

پروژه دمو rest-api فروشگاهی - محمد بشیری نیا - ۰۹۲۱۰۴۴۷۶۱۳ :

آدرس گیت هاب پروژه:

https://github.com/B-Mohammad/online_shop_with_slim_framework

آدرس مستند پستمن از endpoint ها و نحو استفاده از api ها:

<https://documenter.getpostman.com/view/26425641/2sAYkDNgRY>

فایل مربوط به ساختار دیتابیس در: online_shop_db.pdf

راه اندازی:

نیازمندی‌ها:

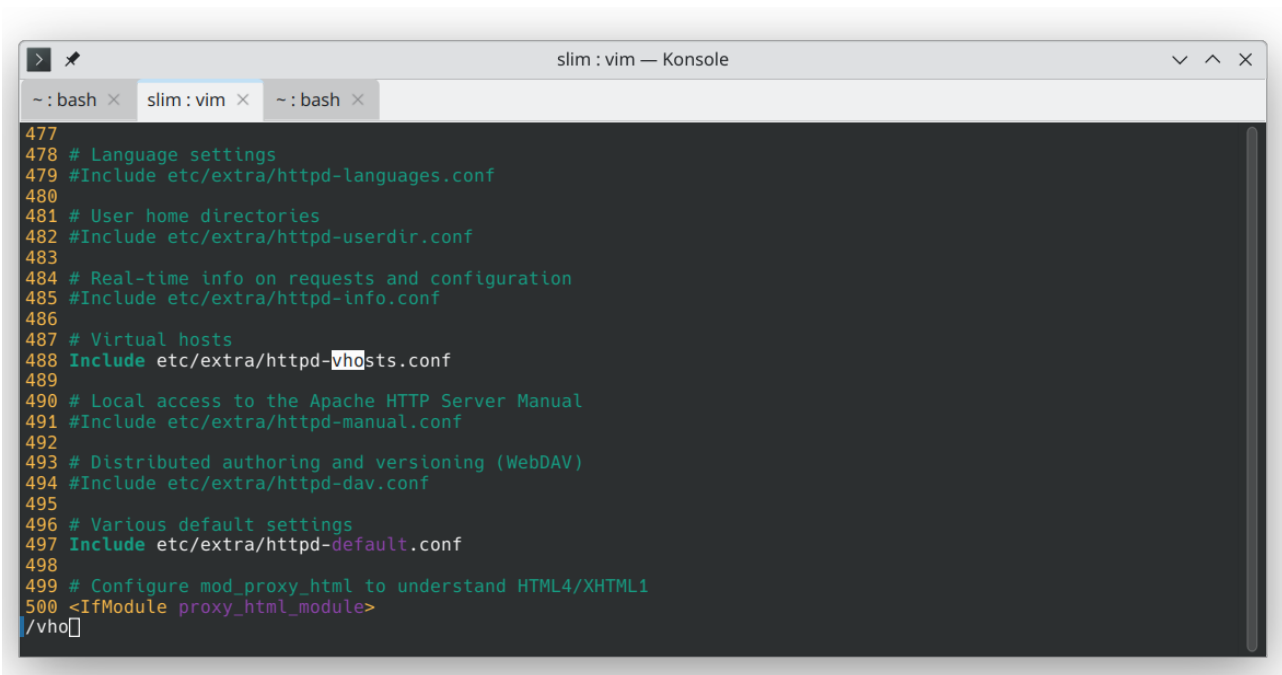
php ۸.۰.۲، دیتابیس mysql و پکیج های مربوط slim, jwt, psr, validator

مراحل راه اندازی:

۱. ابتدا یک دیتا بیس به نام online_shop از طریق پنل phpmyadmin بسازید، بکاپ دیتا بیس موجود در فایل‌ها به نام online_shop.sql را داخل دیتابیس لود کنید.
نکته اگر برای دیتابیس نام دیگری انتخاب کردید می‌توانید از مسیر محل پروژه، config/ddefinition.php نام دیتابیس را تغییر دهید.

