技术文档汇总

目录

[一、Laravel 1](#_Toc485737852)

[1.Vagrant+Virtual+HomeStead环境搭建 1](#_Toc485737853)

[2.Laravel文件存储问题 8](#_Toc485737854)

[3.laravel-mix及模板调用问题 9](#_Toc485737855)

[4. Vue组件化-关注/取关按钮 12](#_Toc485737856)

[5.axios 13](#_Toc485737857)

[6.Laravel5.4 Auth::user()无法获取认证用户信息问题 14](#_Toc485737858)

[7.Vue2生命周期函数中this指向不是组件实例的问题 14](#_Toc485737859)

[二、MySQL 15](#_Toc485737860)

[1.MySql数据库更新5.7版本后datetime不支持的解决办法 15](#_Toc485737861)

[三、前端 16](#_Toc485737862)

[1.颜色搭配 16](#_Toc485737863)

一、Laravel

1.Vagrant+Virtual+HomeStead环境搭建

本文在Windows操作系统下搭建环境，其他操作系统请仅作参考

一、下载工具

1.virtualBox

Oracle 公司的虚拟机软件, 能运行在当前大部分流行的系统上

网址：<https://www.virtualbox.org/wiki/Downloads>

2.vagrant

命令行接口，提供一种命令行接口, 允许自动化安装虚拟机, 并且因为是脚本编写成的 provision 文本文件, 给共享虚拟机配置提供了可能, Homestead 正是构建在 Vagrant 之上

网址：<https://www.vagrantup.com/downloads.html>

二、安装

1.将vagrant和VirtualBox安装完成后打开Gitbash命令窗口进入本地路径**（没有Git请先下载安装Git）**

cd ~

2.克隆 homestead 项目

git clone https://github.com/laravel/homestead.git Homestead

3.在virtualbox虚拟机中加入 homestead box

有两种方法：

（1）执行以下命令添加 box，这种方法下载很慢，容易被墙，我即使开着VPN也下不了，所以我果断用第二种方法

vagrant box add laravel/homestead

（2）命令行下下载速度太慢的话可以利用工具下载以下链接加速，即先将资源下载到本地，再添加到虚拟机中**（我用迅雷下载的）**

资源地址：

<https://atlas.hashicorp.com/laravel/boxes/homestead/versions/2.1.0/providers/virtualbox.box>

下载完毕我们得到了一个virtualbox.box文件，将它放在C：根目录下**（环境搭建完后可以删除这个文件）**

创建一个metadata.json文件放置在本地目录下（C:用户/当前用户/）**（很重要）**

metadata.json内容：

{

"name": "laravel/homestead",

"versions": [{

"version": "2.1.0",

"providers": [{

"name": "virtualbox",

"url": "file://C:/virtualbox.box"

}]

}]

}

**（我的盒子是2.1.0版所以version是2.1.0，virtualbox，vagrant最好都用最新版避免出现兼容和再更新问题）**

Gitbash：

cd ~

vagrant box add metadata.json

三、配置

1.生成配置文件

cd homestead

bash init.sh

打开Homestead目录，可以找到Homestead.yaml文件

这时开始修改Homestead.yaml文件**（我用的是Sublimt text）**

folders表示主机和虚拟机的共享文件

sites表示域名访问虚拟机位置

我的修改如下：

**（注意：请在源文件上修改，不要复制代码，yaml文件里多一个空格就可能安装失败！）**

ip: "192.168.10.10"

memory: 2048

cpus: 2

provider: virtualbox

authorize: ~/.ssh/id\_rsa.pub

keys:

- ~/.ssh/id\_rsa

folders:

- map: ~/Code/TaskManager

to: /home/vagrant/Code/TaskManager

sites:

- map: taskmanager.app

to: /home/vagrant/Code/TaskManager/public

databases:

- homestead

为了对应共享的文件，在本地目录下新建目录Code，在Code下新建子目录TaskManager

2.域名重定向

修改主机hosts文件添加域重定向

位置：C:\Windows\System32\drivers\etc\hosts

添加一条重定向代码：

192.168.10.10 taskmanager.app

**（如果保存不了，右键hosts-属性-安全-编辑，给当前用户分配此文件的权限）**

3.SSH密钥

在gitbash窗口输入：

ssh-keygen

根据提示输入你要设置的密码**（我设置为空，全部enter跳过）**

然后打开homestead/scripts/homestead.rb，查看版本限制：

config.vm.box\_version = settings["version"] ||= ">= 2.0.0"

如果不符则修改

**（注意：不要看网上的执行vagrant init laravel/homestead 命令，否则Vagrantfile会被初始化，vagrant up的时候不会执行yaml配置！）**

4.安装环境

cd homestead

vagrant up

等待一段时间安装完成

**（命令执行后，显示信息第二行如果是import...laravel/homeatead才是正确的）**

vagrant ssh

进入虚拟机

ls

看是否有个Code文件，如果有，则yaml配置文件执行成功了

四、Box内部署

1.composer

修改虚拟机内composer下载地址转到中国镜像

composer config -g repo.packagist composer https://packagist.phpcomposer.com

下载laravel项目

cd Code

sudo composer create-project laravel/laravel TaskManager

在主机浏览器输入域名：taskmanager.app

访问到Laravel首页则配置成功！

2.修改laravel配置

修改.env文件内数据库名

...

3.安装phpmyadmin

网址：[www.phpmyadmin.net](http://www.phpmyadmin.net)

