，而且mongodb也要设置允许的ip。

#### 2.2 mongodb账号密码

另外，对于线上系统，最好是使用账号密码的方式连接，进一步增强安全性。 这个您自行查阅资料，现在还没有写相关文章。

### 3.Yii给予的安全防范

[最佳安全实践](http://www.yiichina.com/doc/guide/2.0/security-best-practices)

3.1 域名指向@app/web目录，这样可以防止暴露其他文件

3.2 将develop（开发）模式改成product（生产）模式

打开 @app/web/index.php(@app代表 @appfront @apphtml5 等入口)

将代码：

defined('YII\_DEBUG') or define('YII\_DEBUG', true);

defined('YII\_ENV') or define('YII\_ENV', 'dev');

改成

defined('YII\_DEBUG') or define('YII\_DEBUG', false);

defined('YII\_ENV') or define('YII\_ENV', 'prod');

打开后，线上环境的错误信息不回直接报出，只会给予一个错误编号，譬如：

{"code":500,"error\_no":"5a3b0e4d3e00ca3c3f3aaf52"}

然后可以在后台查看具体的报错信息，详细参看： [Fecshop Error Handler](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_error_handler.html)

这样，用户如果有错误，直接把错误编号发过来，开发人员就可以查到相应的错误， 另外，开发人员可以自己去查看后台的ErrorHandle，处理没有发现的错误问题。

3.3 安全加密访问(HTTPS)

现在firefox，chrome，都把http连接默认设置为不安全的连接， 因为是明码传输，而对于https是加密传输，会更安全一些， https的安装和使用可以参看文章：

[centos 下安装 Let’s Encrypt 永久免费 SSL 证书](http://www.fancyecommerce.com/2017/04/07/centos-%e4%b8%8b%e5%ae%89%e8%a3%85-lets-encrypt-%e6%b0%b8%e4%b9%85%e5%85%8d%e8%b4%b9-ssl-%e8%af%81%e4%b9%a6/)

3.4 xss攻击

对于xss 攻击，fecshop默认都已经过滤掉，如果您开发了新功能，一定要注意 处理。

### 4. 密码设置

4.1 后台

后台密码，一定要尽量的长，后台使用其他域名，尽量把后台访问路径保护起来。

4.2 其他密码

对于mysql linux密码等要设置的比较长，防止安全破解。

### 5.cron定时脚本

对于console的脚本知识，你可以参看：[Fecshop console 介绍和配置](http://www.fecshop.com/doc/fecshop-guide/develop/cn-1.0/guide-fecshop-console-about.html)

对于周期性在cron中执行的脚本，要输出log日志，这样可以在出问题的时候查看log。

### 5. 其他安全问题

等想到了其他的安全问题，会继续添加注意防范的地方

如果您有自己对安全的一些见解，可以联系我，qq：2358269014

# Fecshop 性能问题

### 硬件配置

硬件配置当然是越快越好，内存越大越好，这个没得说哈，内存一定要够用。

内存够大，cpu够多，够快

数据库硬盘尽量用SSD，加快读写

### 环境配置

1.php配置，php细节配置参数很多，有很多根据服务器情况的设置和优化

[PHP-FPM进程数的设定](http://www.fecshop.com/topic/459)

开启Opcache，参考：[php 安装 zend opcache](http://www.fancyecommerce.com/2016/09/24/php-%E5%AE%89%E8%A3%85-zend-opcace/)

使用php7，最近新出的php7.2引入的JIT， 比php7.1提升了10%的性能，

php7的优化：[让PHP7达到最高性能的几个Tips](http://www.laruence.com/2015/12/04/3086.html)

等等

2.Mysql配置优化

mysql 的配置文件参数优化

3.Mongodb配置优化

资料：[关于php mongodb的最大连接数](http://www.fecshop.com/topic/445)

注：上面的配置参数的优化，首先要了解相应的配置参数的含义， 然后根据具体情况做优化配置，譬如，机器内存36G，那么可以把 php的进程数设置多一些。

### Fecshop生产环境配置优化

1.asset设置

1.1关闭forceCopy

对于需要css的端口 appfront 和apphtml5，打开 @app/config/fecshop\_local\_services/Page.php ，设置：