۲. فایل زیپ پروژه را داخل مسیر xampp/htdocs در پوشه ای با نام دلخواه از حالت فشرده خارج کنید.

۳. داخل تنظیمات سرور آپاچی ویرژوال هاست را به شرح زیر فعال می‌کنیم.
داخل مسیر نصب xampp ویرژوال هاست را فعال می‌کنیم.
مسیر: etc/httpd.conf

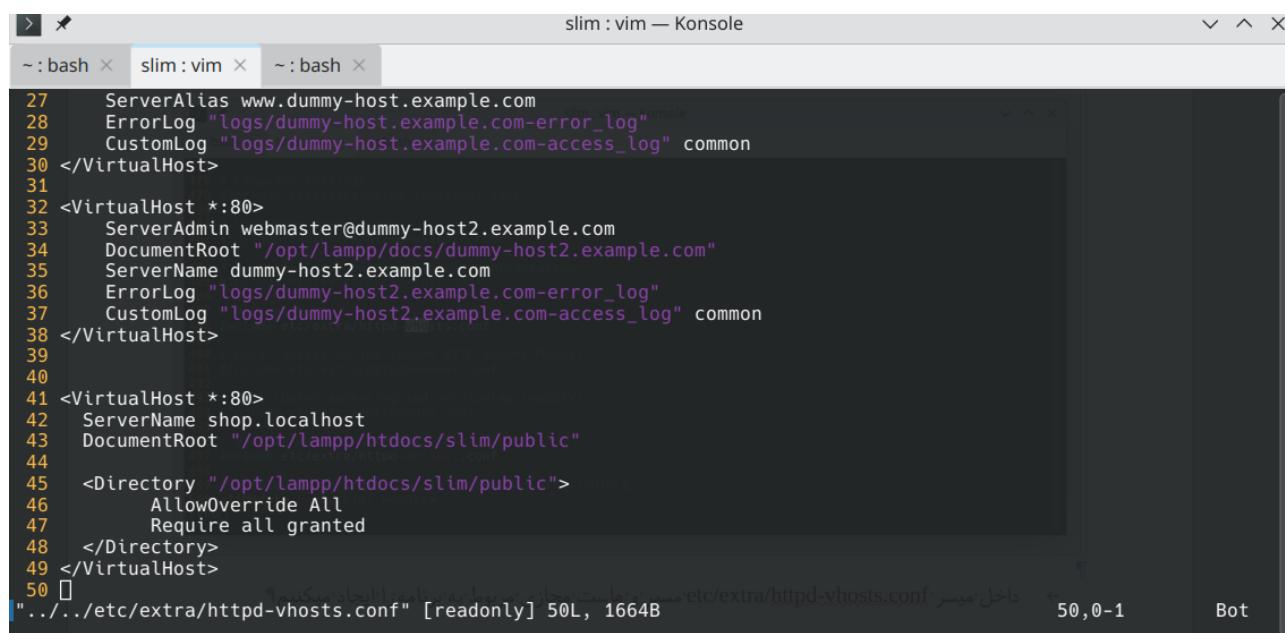


```
slim : vim — Konsole
~: bash x slim: vim x ~: bash x
477
478 # Language settings
479 #Include etc/extra/httpd-languages.conf
480
481 # User home directories
482 #Include etc/extra/httpd-userdir.conf
483
484 # Real-time info on requests and configuration
485 #Include etc/extra/httpd-info.conf
486
487 # Virtual hosts
488 Include etc/extra/httpd-vhosts.conf
489
490 # Local access to the Apache HTTP Server Manual
491 #Include etc/extra/httpd-manual.conf
492
493 # Distributed authoring and versioning (WebDAV)
494 #Include etc/extra/httpd-dav.conf
495
496 # Various default settings
497 Include etc/extra/httpd-default.conf
498
499 # Configure mod_proxy_html to understand HTML4/XHTML1
500 <IfModule proxy_html_module>
/vho
```

