**Make your own auth in laravel (implementing adding multiple authentication)**

First this we have to add the admin guards in the **config/auth.php** file so Laravel will know which model we are trying to use for the admin authentication. Open you **config/auth.php** file and find guards array which will looks like this one.



As you can see we have added the admin and admin-api guard so our application can use two different types of model for authentication purpose. In admin array we are defining session as a authentication driver which will use the **admins** provider and same for the **admin-api** array, basically we just copied the default guards and changed the values for them.

As you can see, we have defined the providers in both **admin** and **admin-api** to **admins**. Now we will add a new provider in **auth.php** config file.

The next array element is the **providers**, which looks like below.



In the above code example, we added a new provider which is **admins** and it uses the eloquent as it’s driver. Next we defined the model which will be used for this provider which is App\Models\Admin::class.

Next, in the passwords array, we will define which table will be used to add the functionality for the admin password reset. We will copy the existing **users** one and replace it to admins.



As you can see, for password we have just added the guard name **admins** and used the same details as **users** one, which means we will use the same password\_reset table for the admins password reset functionality.

Next, we will ad some routes as below;



Let’s go through each routes one by one. Firstly, we are adding a routes group to prefix all our admin routes with /admin. Next within this route group we added the login routes, first one is the GET route **(/admin/login)** which will load the admin login page. Next one is the POST route **(/admin/login)**, where we will send the post request for authentication. Next one is the logout route **(/admin/logout)** which will be again a GET request.

The last one is the dashboard route which will map to /admin URL and will load the dashboard view of admin.

Next, we add out login controller;



In the login method, we are firstly validating the user input for email and password. Then we are using the **Auth::** facade to define our admin guard and then making the attempt to authenticate the admin user.

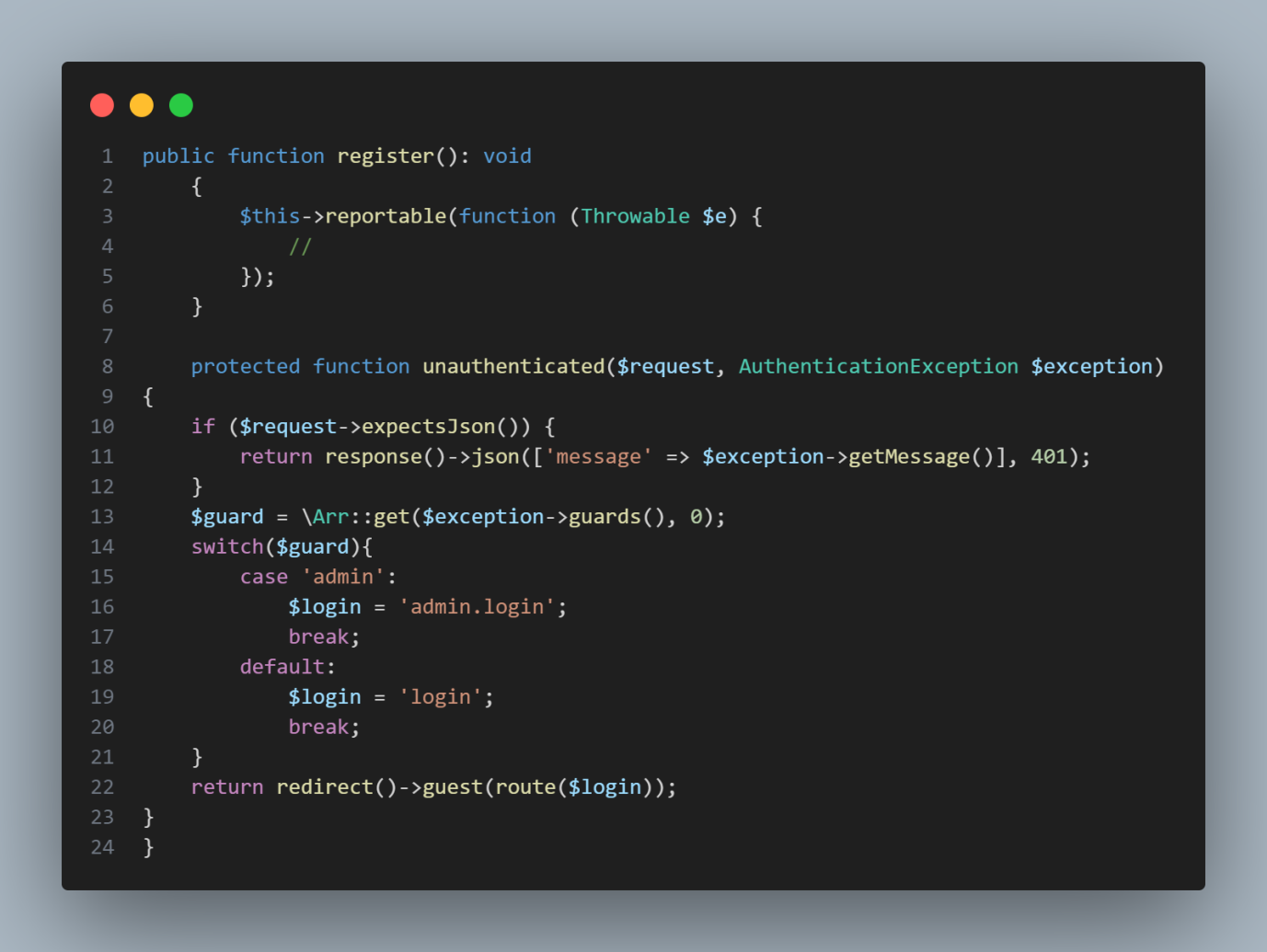
If the authentication is successful we are redirecting the admin users to admin.dashboard route which loads the admin dashboard view. If authentication is not successful then we are returning the user back to the login page.

we want to make sure if an admin is authenticated then redirect them to the dashboard page when they try to load the login route again. As they are already logged in so there is no point to show them the login page.

For this we need to adjust the **RedirectIfAuthenticated** middleware which comes with the Laravel and located at **app/Http/Middleware**. Open this middleware and update the handle() method with the below one.



Next, we want to make a check if the admin is not authenticated then don’t let them load the dashboard view instead redirect them to the login page. For this open **app/Exceptions/Handler.php** file and add the below method at the bottom of the class.



In the above method, we are getting the current guard from the **AuthenticationException** class and running the switch statement on it. Based on the guard value we are redirecting the user to the right login page.

Don’t forget to import, **AuthenticationException** class in the top