به نام خدا

زیرساخت برای ماژول های addon در whmcs

این زیرساخت برای سهولت در توسعه ماژول های addon در whmcs آماده شده است که به صورت component های مجزا توسعه داده شده است.

برای دریافت این package شما فقط می بایست در فایل composer.json ماژول addon خود آدرس زیر قرار دهید.

"repositories": [  
 {  
 "type": "git",  
 "url": "http://git.greenweb.local/p.eraghi/addon.git"  
 }  
],

و سپس دستور composer install را اجرا کنید.

بعد از نصب ، این package در پوشه vendor ماژول شما قرار می گیرد.

نحوه استفاده از زیرساخت :

در ابتدا شما در فایل اصلی ماژول خود باید فایل autoload.php را require کنید.

require\_once *\_\_DIR\_\_*.*DIRECTORY\_SEPARATOR*."vendor".*DIRECTORY\_SEPARATOR*."autoload.php";

سپس شما باید فایل config.php ماژول خود را که به صورت پیش فرض در پوشه config قرار دارد را خوانده و به عنوان پارمتر ورودی اولیه به کلاس Addon زیرساخت پاس داده شده و کلاس را new می کنید.

$config = config('config');  
new Addon($config);

حالا شما به امکانات زیرساخت دسترسی دارید.(به صورت زیر)

Addon::*$instance*

کاکرد فایل config.php :

return [  
 'loader' => [  
 // load default component  
 'routing' => \greenweb\addon\routing\Routing::class,  
 'request' => \greenweb\addon\request\Request::class,  
 'permission' => \greenweb\addon\permission\permission::class,  
 'admin' => \greenweb\addon\Admin\Admin::class,  
 'user' => \greenweb\addon\User\User::class,  
 'session' => \greenweb\addon\session\Session::class,  
 'dateTime' => \greenweb\addon\formatter\DateTime::class,  
 'setting' => \greenweb\addon\setting\Setting::class  
 ],  
  
 'permission' => [  
 // permission list  
 'showUser',  
 'addUser',  
 ],  
  
 'AdminViewTemplatePath' => 'Resources/view/admin/',  
 'ClientViewTemplatePath' => 'Resources/view/client/',  
 'AdminControllerPath' => 'Http/Controller/',  
 'ClientControllerPath' => 'Http/Controller/client/',  
 'RoutePath' => 'Routes/',  
 'ModelPath' => 'Models/',  
  
 'settingConfig' => [  
 'table' => 'setting',  
 'code' => 'cancel\_request\_hook'  
 ]  
];

قسمت loader :

در این قسمت component های استفاده شده در زیرساخت قرار دارند که شما می توانید در صورتی که نمی خواهید از هر یک از آن ها استفاده کنید آن را comment یا پاک کنید.

در ادامه هر یک از component ها را به صورت مجزا توضیح خواهیم داد.

قسمت permission :

در این قسمت شما میتوانید لیست persmission هایی که می خواهید در ماژول خود برای دسترسی کاربرهای ادمین داشته باشید را قرار می دهید.

در ادامه نحوه استفاده از permission را توضیح خواهیم داد.

قسمت AdminViewTemplatePath :

در این قسمت شما مسیر پیش فرض فایل های tpl سمت ادمین را قرار می دهید.

به صورت پیش فرض این مسیر به صورت زیر می باشد.

Resources/view/admin/

قسمت ClientViewTemplatePath :

در این قسمت شما مسیر پیش فرض فایل های tpl سمت کلاینت را قرار می دهید.

به صورت پیش فرض مسیر به صورت زیر می باشد.

Resources/view/client/

قسمت AdminControllerPath :

در این قسمت مسیر پیش فرض controller های سمت ادمین را قرار می دهید.

به صورت پیش فرض مسیر به صورت زیر می باشد.

Http/Controller/

قسمت ClientControllerPath :

در این قسمت مسیر پیش فرض controller های سمت کلاینت را قرار می دهید.

به صورت پیش فرض مسیر به صورت زیر می باشد.

Http/Controller/client/

قسمت RoutePath :

در این قسمت مسیر پیش فرض فایل route.php را قرار خواهید داد.

به صورت پیش فرض مسیر به صورت زیر می باشد.

Routes/

قسمت ModelPath :

در این قسمت مسیر پیش فرض مدل ها را قرار خواهید داد.

به صورت پیش فرض مسیر به صورت زیر می باشد.

Models/

قسمت SettingCoonfig :

Table : نام جدول setting شما قرار می گیرد. (به صورت پیش فرض جدول setting می باشد)

Code : مقدار code در جدول setting مربوط به ماژول فعلی . (این مقدار را حتما وارد نمایید)

خب تا اینجای کار ما فایل config رو توضیح دادیم در قسمت بعد در صورتی بخواهید از routing زیر ساخت استفاده کنید باید در فایل routes.php روت های خودتون رو تعریف کنید.