داخل میسر `etc/extra/httpd-vhosts.conf` مسیر و هاست مجازی مربوط به برنامه را ایجاد میکنیم

```
<VirtualHost *:80>
ServerName shop.localhost
DocumentRoot "/opt/lampp/htdocs/slim/public"

<Directory "/opt/lampp/htdocs/slim/public">
    AllowOverride All
    Require all granted
</Directory>
</VirtualHost>
```



```
slim : vim — Konsole
~: bash x slim: vim x ~: bash x
27 ServerAlias www.dummy-host.example.com
28 ErrorLog "logs/dummy-host.example.com-error_log"
29 CustomLog "logs/dummy-host.example.com-access_log" common
30 </VirtualHost>
31
32 <VirtualHost *:80>
33 ServerAdmin webmaster@dummy-host2.example.com
34 DocumentRoot "/opt/lampp/docs/dummy-host2.example.com"
35 ServerName dummy-host2.example.com
36 ErrorLog "logs/dummy-host2.example.com-error_log"
37 CustomLog "logs/dummy-host2.example.com-access_log" common
38 </VirtualHost>
39
40
41 <VirtualHost *:80>
42 ServerName shop.localhost
43 DocumentRoot "/opt/lampp/htdocs/slim/public"
44
45 <Directory "/opt/lampp/htdocs/slim/public">
46     AllowOverride All
47     Require all granted
48 </Directory>
49 </VirtualHost>
50
"../../etc/extra/httpd-vhosts.conf" [readonly] 50L, 1664B 50,0-1 Bot
```

۴. یک بار سرویس ایاچی را ریست میکنیم. تا تغییرات اعمال شود.

۵. در صورت نبود مشکل پروژه از طریق آدرس `shop.localhost` در دسترس است به عنوان مثال با درخواست `get` به `shop.localhost/api/products` میتوان لیست محصولات را گرفت.

نکته: تمام فایل های مربوط به پکیج ها همراه پروژه است و نیاز به نصب مجدد ندارد.

نکته: در صورت نیاز به رمز عبور کاربران برابر با ۱۲۳۴۵ است به دلیل هش شدن داخل دیتا بیس مشخص نیست.

توضیحات مربوط به ساختار پروژه:

در این پروژه از فریمورک `jwt`، `slim`، کانتینر برای دیپندسی اینجکشن، `Valitron` برای ولیدیشن و ... استفاده شده است. که فایل `composer.json` قابل مشاهده است.

```
{
  "require": {
    "slim/slim": "^4.14",
    "slim/psr7": "^1.7",
    "php-di/php-di": "^7.0",
    "vlucas/valitron": "^1.4",
    "firebase/php-jwt": "^6.11"
  },
  "autoload": {
    "psr-4": {
      "": "src/"
    }
  }
}
```

پوشه بندی:

dir public:

شامل index.php که فریم ورک اسلیم و برنامه را راه اندازی میکند.

Dir config:

اطلاعات استاتیک برنامه و مسیر های اند پوینت ها در این پوشه به ترتیب در فایل های definition.php و routes.php قرار دارند.

Dir src/App

Controller

این پروژه شامل چهار دسته endpoint است احراز هویت، کالاها، سبد خرید، سفارشات. که برای هر دسته فایلی در نظر گرفته شد و متد مربوطه در آن قرار گرفت

Middleware

شامل میدلویر های کد گذار رسیپانس، مربوط به مدیریت خطا ها و دستورات get و احراز هویت است که میتواند به هر مسیر api متصل شود.

Repo

وظیفه برقراری ارتباط با دیتابیس و فراهم کردن داده برای لایه کنترلر را دارد.

Database.php

کانکشن دیتابیس توسط این کلاس برقرار میشود که PDO استفاده شد

نکته: در این پروژه از دیپندسی اینجکشن استفاده شد تا تمام کلاس های وابسته یکبار ایجاد میشوند و در کانتینر نگه داری می شوند که از تولید دوباره آن ها جلوگیری شود.