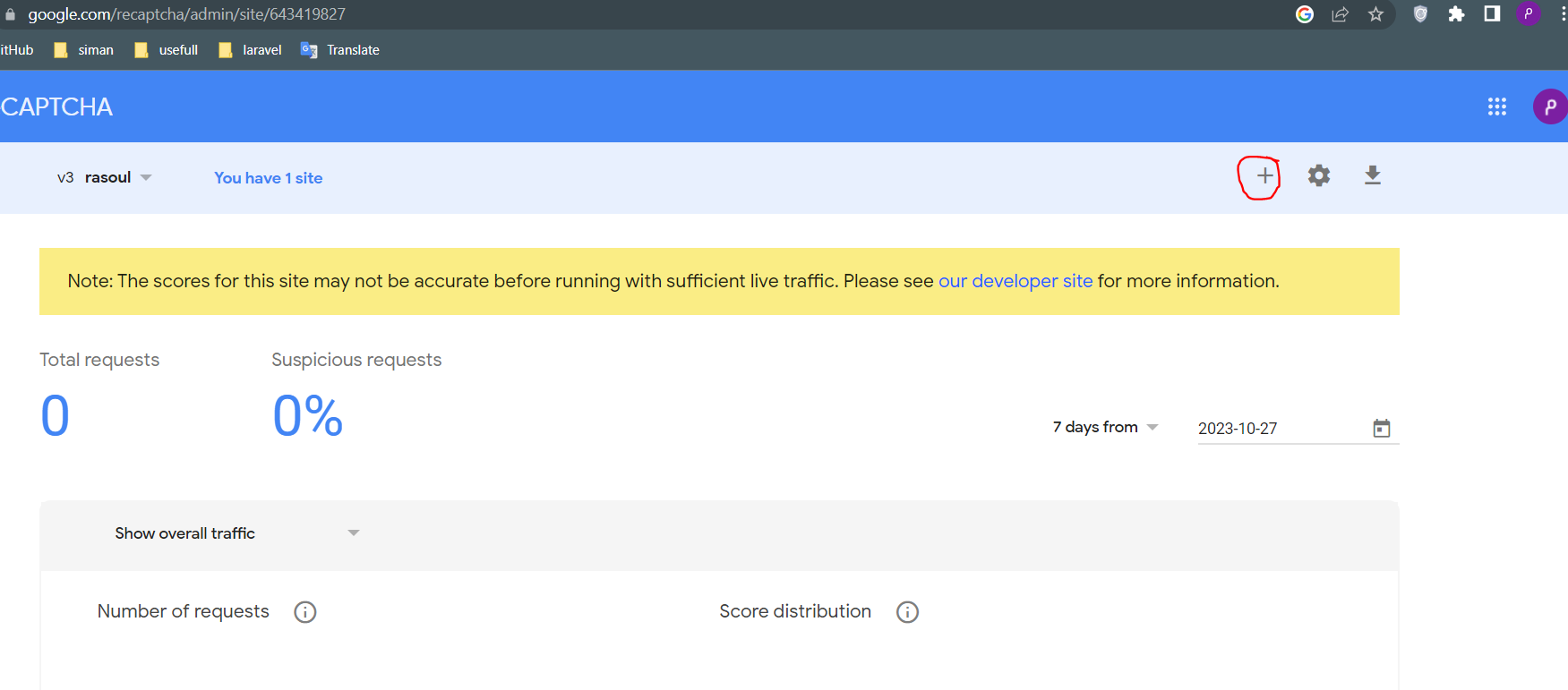
# Methods

## Google recaptua

Api authenticaton : serch google recaptua admin



Googel has a documention for this work but we use a laravel package for more convinence.

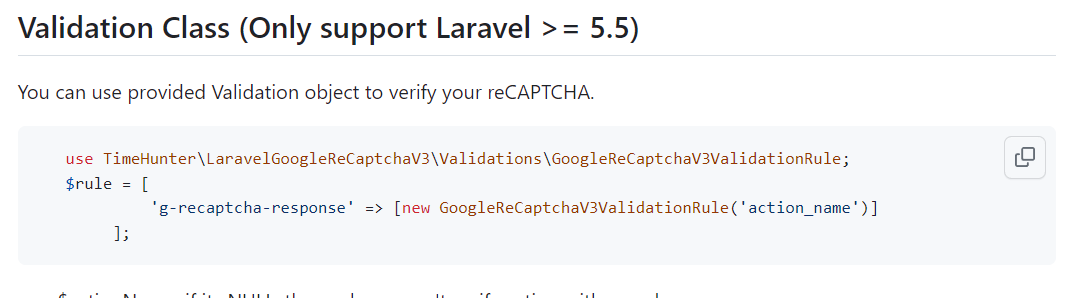
<https://github.com/RyanDaDeng/laravel-google-recaptcha-v3>

Ryan has both version 3 and 2.

### Version 3

Tip- threshold is been 0-1 and 1 is most secure way(I rather 0.9)

Logic and steps:

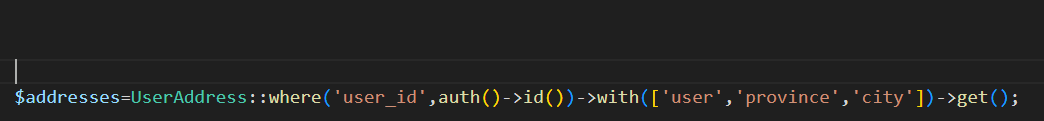
1. We do installation and configeration(api infos)
2. For any form that we want to be check we should create a array of some infos in config file of package (setting variable).(façade will do the same thing but I rather config file)
3. Now we go to form and action part:we connect our form to action that we wright in config file.(I rather renderOne())-after this,tha recaptcha badge will be displayed in the corner
4. Now by sumbiting our form there is a input contain a token and we validate the recaptcha input(like other inputs of form in $req->validate()) acording to te github giude. 



Copy paste rule badged text in laravel validate

1. As we are validatig we should specify a place for display a error(act like it’s a normal input and user @error or..)

## Codes that I get interested to:

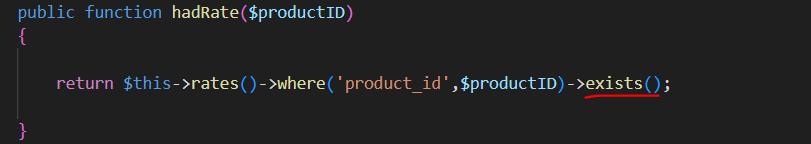




We can use arrays of relasions



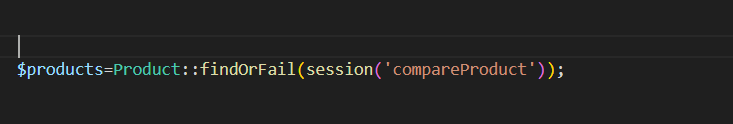
Replacement of first()!



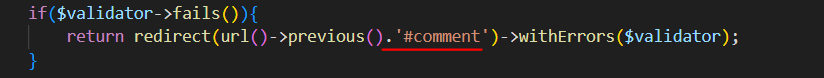
**pluck**



Find() can get an array to return.

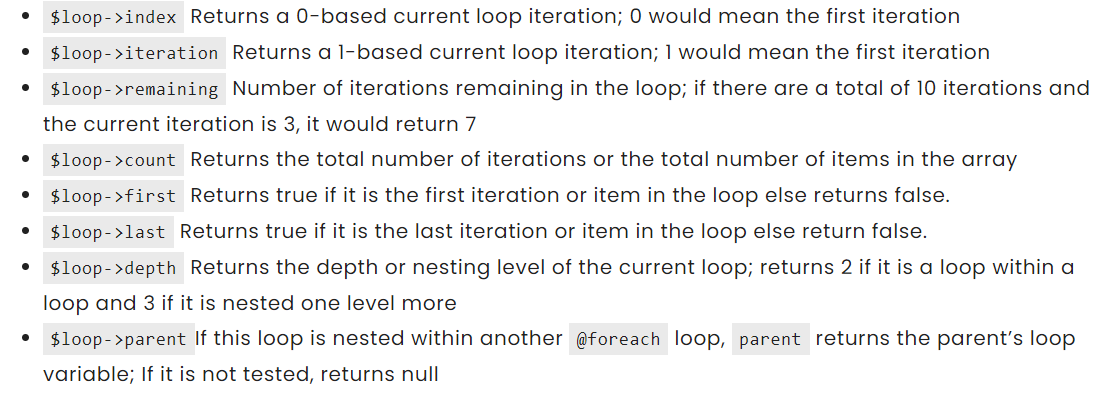


### Set costom redirect with parameters



### $loop:

I did not check but it seems only work on blade foreach



### How to add a php helper file to project?

After setting file in a directory in app directory define the file in composer.json:



Now tell to composer do autoload again: composer dump

## Payment

Logic-send part:

1. we send some infos like amount and callbadck url and etc to a payment gate via curl(or some other thech)
2. we get the answer right away that if there is no error we get a token
3. fill our tables base on policy(oreder table,order items,transactios),token should strore in transactions table
4. after that we return a url+token so user will redirect to gate payment page

logic-verifying:

1. after the payment page, gate will redirect our user to url that we determine for it
2. now in controller we send again a request (base on the token that we got earlier) to gate and we receive infos about this transaction
3. we must get status and tracking code(ref\_id)
4. now we update earlier tables transaction(ref\_id and status- where(token,$token)),orders(status)
5. subtracting the quantity sold from quntity on hand(use \cart to get the variation)
6. clear cart
7. redirect where ever you want

programing architecture:

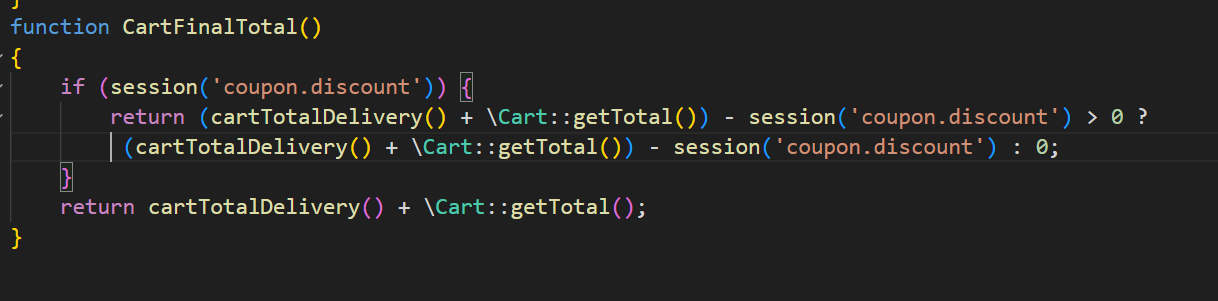
* if our gate using classic php and have no library for laravel its beter to build a [class](#_Biuld_a_class:) for it
* do table storing as a method on a [payment](#_Biuld_a_class:) class that you built your self

## Cart:

Our cart package: <https://github.com/darryldecode/laravelshoppingcart> episod(146)

Logic: we use sesions to store products so user can do some actions without authentication

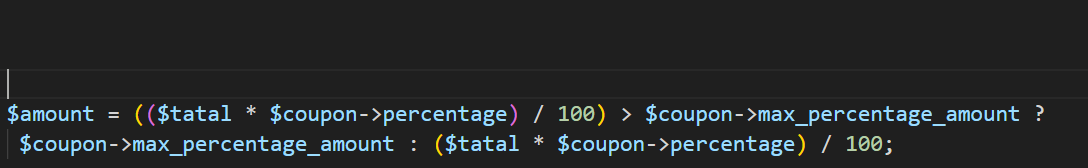
Tip-its beter to do price stuff in functions because all we need are in sesion and we should show them variaious places like picture bellow



### Coupon:

Tip-there is two type of coupon:

* Amount(70000)
* Percenage(20)



Percentage type

For coupons :

* First via a form we check the validation of code(expire,existance,usedbefore in an order)
* socond we run codes to get to discount amount base on our policy
* Store some info in session(couponID,discount,code)

### Get:

\Cart::getContent()

{{number\_format($item->getPriceSum())}}

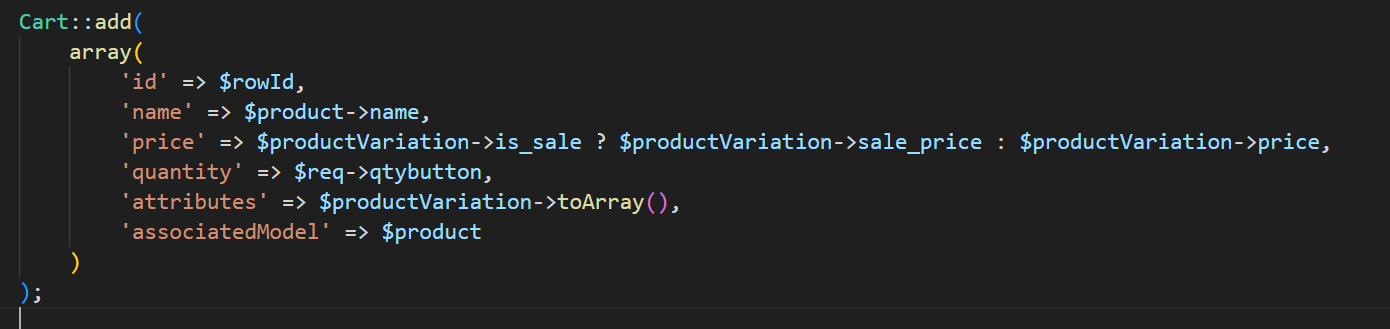
\cart::getTotal()

Tip-getpricesum: times quantity in price for a item,

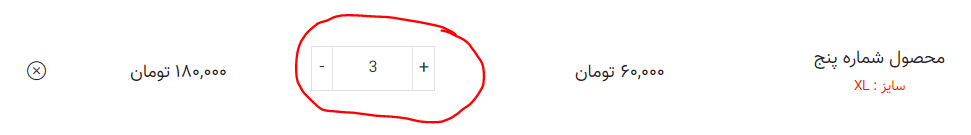
Tip-getTotal():its kinda the sum of getSumPrice of each item in \cart.