'forceCopy' => false,

关闭后，如果更新js和css信息，您需要去@app/web/assets 下面手动清空 ``assets文件，清空后，Yii2的下次访问会自动复制最近的js和css文件

当然，如果你的网站访问量不大，开启也是可以的，更新js和css会方便一些，不需要去@app/web/assets 下面手动清空

1.2 更新js和css后，更新版本号

更新版本号的方式参看[Fecshop theme js and css](http://www.fecshop.com/doc/fecshop-guide/develop/cn-1.0/guide-fecshop-js-css.html#jscss) , 这篇文章的 js和css 浏览器缓存和版本 部分

这样js和css更新后，会强制浏览器加载最新的js和css文件， 而不是去使用浏览器缓存。

1.3 使用cdn

参看[Fecshop theme js and css](http://www.fecshop.com/doc/fecshop-guide/develop/cn-1.0/guide-fecshop-js-css.html#jscss)

2.config优化，开启单文件配置

fecshop的配置最终是由N个配置php文件合并而成， 在每次初始化 前执行，为了加速，可以先把配置文件合并成单文件， 然后加载这个合并成的单配置文件，就会比较节省资源，加快速度。

设置：在入口文件@app/web/index.php 代码： $use\_merge\_config\_file = false; 处设置。 true 代表打开，false代表关闭

打开配置加速开关前，您需要先生成单配置文件。 您需要执行 <http://www.domain.com/index-merge-config.php> 进行生成单文件配置数组。 注意：打开后，当您修改了配置， 都需要重新生成单文件配置数组，否则修改的配置不会生效,。

建议：

1.本地开发环境关闭，开发环境如果访问量不大，关闭也行，

如果访问量大，建议打开

2.http://www.domain.com/index-merge-config.php是访问执行，因此，

我们希望只能由我们自己人执行，因此，

2.1.您可以把这个`index-merge-config.php`文件的名字改一下，改成 `fdad78fds7fds7fds8.php` ，只有自己人

知道的文件路径

2.2在nginx上面对这个文件访问做一下密码控制，详细参看：

[nginx 某个目录访问设置账户密码访问](http://www.fecshop.com/topic/407)

3.开启缓存

fecshop的缓存分为局部缓存和全页缓存，使用的是yii2的机制，

开启缓存可以加速，详细参看： [fecshop cache](http://www.fecshop.com/doc/fecshop-guide/instructions/cn-1.0/guide-fecshop_cache.html)

4.浏览器加载加速

fecshop已经遵循下面的优化方式，如果您开发了新的模块，建议也这样，加快浏览器页面 的渲染。

4.1 js部分

fecshop的css放到html页面的头部，js放到页面的底部，这样可以让浏览器 快速加载

4.2 js和css 链接url用独立的域名 ，图片链接url用独立的域名，使用独立域名的好处，可以参看文章： [网站的图片，css，js 为什么要和网站的域名不一样](http://www.fancyecommerce.com/2017/04/17/%E7%BD%91%E7%AB%99%E7%9A%84%E5%9B%BE%E7%89%87%EF%BC%8Ccss%EF%BC%8Cjs-%E4%B8%BA%E4%BB%80%E4%B9%88%E8%A6%81%E5%92%8C%E7%BD%91%E7%AB%99%E7%9A%84%E5%9F%9F%E5%90%8D%E4%B8%8D%E4%B8%80%E6%A0%B7/)

5 扩展优化

可以使用elasticSearch，用于产品，分类，搜索页查询， 目前fecshop不支持用elasticSearch，可以自己扩展。

6 mysql mongodb 设置慢查询日志，把慢查询的语句找出来，定位到功能， 然后进行处理

7 nginx访问日志设置开始和结束时间，把间隔时间比较长的url找出来

8.services部分也可以开启日志，不过消耗比较大，可以短时间的开启， 查看各个services的各个方法执行消耗的时间。 具体实现可以参看文件 @fecshop/services/Service.php

