Laravel Passport Step By Step Bold Line Should be change

(https://itsolutionstuff.com/post/laravel-5-how-to-create-api-authentication-using-passport-example.html)

**Step 1 : Install Laravel**

In first step, we require to get fresh Laravel application using bellow command, So open your terminal OR command prompt and run bellow command:

***composer create-project --prefer-dist laravel/laravel blog***

**Step 2: Install Package**

In this step we have to laravel/passpor package for passport method so one your terminal and fire bellow command:

***composer require laravel/passport***

After successfully install package, open **config/app.php** file and add service provider.

**config/app.php**

***'providers' => [***

***....***

***Laravel\Passport\PassportServiceProvider::class,***

***],***

....

Read Also: [Laravel 5.2 API using JWT authentication tutorial from scratch example](https://itsolutionstuff.com/post/laravel-52-api-using-jwt-authentication-tutorial-from-scratch-exampleexample.html)

**Step 3: Run Migration and Install**

After Passport service provider registers, we require to run migration command, after run migration command you will get several new tables in database. So, let's run bellow command:

***php artisan migrate***

Next, we need to install passport using command, Using passport:install command, it will create token keys for security. So let's run bellow command:

***php artisan passport:install***

**app/User.php**

<?php

namespace App;

use Laravel\Passport\HasApiTokens;

use Illuminate\Notifications\Notifiable;

use Illuminate\Foundation\Auth\User as Authenticatable;

class User extends Authenticatable

{

use ***HasApiTokens***, Notifiable;

/\*\*

\* The attributes that are mass assignable.

\*

\* @var array

\*/

protected $fillable = [

'name', 'email', 'password',

];

/\*\*

\* The attributes that should be hidden for arrays.

\*

\* @var array

\*/

protected $hidden = [

'password', 'remember\_token',

];

}

**app/Providers/AuthServiceProvider.php**

<?php

namespace App\Providers;

***use Laravel\Passport\Passport;***

use Illuminate\Support\Facades\Gate;

use Illuminate\Foundation\Support\Providers\AuthServiceProvider as ServiceProvider;

class AuthServiceProvider extends ServiceProvider

{

/\*\*

\* The policy mappings for the application.

\*

\* @var array

\*/

protected $policies = [

'App\Model' => 'App\Policies\ModelPolicy',

];

/\*\*

\* Register any authentication / authorization services.

\*

\* @return void

\*/

public function boot()

{

$this->registerPolicies();

***Passport::routes();***

}

}

**config/auth.php**

<?php

return [

.....

'guards' => [

'web' => [

'driver' => 'session',

'provider' => 'users',

],

***'api' => [***

***'driver' => 'passport',***

***'provider' => 'users',***

***],***

],

.....

]

**Step 5: Create API Route**

In this step, we will create api routes. Laravel provide api.php file for write web services route. So, let's add new route on that file.

**routes/api.php**

<?php

/\*

|--------------------------------------------------------------------------

| API Routes

|--------------------------------------------------------------------------

|

| Here is where you can register API routes for your application. These

| routes are loaded by the RouteServiceProvider within a group which

| is assigned the "api" middleware group. Enjoy building your API!

|

\*/

Route::post('login', 'API\UserController@login');

Route::post('register', 'API\UserController@register');

Route::group(['middleware' => 'auth:api'], function(){

Route::post('details', 'API\UserController@details');

});

**Step 6: Create Controller**

In last step we have to create new controller and three api method, So first create new directory "API" on Controllers folder. So let's create UserController and put bellow code:

**app/Http/Controllers/API/UserController.php**

<?php

namespace App\Http\Controllers\API;

use Illuminate\Http\Request;

use App\Http\Controllers\Controller;

use App\User;

use Illuminate\Support\Facades\Auth;

use Validator;

class UserController extends Controller

{

public $successStatus = 200;

/\*\*

\* login api

\*

\* @return \Illuminate\Http\Response

\*/

public function login(){

if(Auth::attempt(['email' => request('email'), 'password' => request('password')])){

$user = Auth::user();

$success['token'] = $user->createToken('MyApp')->accessToken;

return response()->json(['success' => $success], $this->successStatus);

}

else{

return response()->json(['error'=>'Unauthorised'], 401);

}

}

/\*\*

\* Register api

\*

\* @return \Illuminate\Http\Response

\*/

public function register(Request $request)

{

$validator = Validator::make($request->all(), [

'name' => 'required',

'email' => 'required|email',

'password' => 'required',

'c\_password' => 'required|same:password',

]);

if ($validator->fails()) {

return response()->json(['error'=>$validator->errors()], 401);

}

$input = $request->all();

$input['password'] = bcrypt($input['password']);

$user = User::create($input);

$success['token'] = $user->createToken('MyApp')->accessToken;

$success['name'] = $user->name;

return response()->json(['success'=>$success], $this->successStatus);

}

/\*\*

\* details api

\*

\* @return \Illuminate\Http\Response

\*/

public function details()

{

$user = Auth::user();

return response()->json(['success' => $user], $this->successStatus);

}

}

Now we are ready to run our example so run bellow command ro quick run:

php artisan serve

Fetching Data

$.ajax({

url: 'YourRestEndPoint',

headers: {

'Authorization': `Bearer ${token}`,

},

method: 'POST',

data: YourData,

success: function(data){

console.log('succes: '+data);

}

});

// axios

axios({

method: 'post',

url: yourEndpoint,

data: yourData,

headers: { 'authorization': `Bearer ${token}` }

});