Routes.php :

return [  
 'admin' => [  
 'user/list' => [  
 'controller' => 'UserController@index',  
 'middleware' => [  
  
 ],  
 'name' => 'لیست کاربران',  
 ],  
 'admin/index' => [  
 'controller' => 'AdminController@index',  
 'middleware' => [  
  
 ],  
 'name' => 'لیست کاربران',  
 ]  
 ],  
 'client' => [  
 'order/list' => [  
 'controller' => 'OrderController@index',  
 'middleware' => [  
  
 ],  
 'name' => 'لیست سفارشات'  
 ]  
 ]  
  
];

خب در این فایل همونطور که مشاهده می کنید ما یک آرایه بر می گردونیم که از دو قسمت تشکیل شده :

1. Admin : در این قسمت روت های مربوط به ادمین قرار می گیرد.
2. Client : در این قسمت روت های مربوط به کلاینت قرار می گیرد.

خب برای تعریف روت در هر قسمتی که می خواهیم اضافه کنیم ما در ابتدا باید یک key قرار دهیم که در آن مقدار action در url شما خواهد بود سپس value برای key شما باید یه آرایه قرار دهید که در قسمت اول شما controller خود را وارد نمایید که مقدار کنترلر یک رشته دو قسمتی خواهد بود که با @ از یکدیگر جدا شده اند که در مثال بالا مشاهده می کنید قسمت اول رشته نام کنترلر و قسمت دوم نام متد در کنترولر خواهد بود.

برای تعریف کنترلر ها شما باید در مسیر پیش فرضی که در config.php قرار دادید یک کلاس درست کنید و آن را بسته به نوع کنترلر که برای admin هست یا برای client به ترتیب از AdminController یا ClientController ارث بری کنید تا مقادیر اولیه کنترل های شما ست شود.

به عنوان مثاب در زیر کنترلر را مشاهده کنید.

<?php  
  
  
use greenweb\addon\controller\AdminController;  
  
class UserController extends AdminController  
{  
 public function index()  
 {  
 return $this->view('child/index',[  
 'vars' => $this->vars  
 ]);  
 }  
}

خب همانطور که مشاهده می کنید شما برای ارسال اطلاعاتتون به قسمت view های خودتون باید از متد view استفاده کنید.

نکته مهم : شما در کنترلر برای استفاده از component ها و دریافت vars به ترتیب به صورت زیر میتوانید دسترسی داشته باشید.

$this->app; // برای دسترسی به component ها  
$this->vars; // مقادیر پیش فرض vars

به عنوان مثال : (admin)

return $this->view('child/index',[  
 'vars' => $this->vars  
 ]);

در قسمت view در سمت ادمین که در بالا مشاهده میکنید شما در پارامتر اول باید نام فایل را قرار دهید (در صورتی که در پوشه خاصی در این قسمت هست باید با "/" یا "." نام پوشه و فایل را جدا کنید) و پارامتر دوم شما میتوانید تمامی مقادیری که میخواهید در فایل tpl خود استفاده کنید را به صورت key,value در آرایه ارسال کنید.

پارامتر اول : ارسال نام فایل tpl

پارامتر دوم : ارسال اطلاعات قابل استفاده در فایل tpl

به عنوان مثال : (client)

return $this->view('index','title','breadcrump', 'data', 'requireLogin');

در قسمت view در سمت client کمی اطلاعات بیشتر را باید پاس بدهیم .

پارامتر اول : نام فایل tpl

پارامتر دوم : عنوان صفحه

پارمتر سوم : breadcrump

پارامتر چهارم : داده های مورد استفاده در فایل tpl

پارامتر پنجم : نیاز به لاگین دارد یا خیر ؟(به صورت پیش فرض false می باشد)

نکته : مسیر پیش فرض فایل های tpl در config.php تعریف شده است.

خب تقریبا کنترلر ها و ویو ها رو توضیح دادم در قسمت بعد بریم سراغ کامپوننت permission که خیلی مهم هستش.

کامپوننت permission :

در این کامپوننت ما با استفاده از permission هایی که در config.php تعریف شده اند ما به هر کاربر سمت ادمین permission های مربوط به خودش رو توسط full admin نسب می دهیم.

روش مقدار دهی هر permission به یک کاربر :

در زیر ساخت دو روت تعریف شده است که به صورت پیش فرض آدرس آن در AdminController هستش

و آدرس آن admin/permission و admin/addPermission قرار داده شده است که آدرس اول برای نمایش صفحه permission ها هستش و آدرس دوم قسمت ذخیره کردن permission ها می باشد.

برای استفاده از permission ها شما در کنترل های خود میتوانید با استفاده از دستور زیر دسترسی کاربر را چک کنید.