9.如果访问量比较大，

9.1 数据库：可以把mongodb做复制集，mysql做主从复制

9.2 图片：图片可以放到云库，如果图片放到自己的服务器，可以通过NFS，将 图片服务器的磁盘挂载到php服务器的图片路径下，然后把域名指向图片服务器。

9.3 加几个php机器，nginx做负载均衡

9.4 缓存和session 放到redis服务器里面

这样各个层都是单独的服务器，分布式拆开，性能会提升很多。

最后，如果您开发了新模块，可以参看一下[Yii2的性能优化建议](http://www.yiichina.com/doc/guide/2.0/tutorial-performance-tuning)

### 采用fecshop官方插件提速

[Fecshop 官方扩展列表](http://www.fecshop.com/doc/fecshop-guide/develop/cn-1.0/guide-fecshop-pkg-list.html)

### 其他

等等，如果你有新的优化建议，可以在fecshop.com发帖。

# Fecshop 安装视频

fecshop对技术初学者的安装有一定的难度，因此做了一下安装视频 ,fecshop安装视频都已经撸完，并已经全部上传完成，优酷视频质量差一些，高清视频可以去百度网盘下载。

### 1. 虚拟Linux环境

1.1 Vagrant + Virtual box 安装linux环境

优酷观看地址： [1.vagrant搭建linux虚拟机](http://v.youku.com/v_show/id_XMjg0NzAwMTM2MA==.html?spm=a2h3j.8428770.3416059.1)

百度云盘地址： [1.vagrant搭建linux虚拟机.avi](http://pan.baidu.com/s/1skUvhU1#list/path=%2F%E8%A7%86%E9%A2%91%2F%E7%8E%AF%E5%A2%83%E6%90%AD%E5%BB%BA)

对于该视频中出现的文档， 在fecshop文档中已经做了更改，这个地址找不到了

如果想查看文档，可以参看： [Fecshop vagrant安装](http://www.fecshop.com/doc/fecshop-guide/develop/cn-1.0/guide-fecshop-about-vagrantinstall.html)

### 2.在linux上面安装相应的软件

下面安装的软件，可以去官网下载，也可以去我的百度云盘下载， 下载地址为：[http://pan.baidu.com/s/1b63eXo#list/path=/](http://pan.baidu.com/s/1b63eXo#list/path=%2F)

2.1 安装mysql

优酷观看地址：[2.Linux下安装mysql.avi](http://v.youku.com/v_show/id_XMjg0NzU0NTUwNA==.html?spm=a2h3j.8428770.3416059.1)

百度云盘地址：[2.Linux下安装mysql.avi](http://pan.baidu.com/s/1skUvhU1#list/path=%2F%E8%A7%86%E9%A2%91%2F%E7%8E%AF%E5%A2%83%E6%90%AD%E5%BB%BA)

2.2 安装php.avi

优酷观看地址：[3.安装php.avi](http://v.youku.com/v_show/id_XMjg1NDMyMTMzMg==.html?spm=a2h3j.8428770.3416059.1)

百度云盘地址：[3.安装php.avi](http://pan.baidu.com/s/1skUvhU1#list/path=%2F%E8%A7%86%E9%A2%91%2F%E7%8E%AF%E5%A2%83%E6%90%AD%E5%BB%BA)

2.3 安装nginx和redis

优酷观看地址：[4.安装nginx和redis.avi](http://v.youku.com/v_show/id_XMjg1MzM5MzA2OA==.html?spm=a2h3j.8428770.3416059.1)

百度云盘地址：[4.安装nginx和redis.avi](http://pan.baidu.com/s/1skUvhU1#list/path=%2F%E8%A7%86%E9%A2%91%2F%E7%8E%AF%E5%A2%83%E6%90%AD%E5%BB%BA)

2.4 安装php的扩展（里面含有安装mongodb的部分）

优酷观看地址：[5.安装php的扩展和安装mongodb.avi](http://v.youku.com/v_show/id_XMjg1MzM5MjkwNA==.html?spm=a2h3j.8428770.3416059.1)

