**Video 4**

Create repository in github

Remove vendor from gitnore

git-scm.com

**Video 5**

Php artisan make:model Model\Product –a

-a is to create factory

Factory for dumy data and

Model review and products

php artisan make:model Reviews -m for model with database

use php artisan route:list check on Route::Resource and

Route::apiResource Edit and create are deleted from route list

php artisan make:controller ProductController --resource

Route prefix -It is used to set a common **prefix** for an entire controller

Add everthing to git hub but gitscm should must installed in our system

In terminal >>cd git init

>>git add README.md

>>git commit –m “message”

>>git remote add origin <https://github.com/bitfumes/api.git> ///this above code is given from github repository

>>git push –u origin master

>>cd Git status

>>cd git add .

>>cd git status

>>cd git commit –m “Created Model -a” ///this is message between double quotes ///for git commit-m “message”

>>cd git status

>>cd git push

$table->integer('product\_id')->unsigned()->index();

$table->foreign('product\_id')->references('id')->on('products')->onDelete('cascade');///this is done as froeign key if product deleted then review will automatically deleted

After migration goto github >>cd git add .

>>git commit –m “Migration created for product and review”

Used DB folder and stored sql file init and pushed to github for users to see like us not important

**Video 7**

**php artisan db:seed** after factory and database seeding

without using this db:seed it will show null value in cmd

**error:** made product for column product\_id

and don’t forget to write code: use App\Model\Product; use App\Model\Review;

use in terminal

**ecomapi>>php artisan tinker**

**>>App\Model\Product**

**Error: PHP Fatal error: Undefined constant app/Model/Product in Psy**

**Shell code on line 1**

For writing this above code

**>>App\Model\Product::find(3)**

**>>App\Model\Product::find(5)**

**>>App\Model\Product::find(2)**

**>>App\Model\Product::find(8)->review** for both review and product model items together

**>>App\Model\Review::find(8)-> product**

**Video 8:creating relationship** in model

In revivew.php

public function product()

{

return $this->belongsTo(Product::class);

}

In Product.php

public function review()

{

return $this->hasMany(Review::class);

}

**Video 9**

php artisan help make:resource for making resource

php artisan make:resource Product/ProductCollection works in 5.5 and above only

for resource collection creating laravel Eloquent API Resources to Convert Models into JSON

php artisan make:model Model/Project

**error:** use App\Model\Product made error

error: could not find driver (SQL: select \* from `products`)

(SQL: select \* from information\_schema.tables where table\_schema = ecomapi and table\_name = migrations)

**Error:** Parse error: syntax error, unexpected '?', expecting variable (T\_VARIABLE)in Translator.php (line 79)

Sol:php 7.0.1 but its needed higher version

**ProductController.php**

 public function index(){

        return Product::all();

 }

**ProductCollection.php**

public function toArray($request)    {

        return parent::toArray($request);

   }

ProductController.php and ProductCollection.php

**ProductResource.php** don’t show id created\_at and update\_at to url: <http://localhost:8000/api/products/4>

For reviews php artisan route:list | GET|HEAD | api/products/{product}/reviews **| reviews.index** | App\Http\Controllers\ReviewController@index | api

Check for discount

**ProductController.php**

public function show(Product $product){

        return new ProductResource($product);

  }

Error:property [name] does not exist on this collection instance

Solution:  return new ProductCollection(Product::all());//it will transform only one product        return ProductCollection::collection(Product::all());

**Video 11**

Php artisan make:Resource ReviewResource

Error:Class App\Http\Controllers\Product does not exist

**Video 12(Get Review)**

ReviewController.php

public function index(Product $product)

    {

         //return $product;//this addded to show review for particular product

        //return Review::all();// working shows all product review but needed only particular product

      //return $product->review; //goto Product model relationships prints same structure of database so transform needed

                                    //shows all review for selected product

   return ReviewResource::collection($product->review);//collection of review not for single review shows all review

    }

Php artisan make:Resource ReviewResource.php

Error:Call to a member function first() on null solved due review to reviews in product.php model

