Fecshop 介绍

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'])) {

return;

}

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

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

if (!$enable) {

return;

}

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

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

if (is\_array($condition) && !empty($condition) && !empty($redirectDomain)) {

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

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

if ($mobileDetect->isMobile()) {

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

}

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

if ($mobileDetect->isMobile() && !$mobileDetect->isTablet()) {

$this->redirectMobile($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。