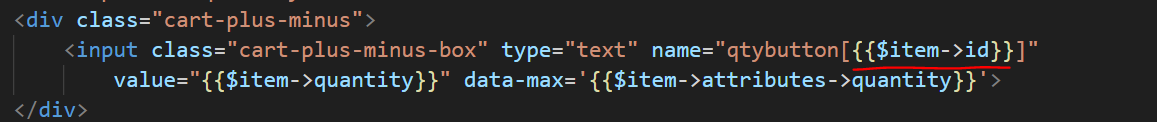
### Add:

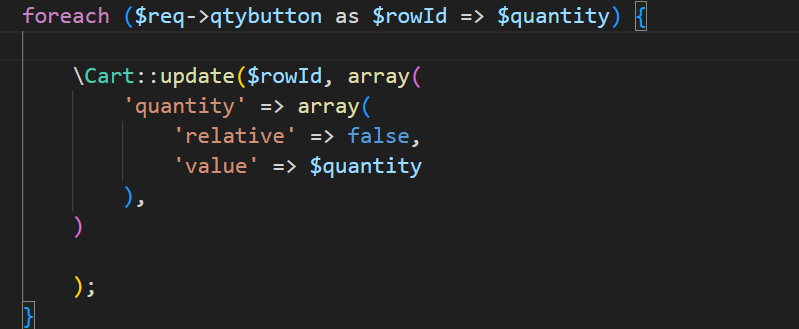
Tip:id should be uniqe so we cant attach ids of product and variation via(‘.’).



### Update:







Replace value, in defaul it plus the value

### Remove and clear:

$id is $rowID when we did add method.



\Cart::clear();

## Filter:

Logic:alwase for filter first we get collections of a model($products) then we filter and choose between them.

### 1-filter for a dress store:

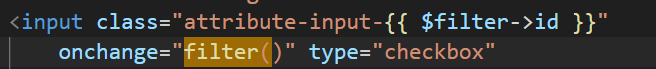
First for any kinda filter we always have a form to submit our form have no action to load on curent page.

#### Front:

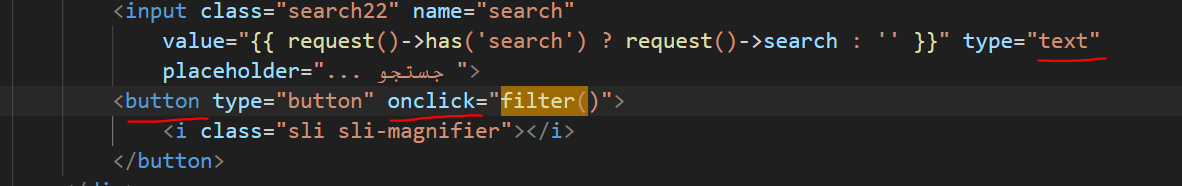
Episode:104-106



-In here we want by any changes on page a filter happen so we buld a js function and use onchange attribute event on tags to do the function:



consider for any type of input what event we use



Socond: we need hidden inputs destinations,for having a good data face for working on ,in backend

Our desire input form:we have as many as want attribute so make it uniqe by attach id.



In asset there is filter directory for review how to do second part to check jquery codes.

#### Backend:

For rendering filter request we build a scop to be able to customise our query.remember the only good way to build a filter backend is with scope.



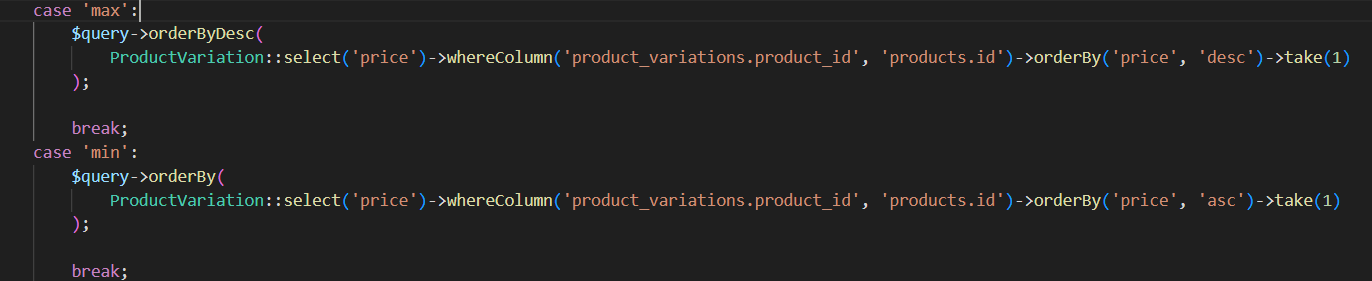
Relation that exists in product class

Logic-above:We have to use whereHas because in query we have a lot of products that are from a category so we are not face with one product to check its variation with where method we should use whereHas to directly put the where on target.

[Clik](#_whereHas_and_Has:) for more about whereHas

-for sort base on price, as price is in another table we use bellow method:

Its kindalike whereHas because we actully need to evaluate another table that is in relation with us.



We only can use take

Above code only work on ordering products it has no power to filter and decrease count of products.

-scope codes are in assets.

### Scope

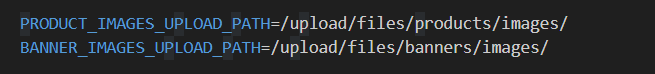
Syntax AND name rules:





## Upload file

### Initials

1. Define a path in env file: 
2. We need a helper function to make the names unicqe,to attach time to file name:

use Carbon\Carbon;

if (!function\_exists('generateFileName')) {

    function generateFileName($name)

    {

        $year = Carbon::now()->year;

        $month = Carbon::now()->month;

        $day = Carbon::now()->day;

        $hour = Carbon::now()->hour;

        $minute = Carbon::now()->minute;

        $second = Carbon::now()->second;

        $microsecond = Carbon::now()->microsecond;

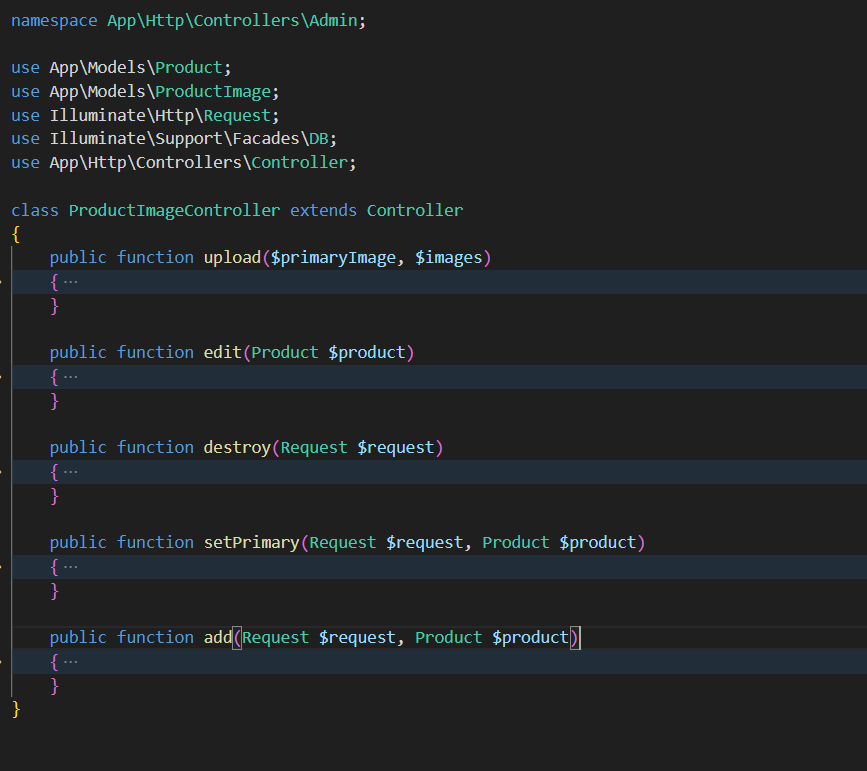
        return $year . '\_' . $month . '\_' . $day . '\_' . $hour . '\_' . $minute . '\_' . $second . '\_' . $microsecond . '\_' . $name;

    }

}

1. If we need to upload image in more than one time its beter to create a contrller to make it easer:

Click [link](#_Use_methods_of) to tutorial of object and controll tips.



### Upload:

It only have three step first make a uniqe name second use move method and then uplaod file name on database:



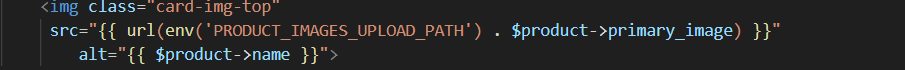
### Delete

Via classic php:





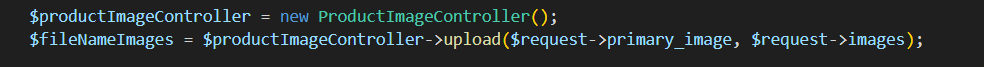
### Show



## Work with objects and classes

### Use methods of a controller to another:

First we simply make a new object of our class,second we can use its method with our parameters:



There is a [reference](#_Initials) in this document

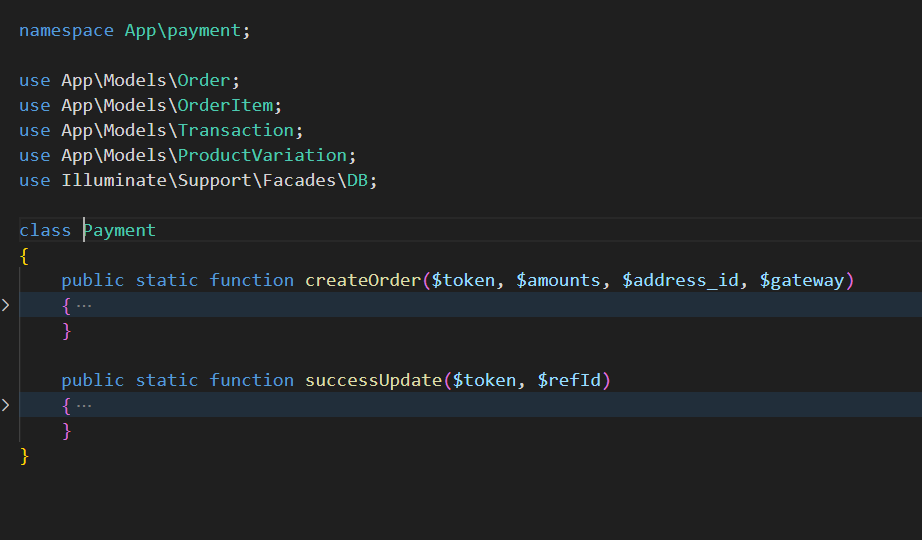
### Biuld a class:

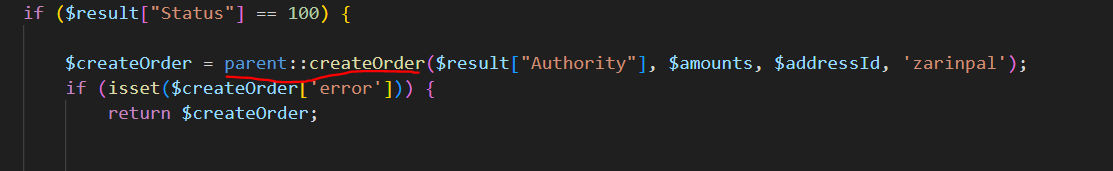
Our example is for a gate payment codes.

