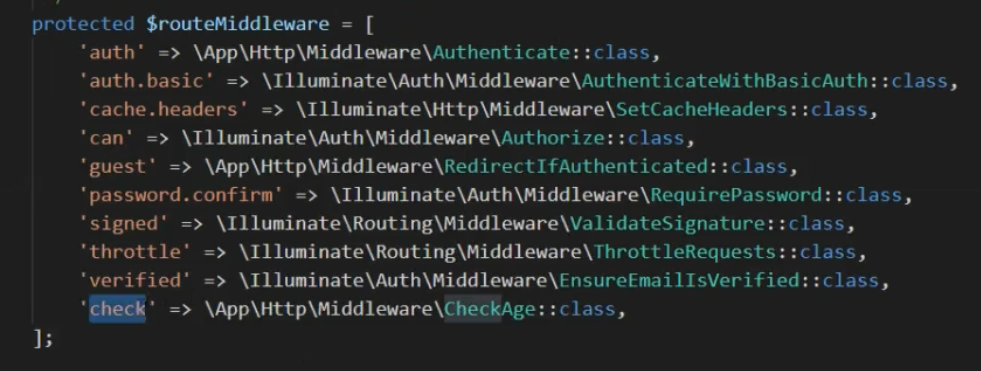
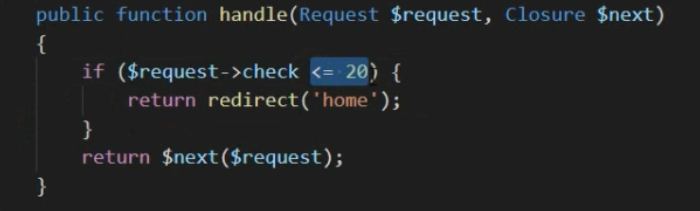
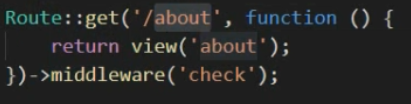
# RouteMiddleware

Whenever we create a middleware, we need to register this in the **kernel.php,** inside the **routeMiddleware** array.

****

****

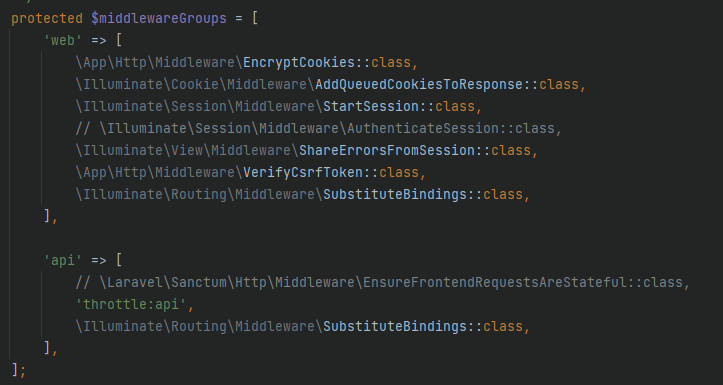
****



Here its key value pair, key is the short name, and value is the actual class name with full path. We use the short name in the rest of the project.

# Middleware group

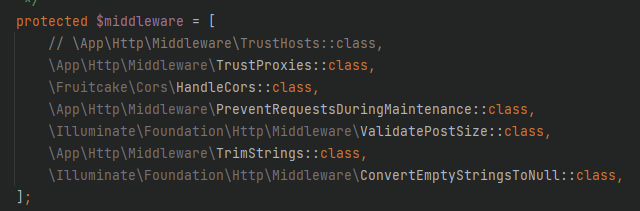
In many cases, we want to use multiple middleware for a route. Following the DRY principle Laravel provides a middleware group array in the **kernel.php**. This is an array of array.



Here, we can see web and api middleware groups. So, question is where is these used? We can go to **app -> Providers -> RouteServiceProvider.php.** Here we can see configuration of rate limiting and route configuration.

# Middleware

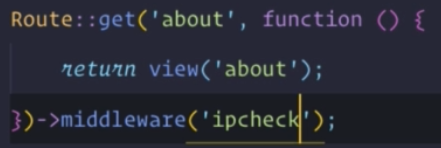
This is an array of middleware that are applied to all the routes automatically.



So, we can see there is multiple ways of applying middleware in Laravel. We can define globally by using Middleware array, we can define a group of middleware or we can define a single middleware.

## Creating our own middleware:

* We can use command **php artisan make:middleware IpCheck** and it will create a new middleware named IpCheck.
* Now, if we want to add this middleware to any route, we can simply use the following syntax.

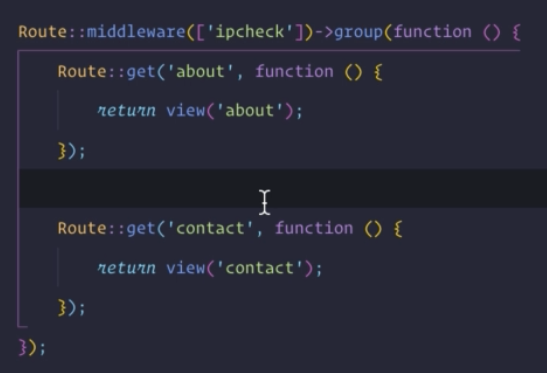


* Finally, we need to add the middleware to the routeMiddleware array in the kernel.php file.



## Applying middleware to group of routes:

Below is an example of how we can apply same middleware to bunch of routes. The functions take a function as parameter, the argument is an array of middleware.



## Applying middleware to a controller:

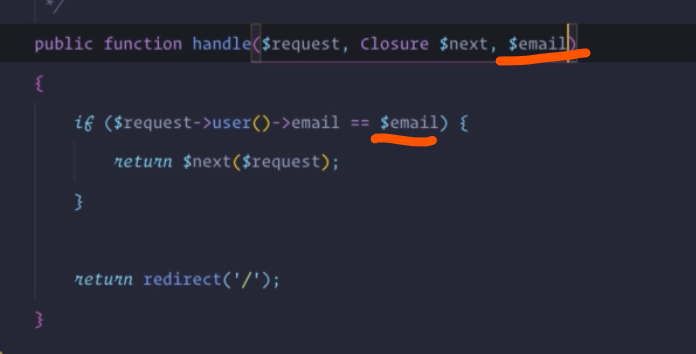
To apply a middleware into a controller, we can go to its constructor and define the middleware. The parameter is an array of middleware so we can add multiple middleware as well.



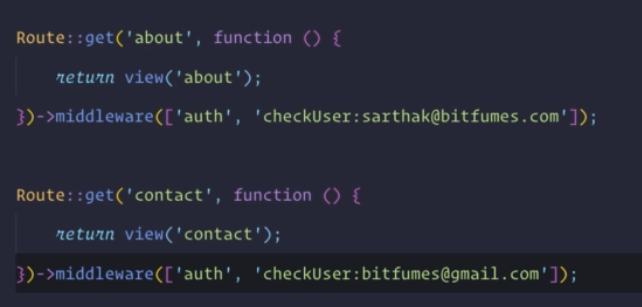
* **To remove a function of controller from the middleware stack:** We can define them in the except function in the constructor.



## Middleware with parameters:



In cased like above, we need to pass parameters into middleware. To achieve this, we can use the following example. Here, both routes use the auth and checkuser middleware. But, checkuser has different arguments

****

# Middleware priority:

In the kernel.php (by default it may not exists) we can define the order of the middleware as we want.