**Video 13**

composer require laravel/passport

create passport then php artisan migrate after composer install Passport service provider registers its own database migration directory with the framework

php artisan passport:install

After passport set up and postman url collection setting make auth and serve

**In Postman url:**POST localhost:8000/oauth/token

**Body->form-data direct and below is for raw**

{

"grant\_type" : "password",

"client\_id" : "2",

"client\_secret" : "BfANIMk3rNLiM5q9AUWtjVBYbsDH26LXFUw0LNXP"(copied from passport key generate)

"username" : "abhi144k@gmail.com" //from auth

"password" : "123456"

}

Just way to get Oauth token

Get details through access token

Route::middleware('auth:api')->get('/user', function (Request $request) {

    return $request->user(); });

In Postman GETlocalhost:8000/api/user

Header->

Accept = application/json

Content-Type=

Authorization=Bearer \_\_\_\_\_access token\_\_\_\_\_//uncheck this message will come unauthenticated

Avoid repeat everytime entering create Environment for access token that everytime no need to write again and again

Environment create setting

Environment name:EAPI

Variable/Key=**auth**

Initial Value= Bearer \_\_\_\_\_access token\_\_\_\_\_

Now,

Select Environment name:EAPI

In Postman GETlocalhost:8000/api/user

Header->

Accept = application/json

Content-Type=

Authorization={{auth}}

**Video 14(Create Product)**

Creation of product it should be authenticated for POST,DELETE,PUT so middleware to created

ProductController.php

ceate \_constructor

  $this->middleware('auth:api')->except('index','show');

php artisan make:request ProductRequest its for validation

when nothing error is showing in postman then there is no data or code is made for that content

**ProductRequest.php**

 public function authorize()

    {

        //return false;

        return true;

    }

public function rules()

    {

        return [

            'name' => 'required|max:255|unique:products',

            'description' => 'required',

            'price' => 'required|max:10',

            'stock' => 'required|max:6',

            'discount' => 'required|max:2',

        ];

    }

Postman url:GET <http://127.0.0.1:8000/api/products>

------------------Pending-----------------

For response json

Vendor\Symphony\Response.php

Php namespace resolver and php intelliphense ctrl+p

Postman url:POST <http://127.0.0.1:8000/api/products>

Body-form\_data ‘description’ is from html/postman and ‘detail’ is table attributes

**Video 15(Update Product)**

ProductController.php

 public function update(Request $request, Product $product)

    {

        //return $request->all();

        $product->update($request->all());

    }

Product.php

**Video 15(Delete Product)**

ProductController.php

public function destroy(Product $product)

    {

         $product->delete();

         return response(null,Response::HTTP\_NO\_CONTENT);

    }

**Video 17(Handle Exception)**

Handle exception for product which is not found in database

Goto app/Exception/Handler.php

public function render($request, Exception $exception)

    {

        dd($exception);

        return parent::render($request, $exception);

    }

In Postman

Preview

Error:

ModelNotFoundException {#1155

#model: "App\Model\Product"

#ids: []

#message: "No query results for model [App\Model\Product]."

#code: 0

#file: "C:\wamp64\www\ecomapi2\vendor\laravel

Sol:

public function render($request, Exception $exception){

    if($exception instanceof ModelNotFoundException){

            return response('Model Not Found',404);

        }

        dd($exception);

        return parent::render($request, $exception);

    }

Diendump dd() function is used for unknown product request

public function render($request, Exception $exception)

    {

        if($exception instanceof ModelNotFoundException){

            return response('Model Not Found',404);//return model not found

        }

        dd($exception);

       if($request->expectsJson()){

           if($exception instanceof ModelNotFoundException){

               return response()->json([

                   'errors' => 'Product Model not Found'

               ],Response::HTTP\_NOT\_FOUND);

          }

       }

       //Incorrect route

      if($exception instanceof NotFoundHttpException){

            return response()->json([

                'errors' => 'Incorrect Route Request'

            ],Response::HTTP\_NOT\_FOUND);

      }

        return parent::render($request, $exception);

    }

**Video 18(Custom Exception)**

Refer to video for full understanding

app/Exception/Handler.php

public function render($request, Exception $exception)

    {

      /\*  if($exception instanceof ModelNotFoundException){

            return response('Model Not Found',404);//return model not found

        }

 use ExceptionTrait;

 public function render($request, Exception $exception)

    {

        if($request->expectsJson()){

          return $this->apiException($request,$exception);