* create a folder in app directory
* create php file in folder
* write the namespace and then start methods writing



Its father is right next to him- in same namespace



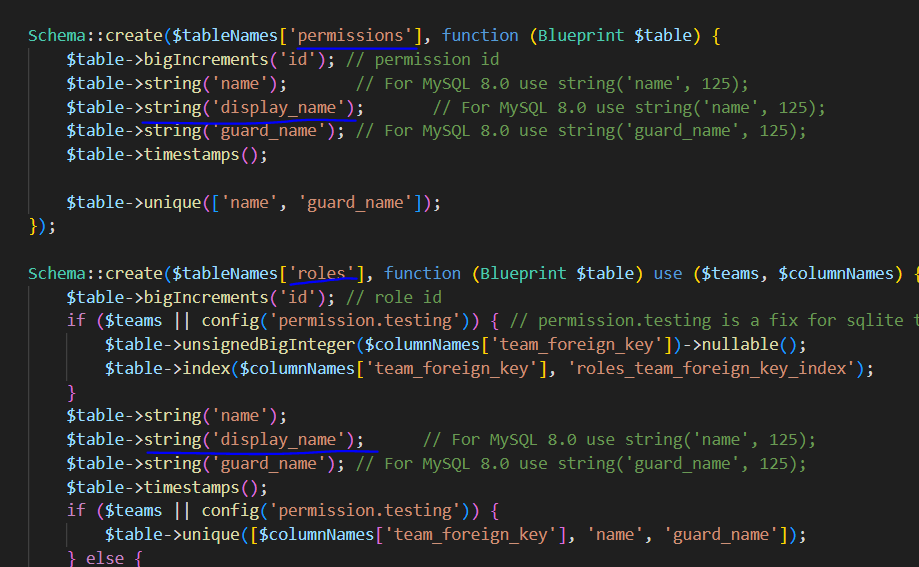
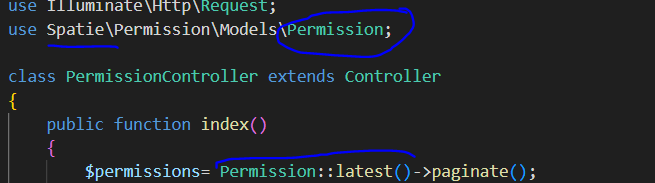


# Authorisation

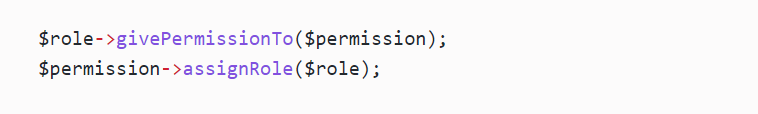
Tip-I rather role and permission policy for this work.

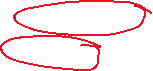
We use laravel permission package : <https://spatie.be/docs/laravel-permission/v6/introduction>

Logic and steps:

* go to installaton in laravel link and do the steps
* this package has 5 table that only with 2(roles and permissions) of them w’ll work directly for that we have to publish migration of package
* after publishing package migration file before to run migrate command I rather to change migration file and add display\_name(for both permision and role tables) field because we are persian and it has beter convinice
* user model should contain package trait
* we don’t need role and permission models the package has them we can use this models like a normal model plus more usefull methods. If we want to get permissions,just think that is a normal permission and do what should. 
* first we write and fill permissions that we have then we give the permissions to a role,we create a role then by using package method give permissions to a role(this is plused method that I mentined and they are in its docs)

tip-for signed place bellow it is beter to be an array of names because the package did mention use name but it work on ids





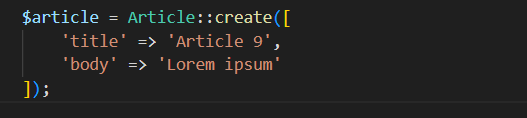
Tip- if we create permission and role model by adding hasRole trait to them we can use package classes but its beter that we don’t build models for it and use packages models.

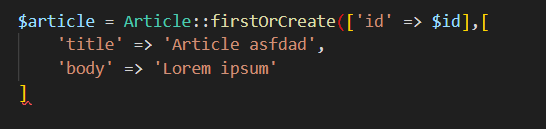
Tip-when we using directives to show a part of view to user or not its beter only use @can and pass to it permissions because it will do direct permission to a user and even accept if permission is on role(I explaind bad but if I go with permiison there is no need to use @role)

# DB

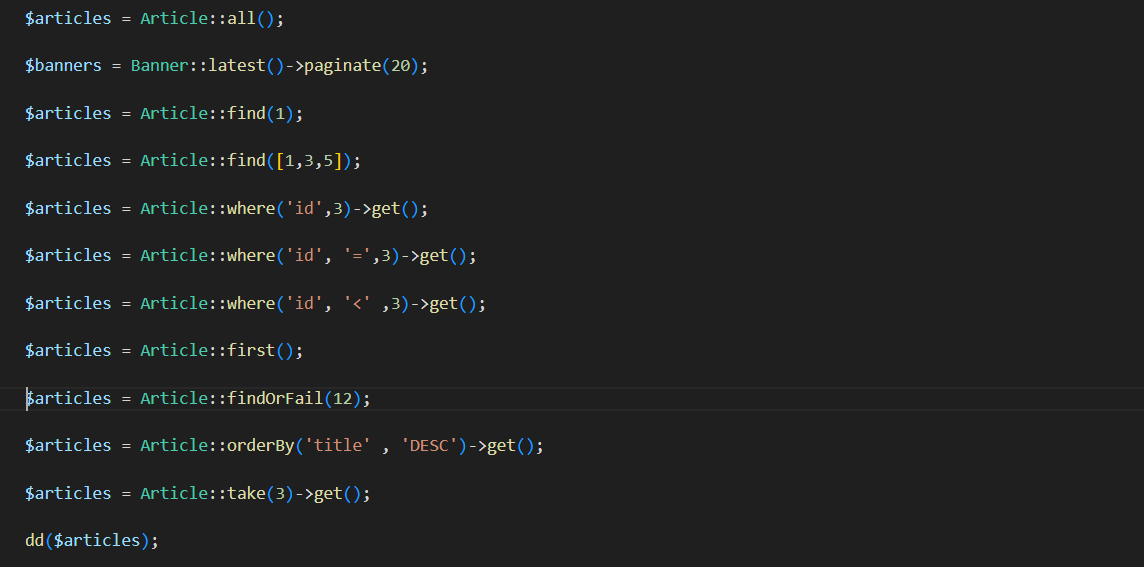
## Crud:

Insert:

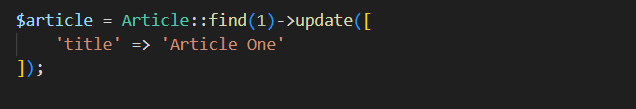


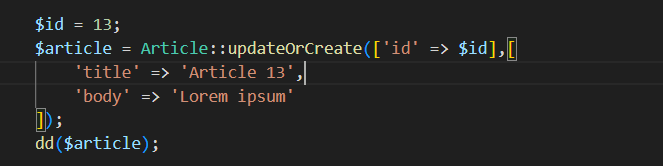


Select:



Update:





Delete:



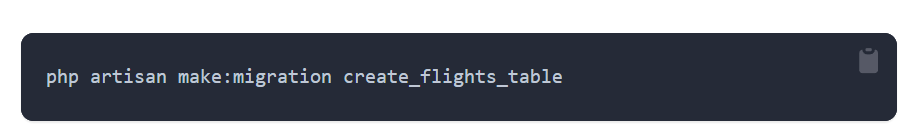


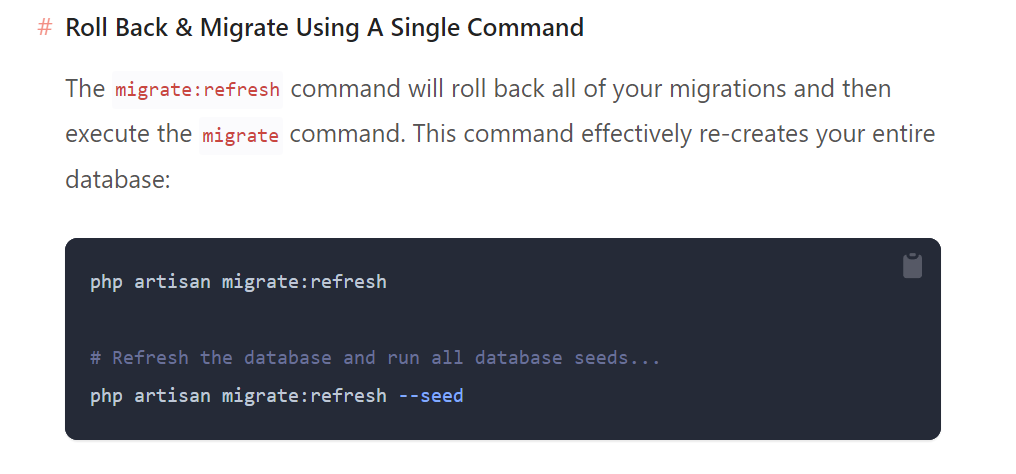
DB transaction:

----------------

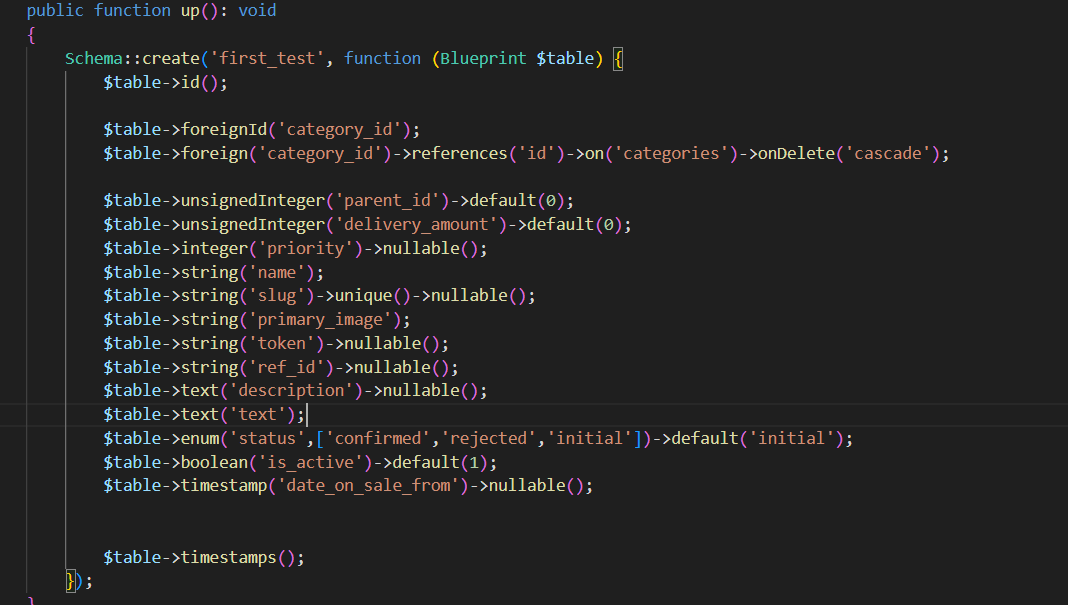
## Migration

#### initial

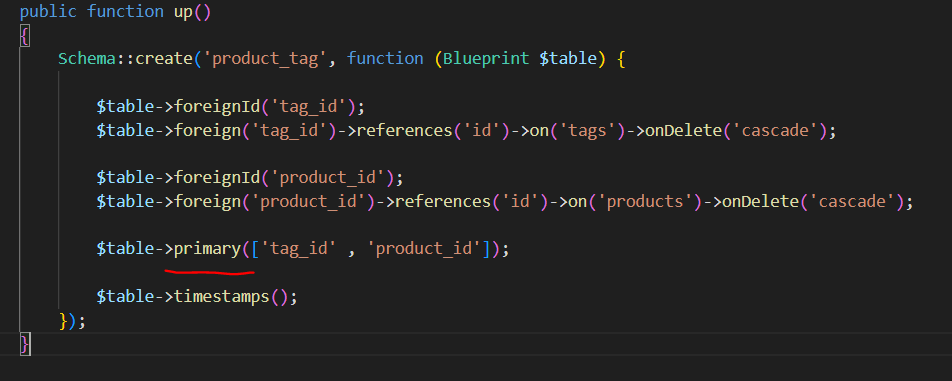




#### Schema tabel



#### pivot table schema

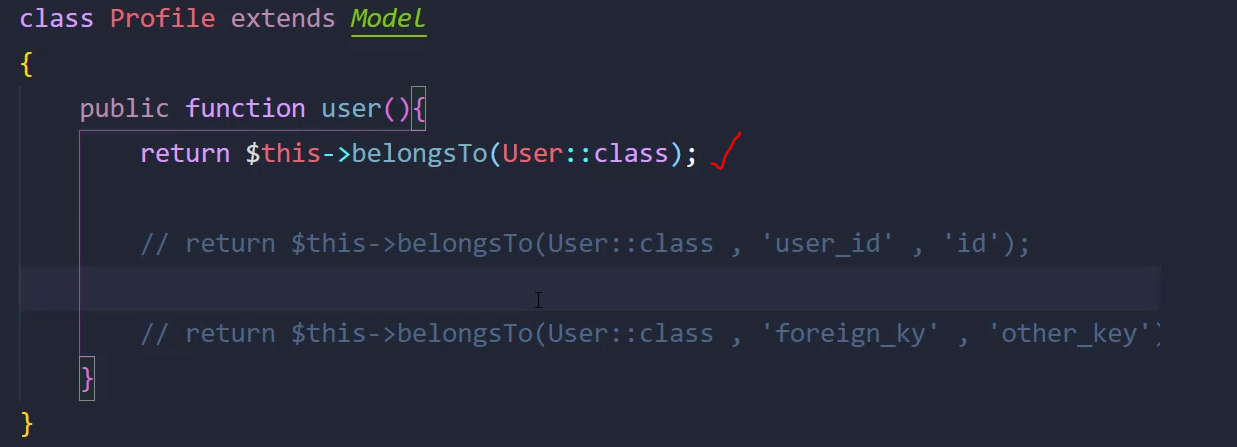


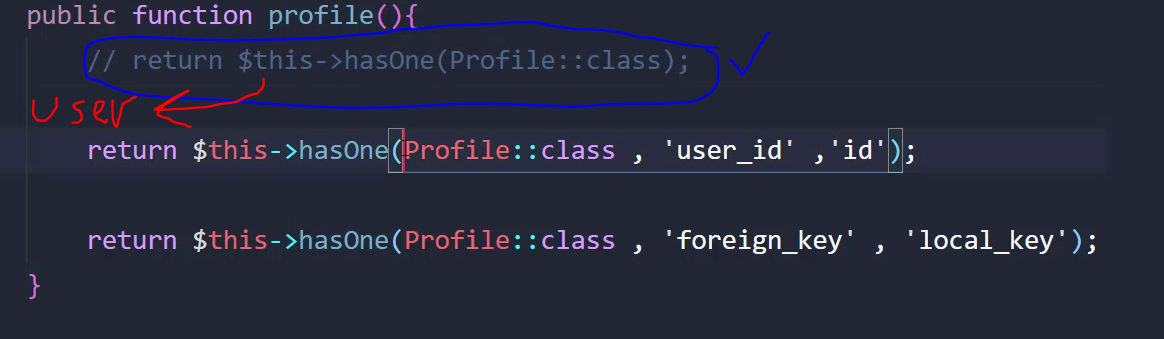
## Relations

**Note: every time we say belongs that mean in orgin(our object) there is a forigen key that belong to somewhere,**

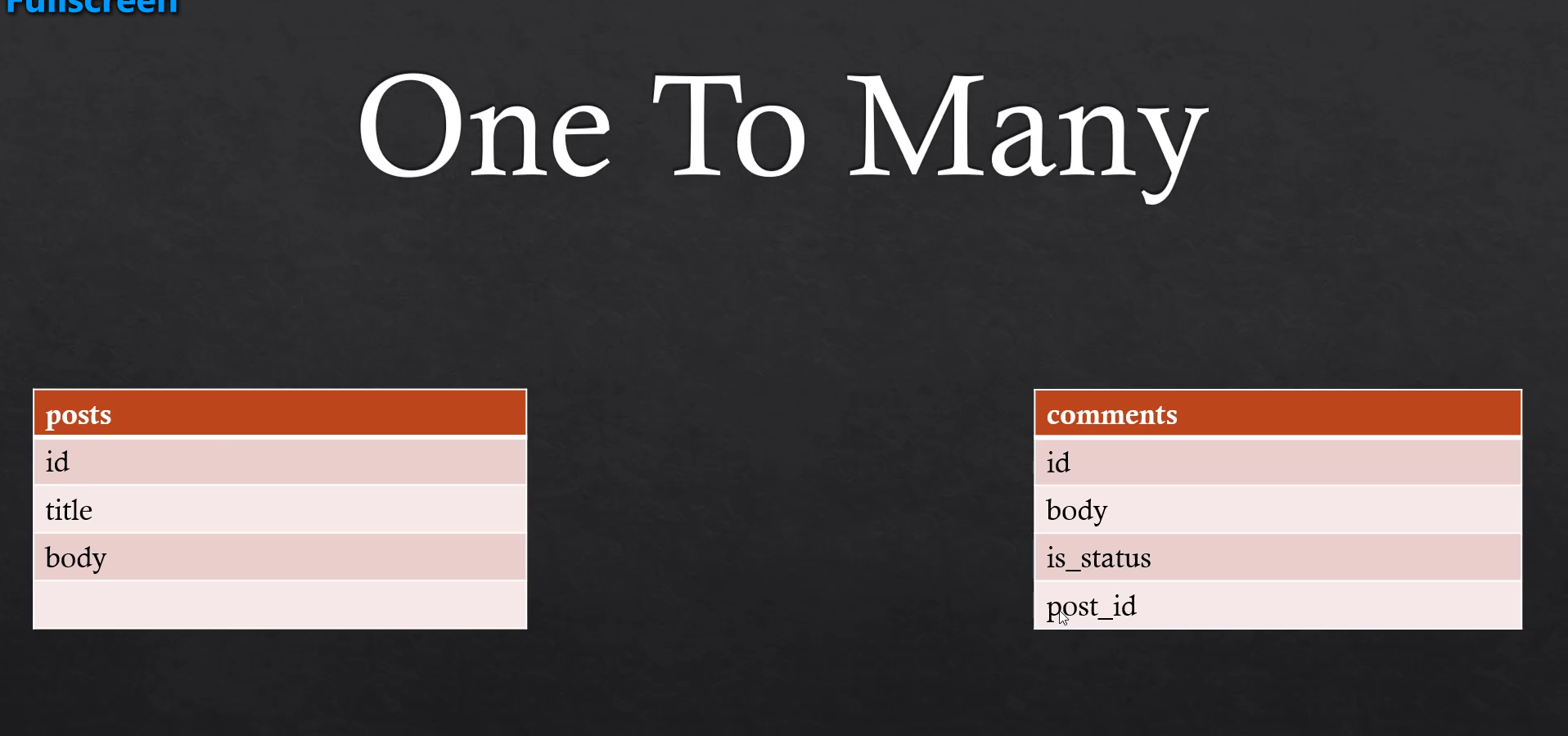
**And in other way when we say origin has that mean its forigen key exsists in destination.**

#### One to one:

Relation between user and profile, 



#### One to many:

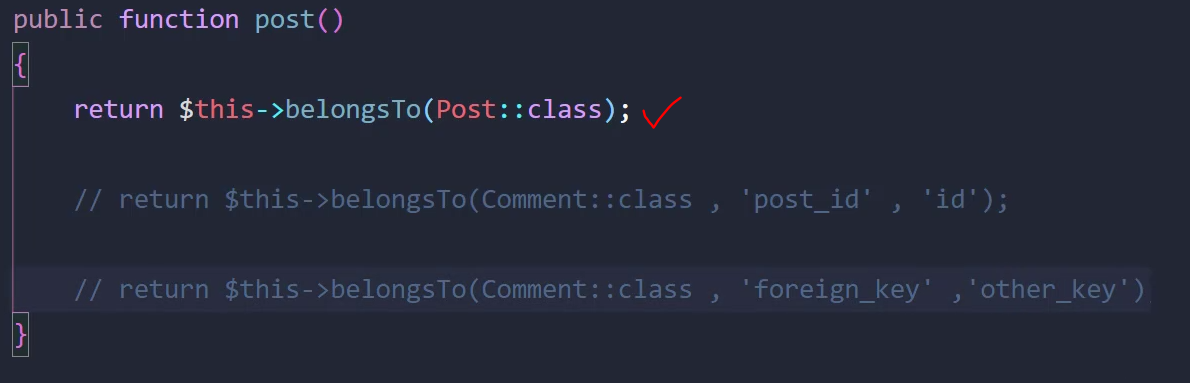


Any post has many comments,

Post foreign key is in comment so we can say

Post has in comment.



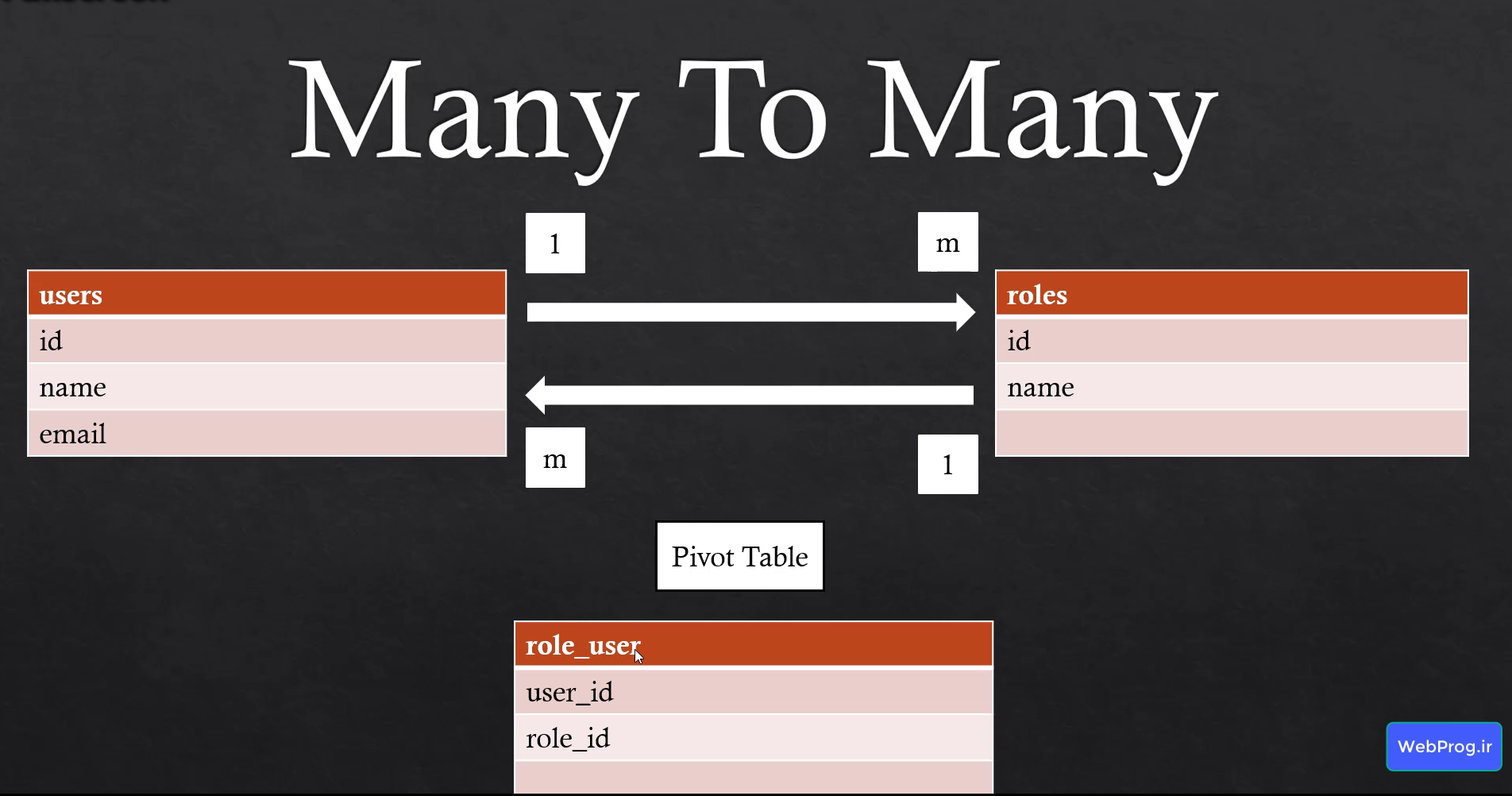


#### Many to many

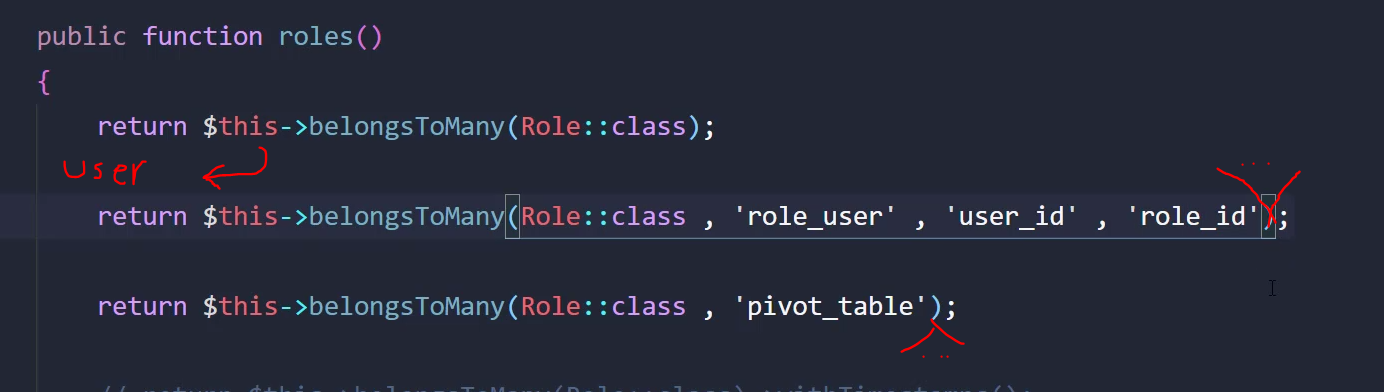
Laravel standard:

1-words are selected in alpheba order

2-words are single



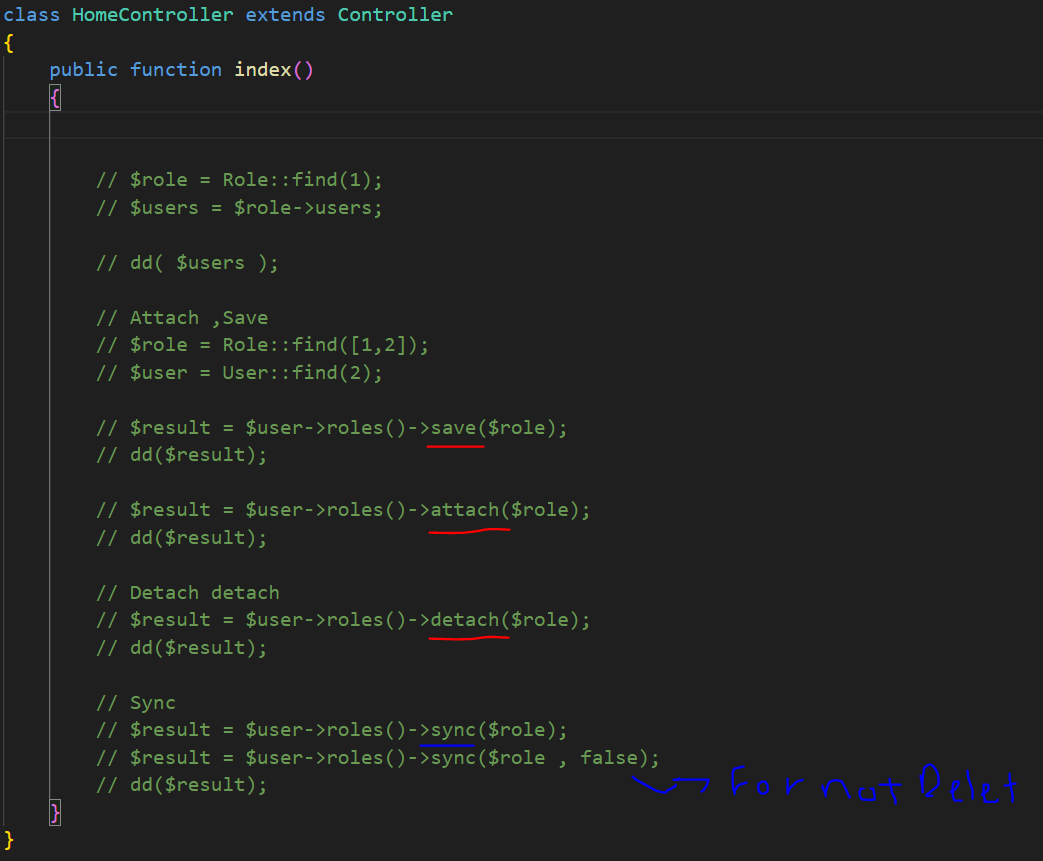
The folowing relation is equl in role and user mode



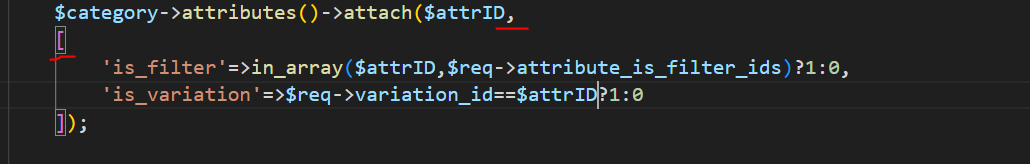
#### How to add to pivote tabel

As we don’t biuld role\_user model(if we want we can have one) we can use the relation between to table to add to there pivote.

Save and attach are same ,and detach is obvious but sync is usefull because first it delete orgin rows(on below code any roles that belongs to user number 2) on pivot tabel then do attach that it fix malfuncton of duplicated rows.



How to fill additional colums?



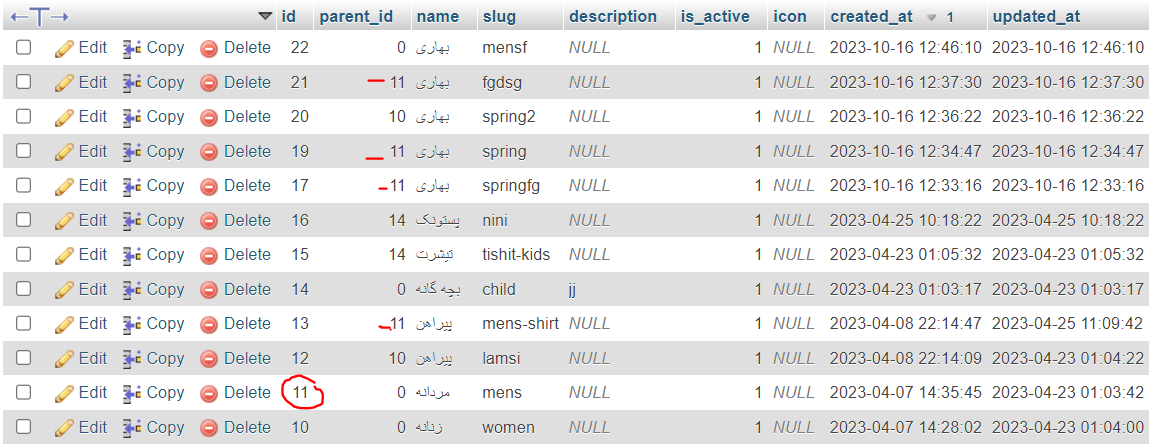
How to delete rows belog for a relation from an oring:

$categoy=Category::find(2);

 $category->attributes()->detach();

## Model class methods

1-a cool thing is that we can make a relation between one table, find children of a row:



Logic: consider we have a category row and we want its children,we need to see which rows has its id as parent\_id.

class Category extends Model

{

 public function parent()

    {

        return $this->belongsTo(Category::class, 'parent\_id');

    }

    public function children(){

        return $this->hasMany(Category::class,'parent\_id');

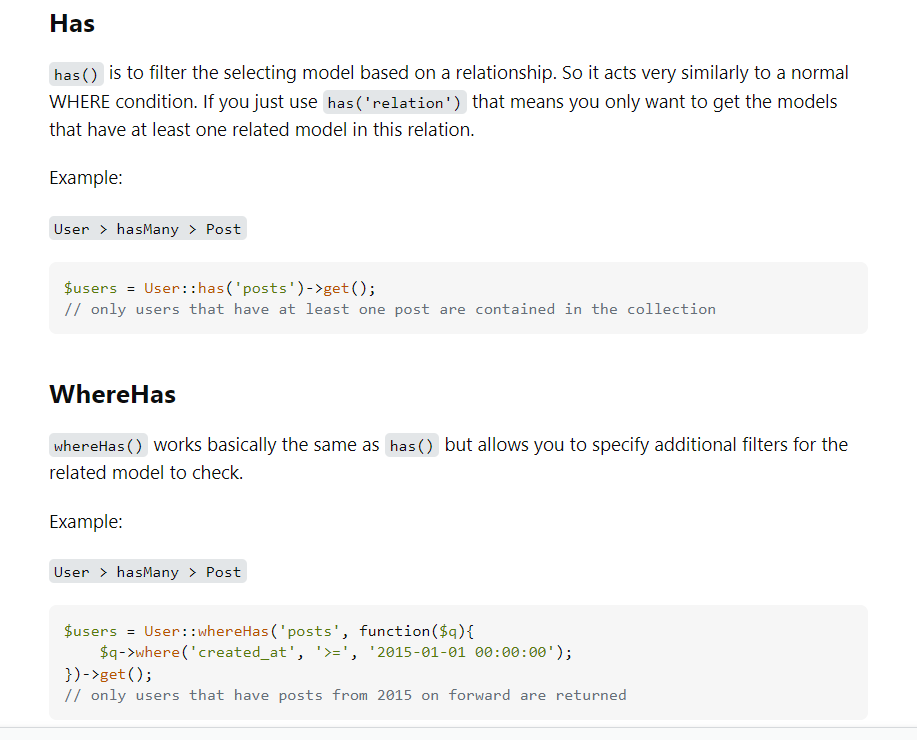
    }

}

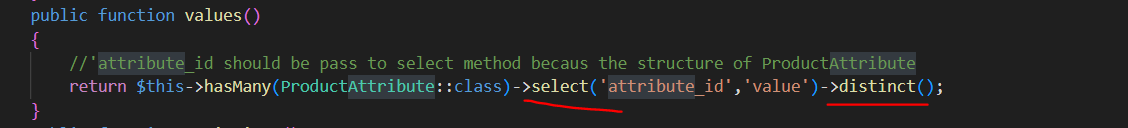
As you can see origin and destination both are in category model class

#### whereHas and Has:

these are for when we have some collection of the orgin model but base of a model that are relation to orgin we filter and decress collectios



#### Use select method to distinct outcome:



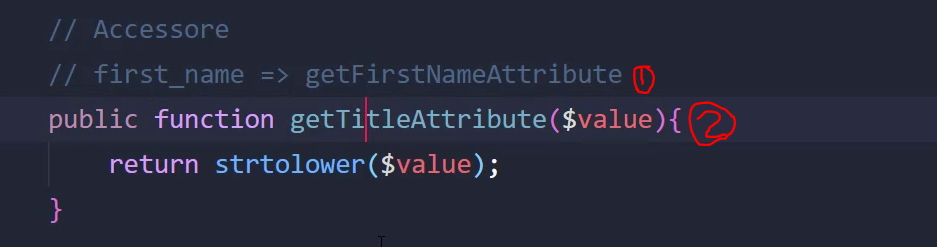
For distinct to happen we must to say in select method what $this has named in product\_attributes table.

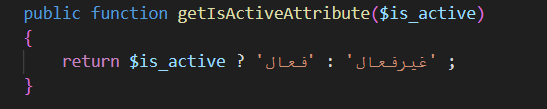
#### Accessor

1->Standard define name; 2->from title word value parameter refrence to title column;

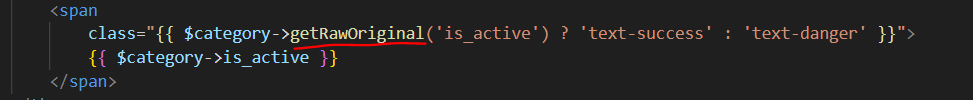


Note: if we want to change the raw outcome of a coulme we pass the name of the coulmn as a variable like the example bellow:





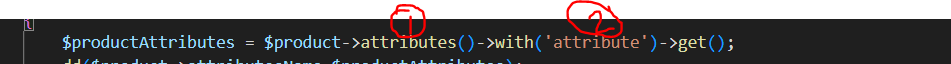
On view:



The accessor will change the original value of column,to discard accessor we can use getRawOriginal

#### **With method**

With methods receive a relation from to add it to a query that actualy going.

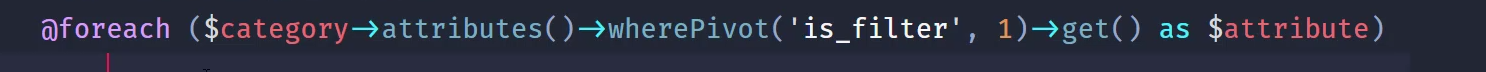


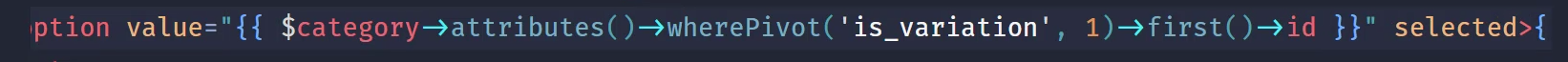
Loginc: 2 is a relation that exsists in model that refers in 1 relatoin.



Logic2:if we don’t use with method above code stil will work correctly the point is that with,with method after above code there is no more qurey to get the datas.

#### **wherePivote**:

bring me the attributes that are related to this category but first check the pivote table and bring only filterable attribute for this category.

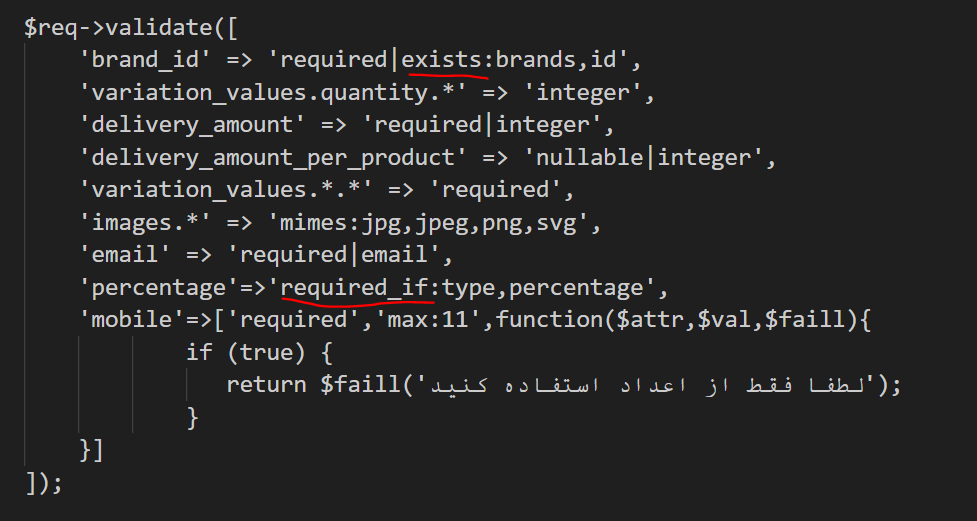


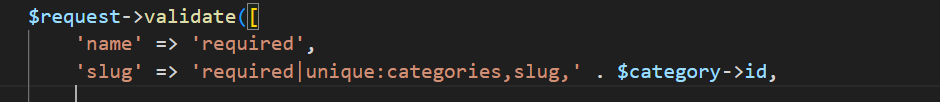
# Validation

## Rules and syntax

Some usefull rulles and syntax:

Tip-require\_if ->logic:it says if amount of type input is equal to percentage I’m required



When we add an id as third parametr to unique it will discard that, id on <<categories>>table

Its too usefull when we are updating tables.