百度云盘地址：[5.安装php的扩展和安装mongodb.avi](http://pan.baidu.com/s/1skUvhU1#list/path=%2F%E8%A7%86%E9%A2%91%2F%E7%8E%AF%E5%A2%83%E6%90%AD%E5%BB%BA)

2.5 composer安装fecshop

优酷观看地址：[6.composer安装fecshop.avi](http://v.youku.com/v_show/id_XMjg1MzM5MjY0NA==.html?spm=a2h3j.8428770.3416059.1)

百度云盘地址：[6.composer安装fecshop.avi](http://pan.baidu.com/s/1skUvhU1#list/path=%2F%E8%A7%86%E9%A2%91%2F%E7%8E%AF%E5%A2%83%E6%90%AD%E5%BB%BA)

2.6 fecshop的配置

优酷观看地址：[7.fecshop的配置.avi](http://v.youku.com/v_show/id_XMjg1MzM5MjQ0MA==.html?spm=a2h3j.8428770.3416059.1)

百度云盘地址：[7.fecshop的配置.avi](http://pan.baidu.com/s/1skUvhU1#list/path=%2F%E8%A7%86%E9%A2%91%2F%E7%8E%AF%E5%A2%83%E6%90%AD%E5%BB%BA)

# Fecshop 开发视频

fecshop 在开放方面的一些视频的介绍，排队等待录制...

### 1. Fecsop 架构结构和文件结构介绍

优酷观看地址： [Fecsop 架构结构和文件结构介绍](http://v.youku.com/v_show/id_XMjg4MTYyMjE2MA==.html?spm=a2h3j.8428770.3416059.1)

百度云盘地址： [Fecsop 架构结构和文件结构介绍](https://pan.baidu.com/s/1skUvhU1#list/path=%2F%E8%A7%86%E9%A2%91%2F%E5%BC%80%E5%8F%91%E8%A7%86%E9%A2%91)

### 2. fecshop 配置结构和功能初始化

优酷观看地址： [fecshop 配置结构和功能初始化](http://v.youku.com/v_show/id_XMjg4MTYyMzMxNg==.html?spm=a2h3j.8428770.3416059.1)

百度云盘地址： [fecshop 配置结构和功能初始化](https://pan.baidu.com/s/1skUvhU1#list/path=%2F%E8%A7%86%E9%A2%91%2F%E5%BC%80%E5%8F%91%E8%A7%86%E9%A2%91)

### 2. fecshop服务和二次开发

优酷观看地址： [fecshop服务和二次开发](http://v.youku.com/v_show/id_XMjg4MTYyNDAyOA==.html?spm=a2h3j.8428770.3416059.1)

百度云盘地址： [fecshop服务和二次开发](https://pan.baidu.com/s/1skUvhU1#list/path=%2F%E8%A7%86%E9%A2%91%2F%E5%BC%80%E5%8F%91%E8%A7%86%E9%A2%91)

# Fecshop 帮助视频

fecshop 帮助文档部分的视频，排队等待录制...

### 1. fecshop帮助视频1

优酷观看地址： [fecshop帮助视频1](http://v.youku.com/v_show/id_XMjg4MTYxOTQ2OA==.html?spm=a2h3j.8428770.3416059.1)

百度云盘地址： [fecshop帮助视频1](https://pan.baidu.com/s/1skUvhU1#list/path=%2F%E8%A7%86%E9%A2%91%2F%E5%B8%AE%E5%8A%A9%E8%A7%86%E9%A2%91&parentPath=%2F)

### 1. fecshop帮助视频2

优酷观看地址： [fecshop帮助视频2](http://v.youku.com/v_show/id_XMjg4MTYyMTMyOA==.html?spm=a2h3j.8428770.3416059.1)

百度云盘地址： [fecshop帮助视频2](https://pan.baidu.com/s/1skUvhU1#list/path=%2F%E8%A7%86%E9%A2%91%2F%E5%B8%AE%E5%8A%A9%E8%A7%86%E9%A2%91&parentPath=%2F)