$this->app->permission->has('permission\_name','user\_id');

در استفاده از این متد شما دو پارامتر باید برای آن ارسال کنید که :

پارامتر اول : نام یکی از permission هایی که در config.php تعریف کرده اید.

پارامتر دوم : شناسه کاربر که میخواهید برای آن دسترسی رو چک کنید (به صورت پیش فرض کاربر فعلی می باشد)

نکته : برای استفاده از permission ها روش دیگری نیز وجود دارد که در قسمت استفاده از کامپوننت Admin توضیح خواهیم داد.

کامپوننت Request :

این کامپوننت بسیار ساده ای هستش که برای دریافت اطلاعات که توسط http ارسال می شود استفاده شده است.

نحوه استفاده از این کامپوننت به صورت زیر خواهد بود.

$action = Addon::*$instance*->request::*get*('action', 'user/list');  
$action = Addon::*$instance*->request::*post*('action', 'user/list');

به ترتیب برای دریافت مقادیری که به صورت get , post ارسال می شوند استفاده می شوند:

پارامتر اول : نام key که می خواهیم مقدارش را دریافت کنیم.

پارامتر دوم : مقدار پیش فرض که در صورت وجود نداشتن مقدار در key آن را بر می گردانیم.

کامپوننت Session :

این کامپوننت برای دسترسی راحت تر به مقادیر session ها توابعی را در اختیار توسعه دهندگان قرار می دهد.

$this->app->session->get('key', 'default\_value');

$this->app->session->set('key', 'value');

$this->app->session->remove('key');

$this->app->session->flush();

متد get : در این متد شما می توانید مقدار key مورد نظر خود را از session دریافت کنید و در صورت نبود میتوانید مقدار پیش فرض برای آن تعیین کنید.

متد set : در این متد شما مقدار مورد نظر خودتان در key مورد نظز خودتان قرار دهید.

متدremove : مقدار key مورد نظر را از session حذف می کند.

متد flush : این متد تمام مقادیر session را حذف میکند.

کامپوننت dateTime :

همانطور که از اسم این کامپوننت مشخص هستش برای استفاده تبدیل های تاریخ و همچنین بازه های زمانی استفاده می شود که متد های زیر را مشاهده می کنید:

$this->app->dateTime->date('1399/02/25')->toJalali()->format('Y-m-d H:i:s');

$this->app->dateTime->date('2015/07/11')->toGeo()->format('Y-m-d H:i:s');

$this->app->dateTime->diffDate('1398/03/22','1398/04/17','day');

$this->app->dateTime->scrollingDate('1398/03/22',-3,'day')->toString();

$this->app->dateTime->modify('1398/03/22','+15 day')->toJalali()->toString();

متد date :تاریخ وارد شده شما را در خود کلاس dateTime نگه می دارد و آبجکت کلاس را بر می گرداند.

متد toJalali : تاریخ نگه داشته شده در کلاس را به جلالی تبدیل می کند و آبجکت کلاس را بر می گرداند.

متد toGeo : تاریخ نگه داشته شده در کلاس را به میلادی تبدیل می کند و آبجکت کلاس را بر می گرداند.

متد diffDate : فاصله بین دو تاریخ (پارمتر اول و پارامتر دوم)را بر حسب مقدار وارد شده در پارامتر سوم بر می گرداند.

مقادیر قابل قبول برای پارمتر سوم این متد :

year  
month  
day  
hours  
minute

متد scrollingDate : در این متد شما می توانید تاریخ وارد شده در پارمتر اول را به اندازه مقدار وارد شده در پارمتر دوم را بر حسب پارمتر سوم وارد شده محاسبه کنید.

مقادیر قابل قبول برای پارامتر سوم :

year  
month  
day  
hours  
minute

متد modify : در این متد شما تاریخ وارد شده در پارامتر اول را با استفاده از مقدار پارمتر دوم تغییر دهید.

نکته : پارمتر دوم تمامی مقادیر قابل استفاده strtotime در php

کامپوننت setting : در این کامپوننت شما می توانید اطلاعات تنظیمات خودتون را در جدول setting که در config.php تعریف کرده اید را به صورت key , value دریافت کنید .

$this->app->setting->config['key'];

کامپوننت Admin : با استفاده از این کامپوننت شما به یک شی از admin لاگین شده فعلی در اختیار دارید.

$this->app->admin->current();  
$this->app->admin->can('permission');

کامپوننت User : با استفاده از این کامپوننت شما به یک شی از user لاگین شده فعلی در اختیار دارید.

$this->app->user->current();  
$this->app->user->can('permission');

در متد های current : شما یه شی از admin دریافت خواهید کرد.

در متدهای can : شما میتوانید با توجه به permission که به آن پاس می دهید متوجه بشید دسترسی وجود دارد یا خیر ؟