## Withbag

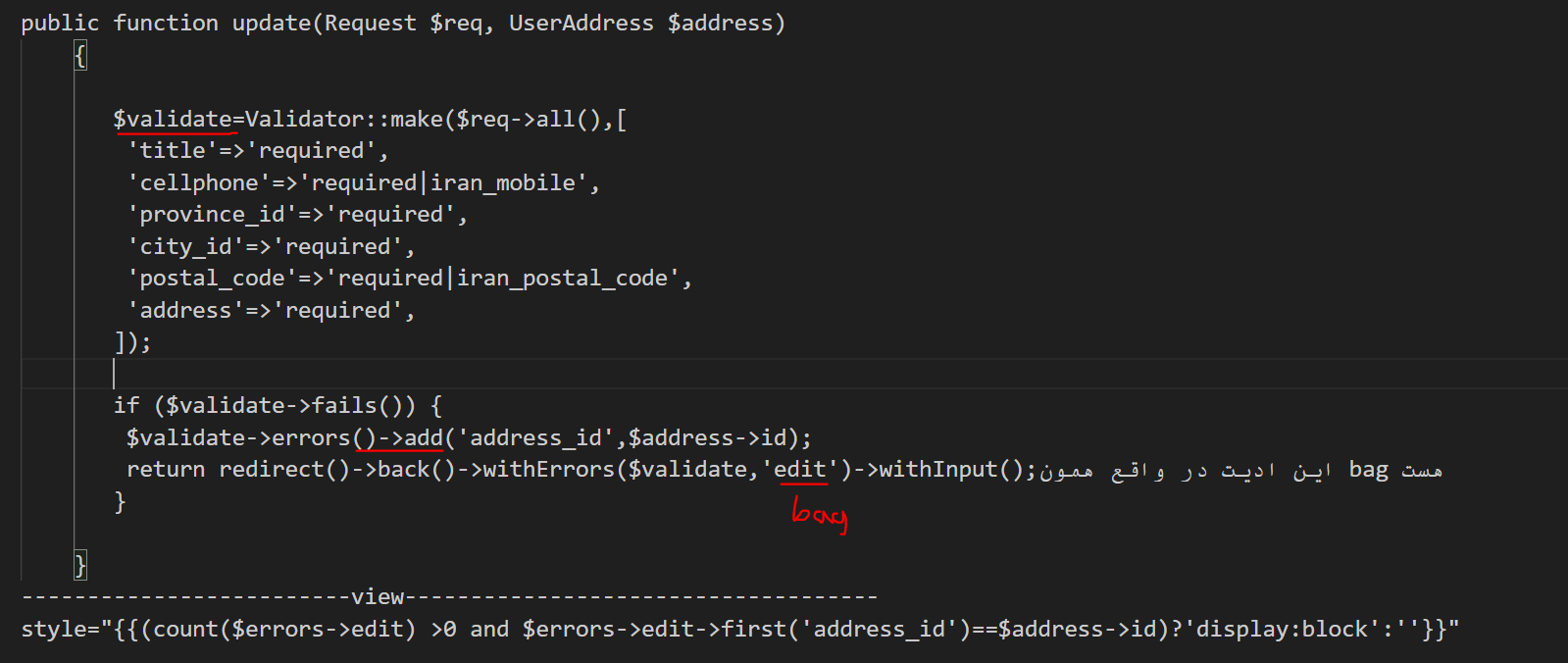
Bags will define in backend and its only for error handling.



Tip- look how bag is on error:



## Validator Façade



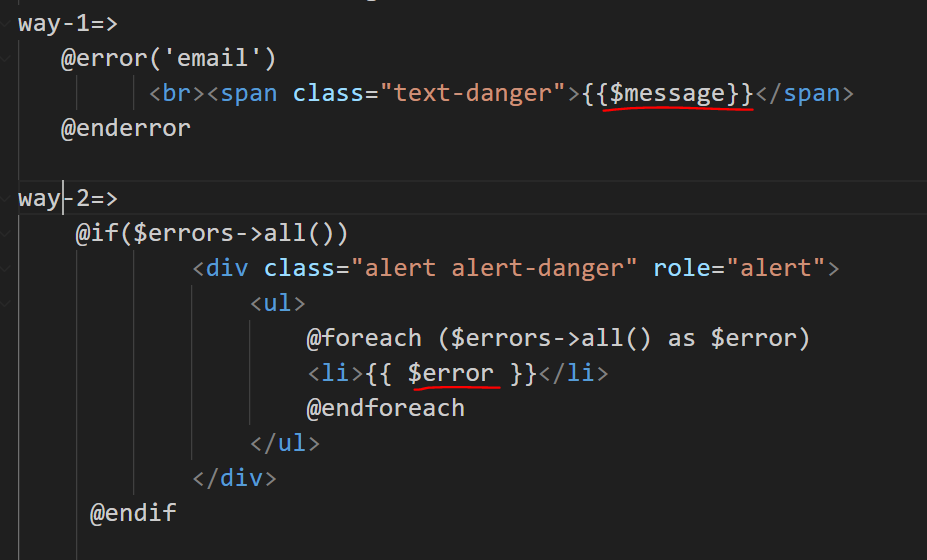
$validate->errors()->get('address\_id')

$validate->errors()->all()

## On view

We can use a blade file and include this whenevey we want:





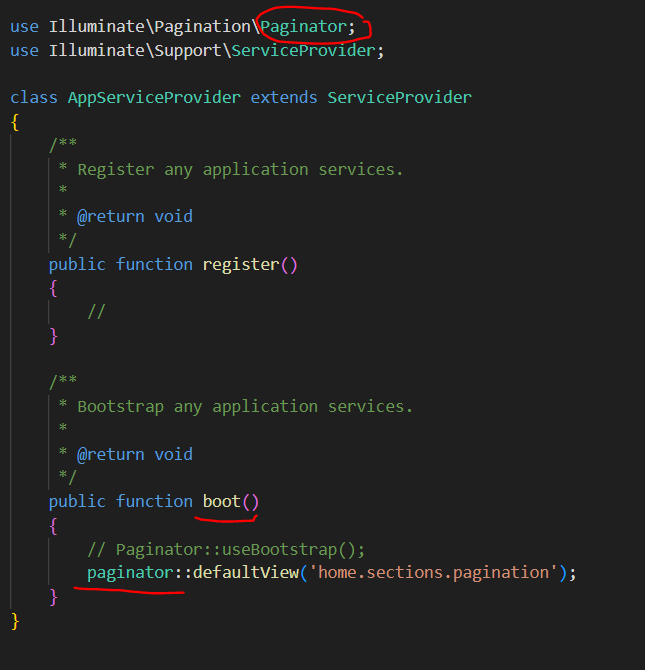
## Iran validation

<https://github.com/anetwork/validation>

# Pagination

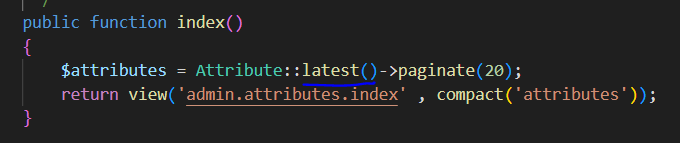
### Instalation

We build a custom blade to be use as buttons for next and previous .then we will added to AppserviesPorvider:



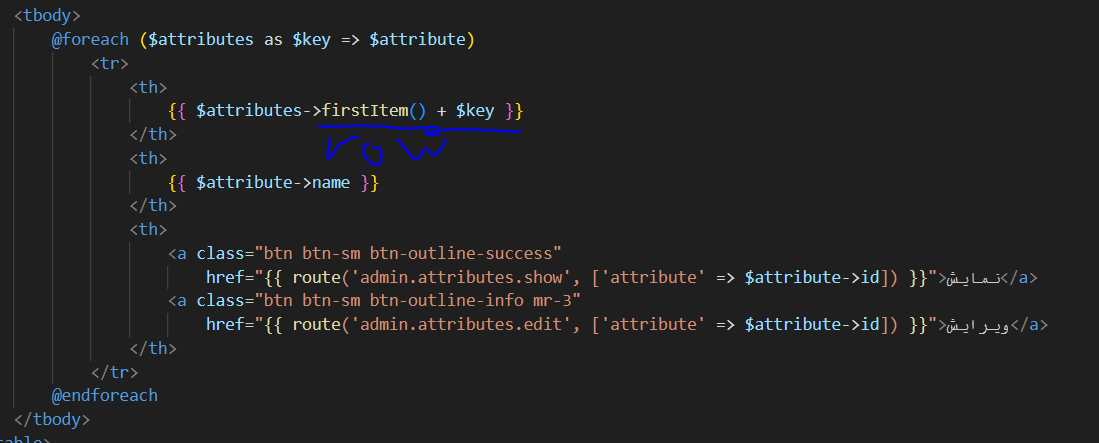
I have a sample of costom blade that style is based on bootstrap so the layout must had access to bootsrap library.

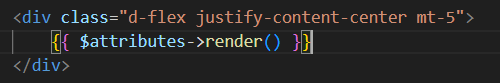
### Controller command



### On view

fefdefd





# How to program:

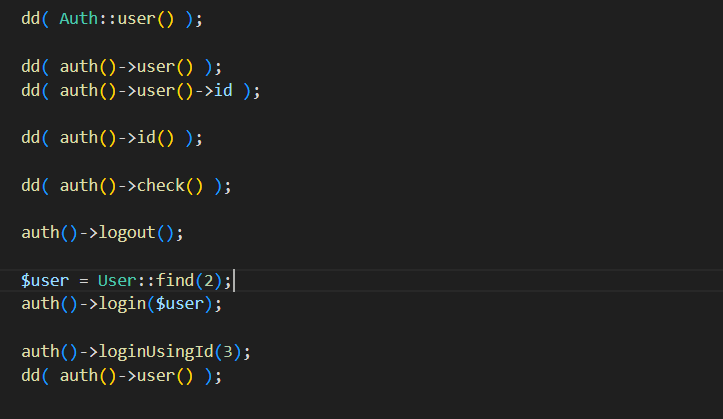
1-what is the target

2-what do we need/what is our parameters

3-bulid parameters

# Autentication:

## Auth helper function:



Second parameter:$remember=true

Is watcher of this page login?

## Normal via fortify:

-for normal authentication we use fortify that should be instal in project I suggest read the documentation.

-fortify only handle backend part of auth.

Instal and config:

1-

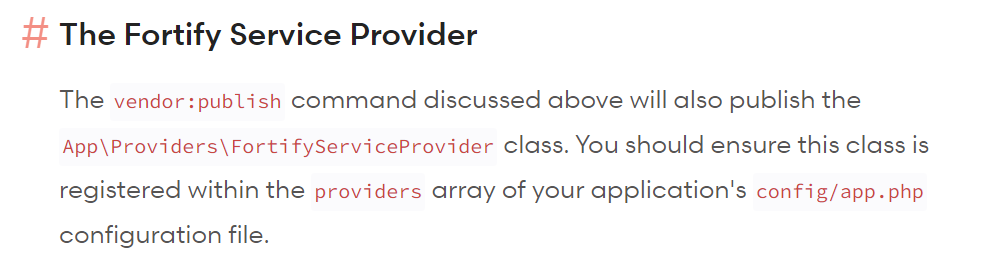
Composer require laravel/fortify

2-

php artisan vendor:publish --provider="Laravel\Fortify\FortifyServiceProvider"

This command will publish Fortify's actions to your app/Actions directory, which will be created if it does not exist. In addition, the FortifyServiceProvider, configuration file, and all necessary database migrations will be published.

3-

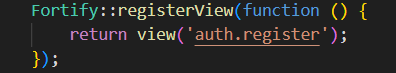


App\Providers\FortifyServiceProvider add this in config/app.php

Now we should check the features that we want, in fortyfy config file .with enabling the feature one of work that happen is to create related route.