下载安装解压改名为phpMyAdmin放在主机的Code目录下**（注意：phpMyAdmin目录下就是doc、examples等子目录）**

修改yml文件：

在共享文件后添加

- map: ~/Code/phpMyAdmin

to: /home/vagrant/Code/phpMyAdmin

在域名重定向后添加：

- map: phpmyadmin.app

to: /home/vagrant/Code/phpMyAdmin

在hosts文件添加：

192.168.10.10 phpmyadmin.app

4.重载yaml

登录虚拟机

vagrant up

重新加载配置

vagrant provision

vagrant ssh

主机浏览器url输入phpmyadmin.app，出现phpmyadmin页面，部署完成！

2.Laravel文件存储问题

Laravel有多种文件存储方式，包括：local、public、第三方云存储等，存储方式定义在config/filesystem.php中

本地存储选择local或者public方式

旧版本中：

本地存储磁盘直接放在/public下，用户可以直接访问到

新版本中：

本地磁盘在/storage/app/public下，用户上传的文件不会被直接访问到，在项目根目录下执行：

php artisan storage:link

建立一个/storage/app/public到/public/storage的软链接

在filesystem中设置url即可

**（但是在virtualbox中出于安全考虑，共享文件夹中禁止建立软链接，这样最好将文件系统部署在云端）**

3.laravel-mix及模板调用问题

Laravel中composer用于管理php依赖，而npm用于管理前端依赖

1.升级npm

确保下载依赖不出错

npm install npm -g

如果有权限问题则

sudo npm install npm -g

检查node和npm版本是否最新：

node -v

npm -v

发生权限错误则给出权限，输入：

sudo chown -R $USER:$(id -gn $USER) /home/vagrant/.config

2.安装package.json中的前端依赖

cd到项目根目录下

npm install

如果在windows虚拟机中，由于禁止共享文件建立软链接，所以需要增加选项

npm install --no-bin-links

**（如果安装出错，可以删除项目根目录下的node\_moudles目录，重新安装）**

3.使用Laravel-mix编译整合前端资源

Laravel5.4中mix抛弃了gulp改用webpack，后者对API更加友好

npm run dev

如果报错cross-env：not found，建议用旧版的指令替换，具体操作为，将package.json中script中的内容修改为：

**"prod"**: **"npm run production"**,  
**"dev"**: **"node node\_modules/cross-env/dist/bin/cross-env.js NODE\_ENV=development node\_modules/webpack/bin/webpack.js --progress --hide-modules --config=node\_modules/laravel-mix/setup/webpack.config.js"**,  
**"watch"**: **"node node\_modules/cross-env/dist/bin/cross-env.js NODE\_ENV=development node\_modules/webpack/bin/webpack.js --watch --progress --hide-modules --config=node\_modules/laravel-mix/setup/webpack.config.js"**,  
**"watch-poll"**: **"node node\_modules/cross-env/dist/bin/cross-env.js NODE\_ENV=development node\_modules/webpack/bin/webpack.js --watch --watch-poll --progress --hide-modules --config=node\_modules/laravel-mix/setup/webpack.config.js"**,  
**"hot"**: **"node node\_modules/cross-env/dist/bin/cross-env.js NODE\_ENV=development node\_modules/webpack-dev-server/bin/webpack-dev-server.js --inline --hot --config=node\_modules/laravel-mix/setup/webpack.config.js"**,  
**"production"**: **"node node\_modules/cross-env/dist/bin/cross-env.js NODE\_ENV=production node\_modules/webpack/bin/webpack.js --progress --hide-modules --config=node\_modules/laravel-mix/setup/webpack.config.js"**

再执行npm run dev

4.下载并导入前端资源

cd /resources/assets/

mkdir css

cd css

wget 前端css资源网址

cd ../../js

wget 前端js资源网址

资源文件存放在assets下的css和js文件夹中，我们需要将其导入到项目中

导入js文件：

打开assets/js/bootstrap.js，在require(jquery)下添加资源名，如：

**window**.**$** = **window**.**jQuery** = **require**(**'jquery'**);  
**require**(**'bootstrap-sass'**);  
**require**(**'./select2.min'**);

导入css文件：

打开assets/sass/app.scss，import资源，如：

**@import "./../css/select2.min"**;

5.mix使用

如果要让前端资源随着开发改动自动编译且避免样式缓存问题，则在webpack.mix.js中添加.version()，如下：

**mix**.**js**(**'resources/assets/js/app.js'**, **'public/js'**)  
 .**sass**(**'resources/assets/sass/app.scss'**, **'public/css'**)  
 .version();

执行监听

npm run watch

在public/css和js下会生成带哈希码的css和js文件，这个就是我们要导入视图的资源

一般把编译整合后的资源放在app.blade.php中，css放开头，js放结尾，如:

<**link href="**{{ mix(**'css/app.css'**) }}**" rel="stylesheet"**>

<**script src="**{{ mix(**'js/app.js'**) }}**"**></**script**>

**（注意：子视图的css和js样式一定要放在section中，通过yield引用，而且要放在公用样式之后，否则会被覆盖！）**

4. Vue组件化-关注/取关按钮

<question-follow-button question="{{$question->id}}">

</question-follow-button>

<template>  
 <button class="btn btn-default"  
 v-bind:class="{'btn-success':followed}"  
 v-text="text"  
 v-on:click="follow"  
 >  
 </button>  
</template>  
  
<script>  
 export default {  
 props:['question'],  
 mounted() {  
 this.$http.post('/api/question/isFollow',{'question':this.question}).then(response => {  
 this.followed = response.data.followed;  
 console.log("question followed is "+response.data.followed);  
 })  
 },  
 data(){  
 return {  
 followed: false,  
 }  
 },  
 methods:{  
 follow(){  
 this.$http.post('/api/question/follow',{'question':this.question}).then(response => {  
 this.followed = response.data.followed;  
 console.log(response.data);  
 })  
 }  
 },  
 computed: {  
 text(){  
 return this.followed ? '已关注' : '关注问题';  
 }  
 }  
 }  
</script>