        }

        return parent::render($request, $exception);

    }

app/Exception /ExceptionTrait.php

trait ExceptionTrait

{

public function apiException($request,$e)

    {

        if($this->isModel($e)){

            return $this->ModelResponse($e);

        }

        //for incorrect route

       if($this->isHttp($e)){

           return $this->HttpResponse($e);

        }

        return parent::render($request, $exception);//no need of this but above

                                                    //two don't works then normal

                                                    //exception will be passed above

    }

    protected function isModel($e)

    {

        return $e instanceof ModelNotFoundException;

    }

    protected function isHttp($e)

    {

        return $e instanceof NotFoundHttpException;

    }

    protected function ModelResponse($e)

    {

        return response()->json([

            'errors' => 'Product Model not Found'

     ],Response::HTTP\_NOT\_FOUND);

    }

    protected function HttpResponse($e)

    {

        return response()->json([

            'errors' => 'Incorrect Route Request'

        ],Response::HTTP\_NOT\_FOUND);

    }

}

**Video 19(Authorize product updates)**

Check for user authentication to add product by it

1.add user\_id to product table and run phpn artisan migrate:refresh

2.ProductFactory.php

'user\_id' => function(){

   return App\User::all()->random();//user\_id we dont have it we created from auth so function

          //we would use default User factory created

          // so special data is required refer video 19 at 03:00 min

        },

3.DatabaseSeeder.php

  factory(App\User::class,5)->create();//create 5 user default User model

4.php artisan db:seed

To get fake data

Error: In FactoryBuilder.php line 248:

Unable to locate factory with name [default] [App\User].

Sol:ModelFactory.php was userfactory was disabled

5.ProductController.php

Public function Update(){

Check for user\_id

}

 public function ProductUserCheck($product)

    {

        if(Auth::id() != $product->user\_id)

        {

            throw new ProductNotBelongsToUser;

        }

    }

ProductNotBelongsToUser to be defined so create

Php artisan make:exception ProductNotBelongsToUser

ProductNotBelongsToUser.php

public function render()

    {

        return ['data' => 'Product Not Belong to Users'];

    }

Now, make php artisan passport:install

Get Token follow

In postman Url:PUT localhost:8000/api/products/16

Url:DELETE localhost:8000/api/products/16

If data belongs to id it will delete n update otherwise it will give "errors": "Product Not Belong to Users"

**Video 20 n 21(create review for product)**

1.php artisan make:request ReviewRequest

   public function authorize()

    {

        //return false;

        return true;

    }

    public function rules()

    {

        return [

            'customer' => 'required',

            'star' => 'required|integer|between:0,5',

            'review' => 'required'

        ];

    }

Postman POST url: <http://127.0.0.1:8000/api/products/24/reviews>

Body form-data

Customer star review

ReviewController.php

public function store(ReviewRequest $request,Product $product)//ReviewRequest added

    {

        //return $product;//just to check for all produt with review return

        $review = new Review($request->all());//Review model

        $product->review()->save($review);//made error reviews() instead of review() in Product model

        return response([

            'data' => new ReviewResource($review)

         ],Response::HTTP\_CREATED);

    }

Error:Exception:MassAssignementException

Sol:add protected $fillable = [

‘star’,’customer’,’review’

]; in Review model

Error:BadMethodCallException

Sol: made error wrote reviews() instead of review() in Product model

Postman PUTurl: http://127.0.0.1:8000/api/products/42/reviews/1

------------------not working-------------------

ReviewController.php

  public function update(Request $request,Product $product, Review $review)

//Request takes new data and old data take Review model refer video 20 3:45

// Product $product is added

{

        return $review;

}

Error:type error:argumnet 2 passed to App\Http\Controllers\ReviewController::update() must be instance of App\Model\Review, string given,

Exception:fatal throwable error

Sol:Product $product added because added 42 and 1 are for same product and review for same product

Delete

 public function destroy(Product $product,Review $review)

    {

        $review->delete();

        return response(null,Response::HTTP\_NO\_CONTENT);

    }

Postman url:DELETE [http://127.0.0.1:8000/api/products/24/reviews/11](http://127.0.0.1:8000/api/products/24/reviews/11ostma)

Header should be same above