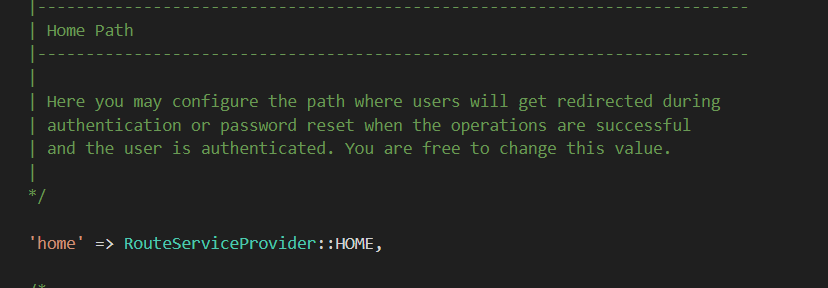
### Registeration:

Introduce view:



Form:

* Post endpiont(‘/register’)
* Name input
* Email/username input(in its config file we determine coulmn name that should check)
* Password input
* password\_confirmation input
* if you have **more** **inputs** or you want **costomize** registeration go to App\Actions\Fortify\CreateNewUser

Where do we redirect after sucseed register: 

Or ‘/mydesire/url’

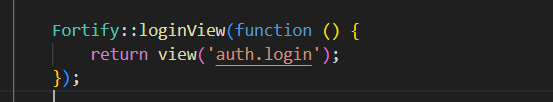
Config/forify.php

Config/forify.php

### Login:

View:

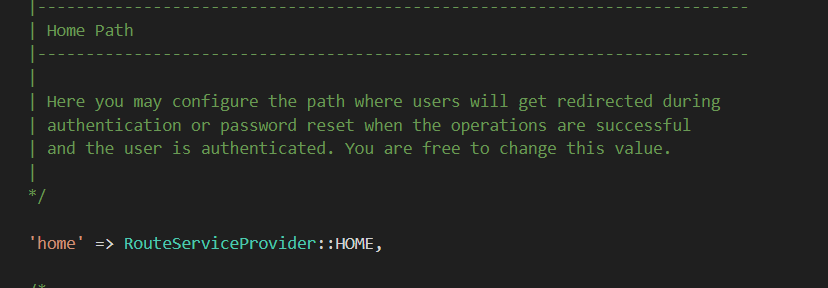
-In login to introduce view for url/login endpoint in forify service provider we add code bellow



-it is not realy important to introduce a view for login the point is that we only need a form to contains our needed inputs to be able to work on it in backend,

Form:

* Endpoint :post to /login.
* Email/username input (in its config file we determine coulmn name that should check)
* Password input
* Remmeber input (must contain boolian)

Where do we redirect after sucseed login: 

Config/fortify.php

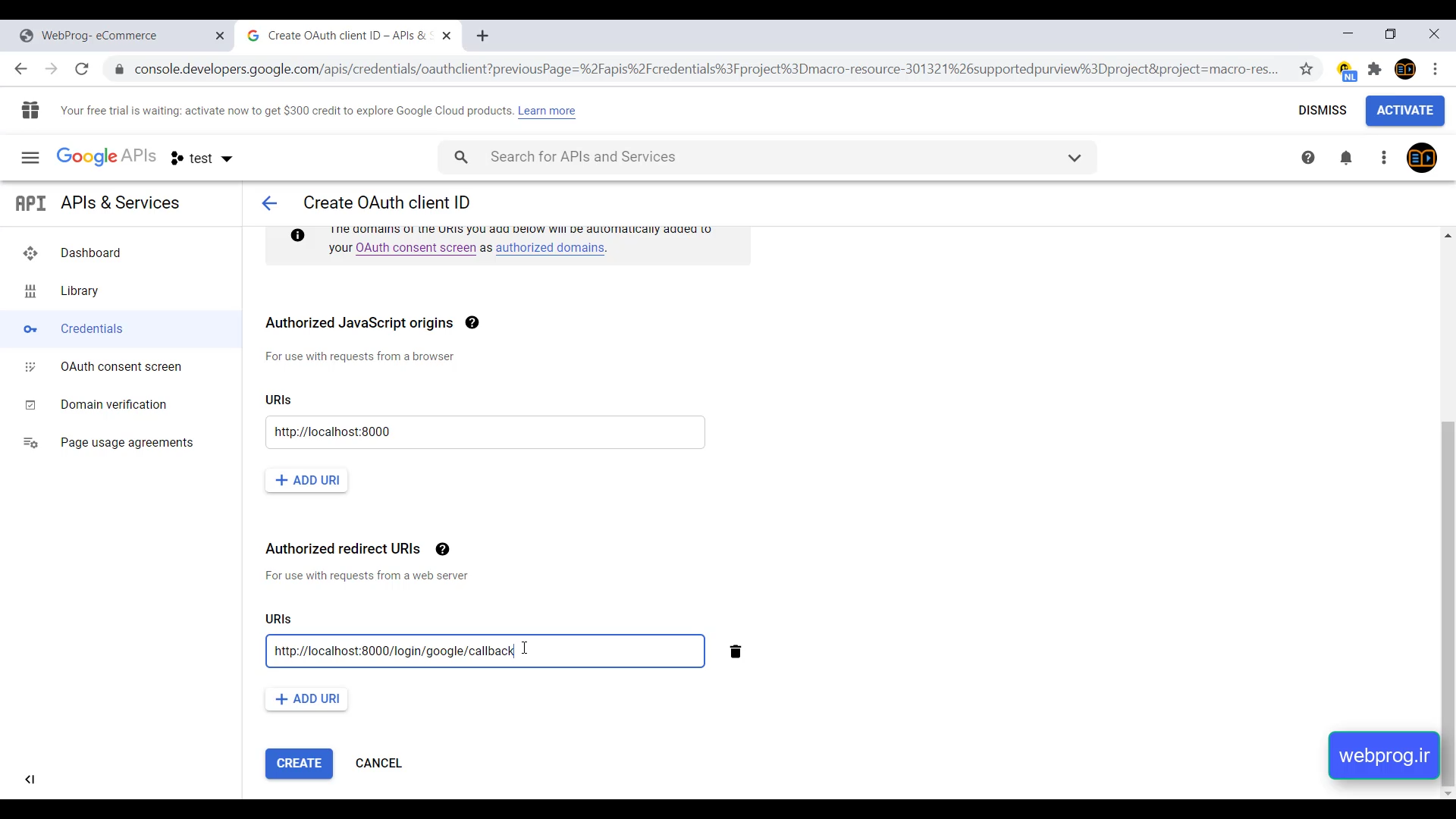
Or ‘/mydesire/url’

To customiz login credentials  reed documation.you can customize for redirect, go to documentation

## oAuth:

oAuth via google:

1- we must have a oauth acount in google console to get clinet id and client secret.

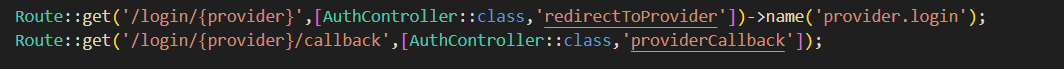


After console acount:

Laravel has a package called socialite for oAuth.

2- instal it by reading documentaion.

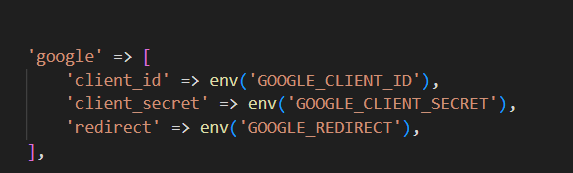
3- we need to create two route,one for sending request and socond for provider callback.



Give first route to related link buttons.for login or register both should go to this route and in backend we check if the user has already registerd or not.

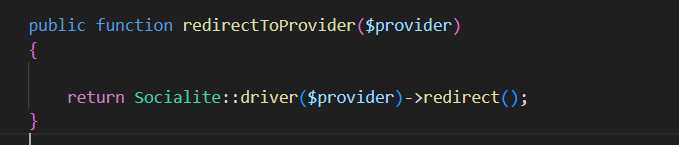
4- we have to add our providers to config/services.php a example for google:

Tip-our redirect index should be equal with what wa write in google console.



Serveces.php

5-redirect route : we just create the code bellow socialite do all the hard work



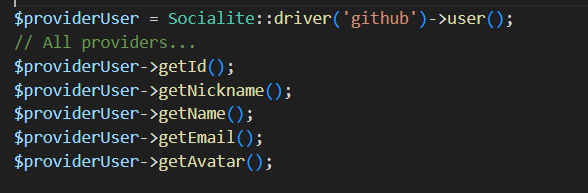
6- callback route:

First we get ansser of provider via socialite pakage.

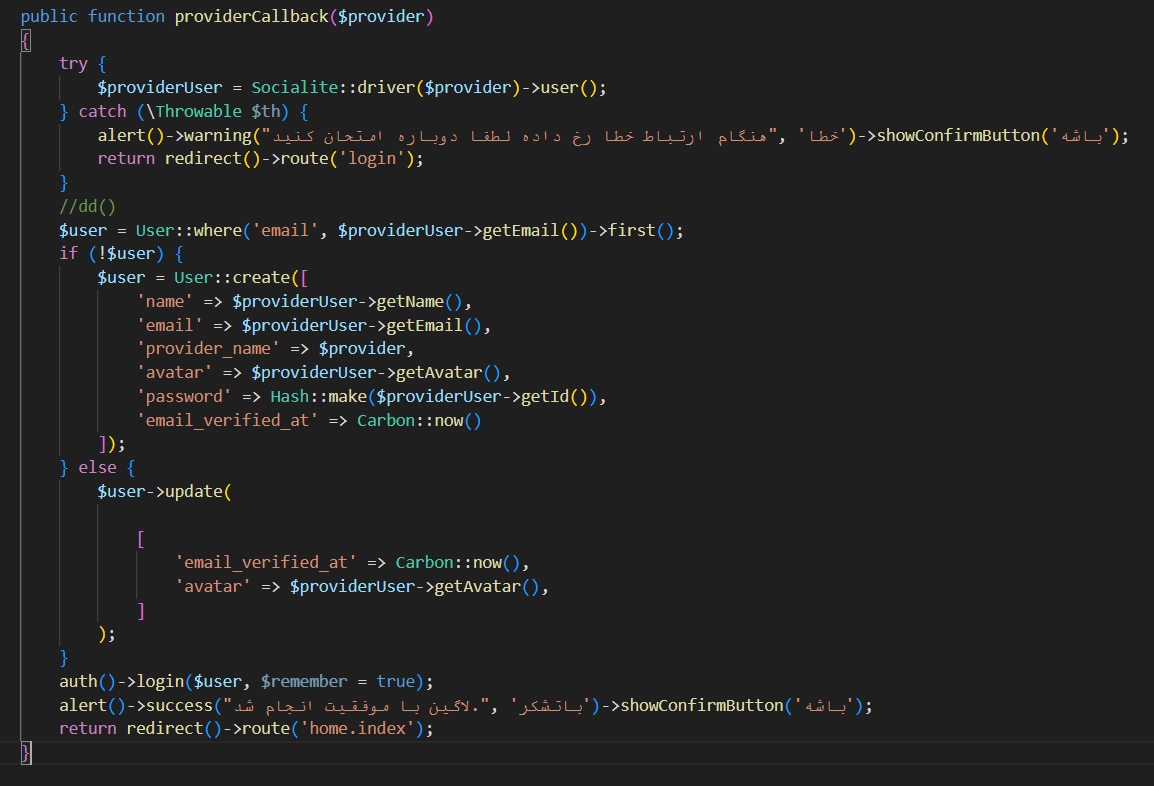
I’m not sure why we using try and catch.

After that by user model we check the user existence and we proceed.

-socialite parameter call back work on all providers:



Tip: in bellow example attention to verfication email,password,remmember user.





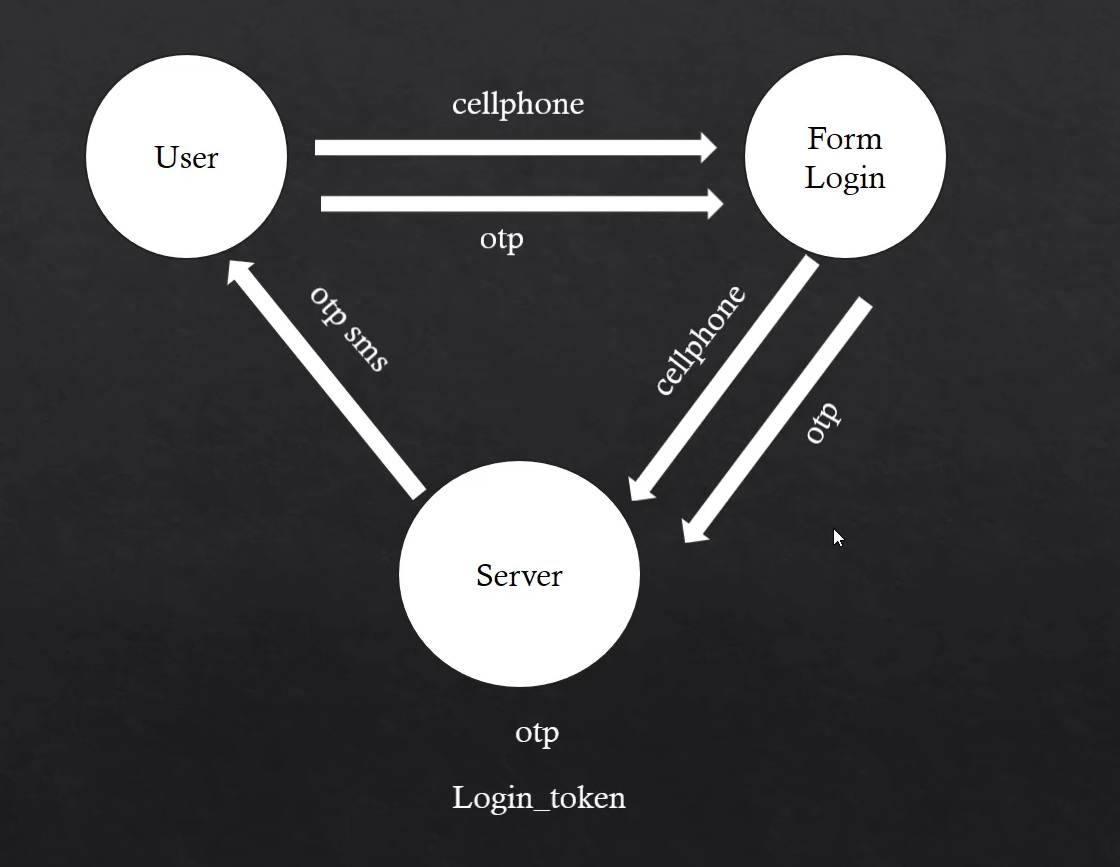
## Otp Autenticaton:

Login and register in otp is like oAuth,first we check that cellphone exisistence if cellphone is in table so we just update otp.