5.axios

vuejs2.0已停止对vue-resource的更新，开始使用axios实现vue组件与后端的异步数据交互

6.Laravel5.4 Auth::user()无法获取认证用户信息问题

原因：未开启Session

在Kernel.php中

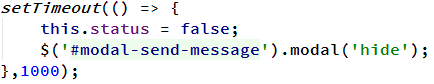
protected $middleware = [  
 \Illuminate\Foundation\Http\Middleware\CheckForMaintenanceMode::class,  
 \Illuminate\Foundation\Http\Middleware\ValidatePostSize::class,  
 \App\Http\Middleware\TrimStrings::class,  
 \Illuminate\Foundation\Http\Middleware\ConvertEmptyStringsToNull::class,  
 \Illuminate\Session\Middleware\StartSession::class,  
 \Illuminate\Session\Middleware\AuthenticateSession::class,  
 \Illuminate\View\Middleware\ShareErrorsFromSession::class,  
];

'web' => [  
 \App\Http\Middleware\EncryptCookies::class,  
 \Illuminate\Cookie\Middleware\AddQueuedCookiesToResponse::class,  
 \Illuminate\Session\Middleware\StartSession::class,  
 \Illuminate\Session\Middleware\AuthenticateSession::class,  
 \Illuminate\View\Middleware\ShareErrorsFromSession::class,  
 \App\Http\Middleware\VerifyCsrfToken::class,  
 \Illuminate\Routing\Middleware\SubstituteBindings::class,  
],

7.Vue2生命周期函数中this指向不是组件实例的问题

在生命周期函数的function(){}中this指向上下文

要使this指向组件实例，需用（）=>{}，这样做相当于function(){}.bind(this)将函数的this改变指向



8.Vue组件报错解决方案

Avoid mutating a prop directly since the value will be overwritten whenever the parent component re-renders. Instead, use a data or computed property based on the prop's value. Prop being mutated:

Vue2中，父组件传入的props属性值不允许直接改变，否则会报以上错误

解决方案：

将props要改变的属性赋给data中建立一个变量值，将该值与组件变量绑定

<**template**>  
 <**button class="btn btn-default"  
 v-bind:class="**{**'btn-primary'**:voted}**"  
 v-on:click="**vote**"** ><**span class="glyphicon glyphicon-thumbs-up" aria-hidden="true"**></**span**>**&ensp;**{{ Mycount }}  
 </**button**>  
</**template**>  
  
<**script**>  
 **export default** {  
 **props**:[**'answer'**,**'count'**],  
 mounted() {  
 axios.post(**'/api/answer/'**+**this**.answer+**'/votes/users'**).then(response => {  
 **this**.**voted** = response.data.voted;  
 })  
 },  
 data(){  
 **return** {  
 **voted**: **false**,  
 **Mycount**:**this**.count,  
 }  
 },  
 **methods**:{  
 vote(){  
 axios.post(**'/api/answer/vote'**,{**'answer'**:**this**.**answer**}).then(response => {  
 **this**.**voted** = response.data.voted;  
 response.data.voted ? **this**.**Mycount**++ : **this**.**Mycount**--;  
 })  
 }  
 }  
 }  
</**script**>

8.关于Laravel5表单验证错误后Session中的$errors和$old不显示的问题

方法1：

将该表单发送的路由中间件由

Route::group(['middleware' => 'web'], function () {}

改为

Route::group(['middlewareGroups' => 'web'], function () {}

方法2：

将Kernel.php中

protected $middlewareGroups ’web’中的\Illuminate\Session\Middleware\StartSession::class,

\Illuminate\View\Middleware\ShareErrorsFromSession::class,

移动到$middleware中

二、MySQL

1.MySql数据库更新5.7版本后datetime不支持的解决办法

MySQL5.7版本兼容问题

具体请参考：

<https://dev.mysql.com/doc/refman/5.7/en/sql-mode.html#sql-mode-changes>

解决办法：

修改datetime数据类型的默认值，比如直接无定义，勾选可以为空

然后在自己的项目里修改相应的含有datetime字段的程序

**（注意：如果原来多个datatime类型字段默认值都是：0000-00-00 00:00:00，那么修改字段的时候需要多个字段同时修改，否则仍会报错！）**

三、前端

1.颜色搭配

蓝灰色：#304356

深蓝色：#3a5775

淡金色：#9c9987

1140x350

219x162

240x178