Gasedak sms service(episod 124)

path of picture bellow

1. We have a cellphone input and send it to server
2. We check the existece of cellphone if exist we update table and if not create
3. We make a random number via rand function base the digit we want(otp)
4. We need a uniqe random string(Hash::make) to place it in table to act as our id,(login\_token is a proper name)
5. Now we do table part(create for first time and update for more than one)
6. Sms otp to user
7. We need a form for user to place otp in it and we check theotp correction by the login\_token column
8. And after that we do login.



3

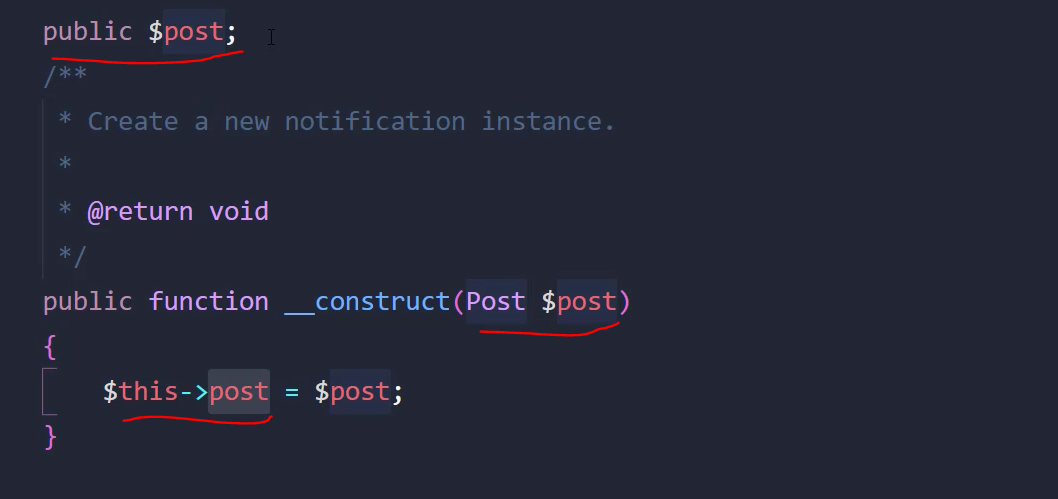
2

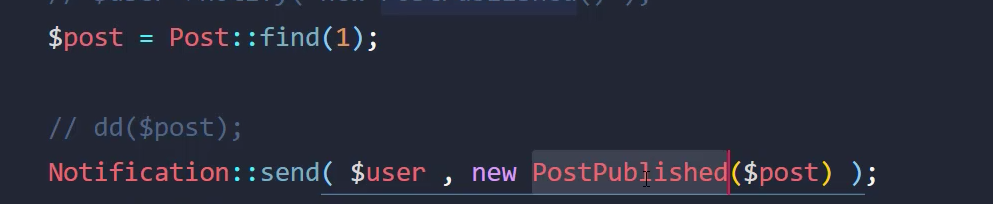
3

1

# Notification:

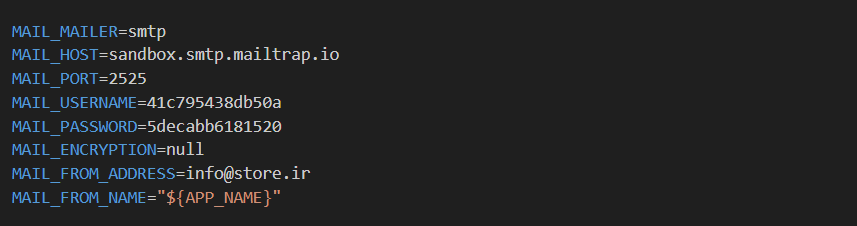
sending parameter? in our notification, is via notification constructor:





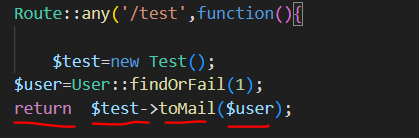
## Via Email:

First we need to cofiger mail class of laravel because notificaton use that.



Second: we make a notification via artisan.

How to use preview laravel mail when using notification?



Notification setup:

Tip-After thinking I rather to use way 1 for sending email,

customize words? of MailMassage of laravel run code bellow:

php artisan vendor:publish --tag=laravel-notifications

how to customize css?

Run code bellow:it wil make resources/views/vendor/mail and in it find default.css and palce your style.

php artisan vendor:publish --tag=laravel-mail

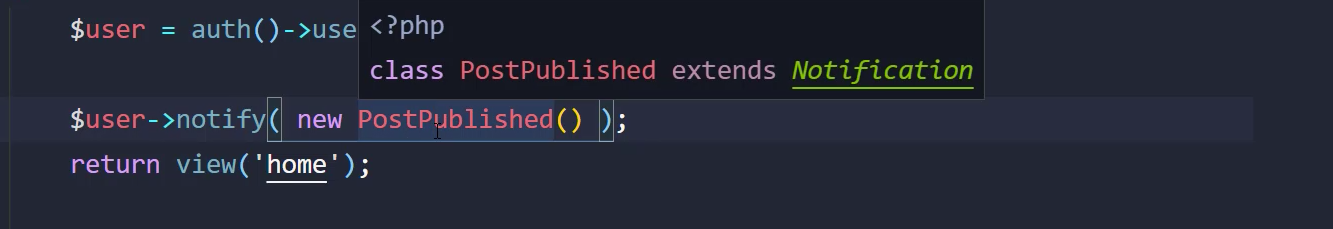
Tip-$notifable in bellow is an object of user model so we can use its index.





How we send a notification?

User model has a trait in it as notifable because of that we can do this:



We are in a controller

Or we can do it via notificaton façade:



Tip-if we have more than one user to send email just set a collection of users in parameter one.

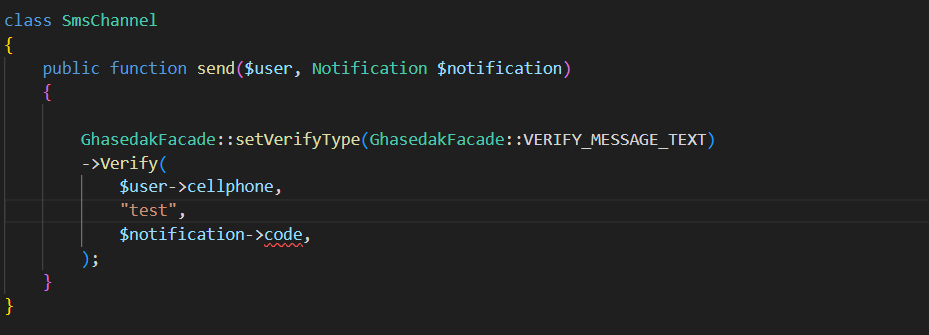
## Via sms:

1-we need a sms channel so we create a channels directory in app directory:

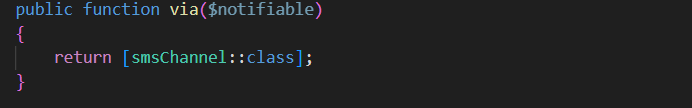
In sms channel we need a

1)send method with tow parameter first a user and second our built notification

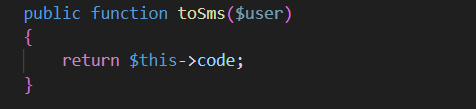
2)a web serviec to send the massage (qasedack has a good doucmentaion)



2-create a notification for it. And do bellow setups:



OtpSms-notification



OtpSms-notification

# Localization:

There is simples in assets.

We should change language in config/app.php

## Verify email and reset password:

The statements of this notification are in the pass bellow and as they are in lang function we can localize it easly

Vender/laravel/framefork/src/auth/notifications.

1-chcek the above path for statements

2-place and translate statements in fa.json in rescoures/lang

There is simples in assets.

# Documentation and other

## Hashing:

Just go for documentation I’m not in mood.

## Session

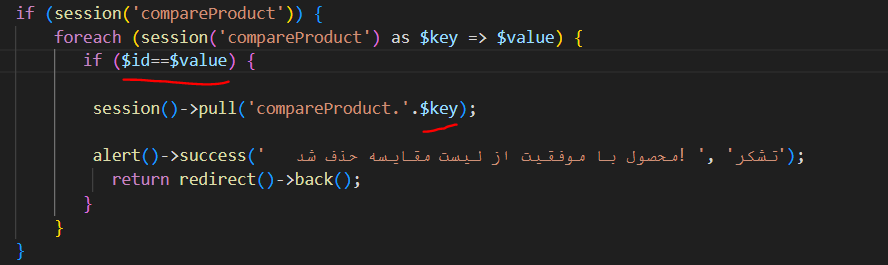
How to add to an existing array session?



Caution for the first time we should:



Delete from a array?



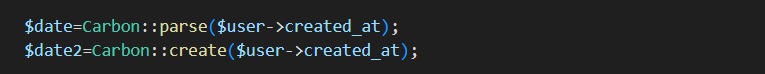
# Date

## Carbon

-Carbon::now();

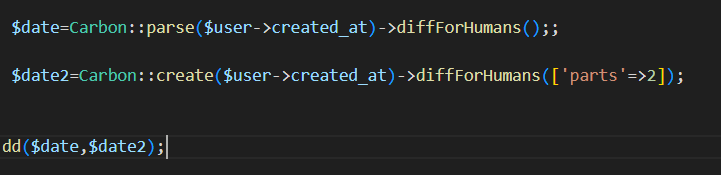
Working on a times that stored in database:

First we should bulid an instance of that time string on Carbon:



Result of both are same

Now we can get some info by Carbon methods:





Note-as you can see carbon some how (base on local of project) echo in persion.

## Verta

Note-verta is customized for shamsi.

Note2-it have a global helper and even a façade.

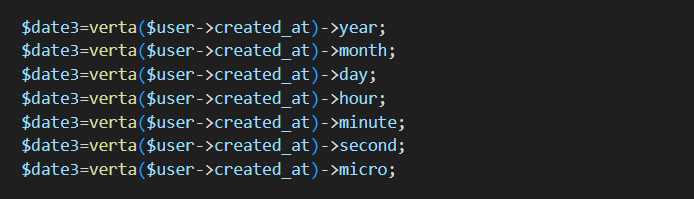
 $date3=verta($user->created\_at);

### Sahmsi to miladi:

 $date2=verta($date3)->toCarbon();

### Getters

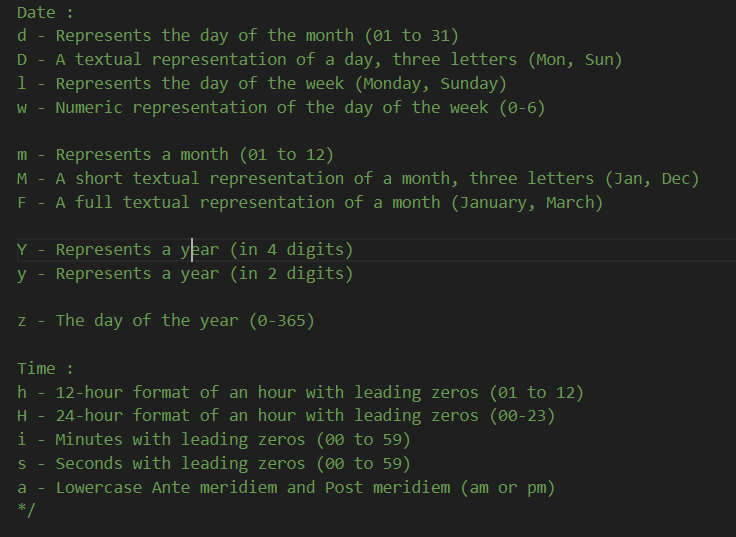
There is more than bellow getters but these are more important.

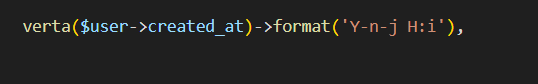


Tip:we have setter to change the date elements like year and hour… ,if I need I can check docs.

### Format

Most usefull:





Others:

Modificatoin,boundries,compersion,diffrence

Note for up situations first check github then click on view More.

<https://github.com/hekmatinasser/verta>