Authentication

THAT PASSWORD ISN'T VERY STRONG.
I'LL HAVE YOU KNOW IT'S ACTUALLY THE BEST.
I FOUND IT IN THIS ARTICLE, "MOST POPULAR PASSWORDS OF 2015"
BUTTFEED

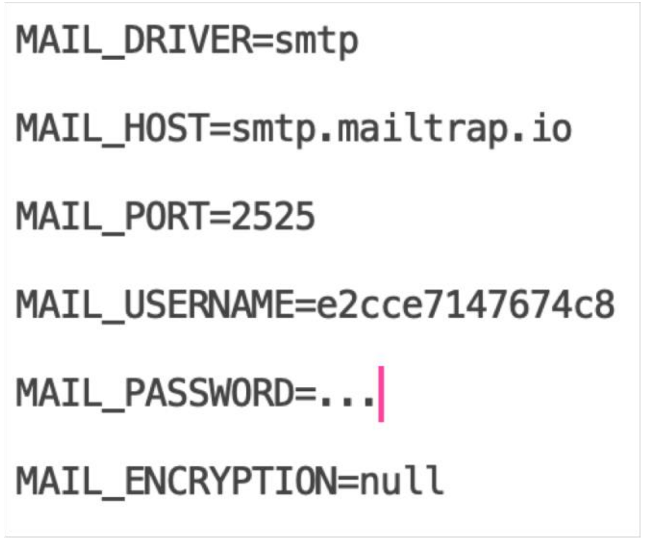
Aside – Configuring Email

When we start working with user accounts there may be the need to send out emails!

Configuring Email

If you want Laravel to send out email, then you will need to configure it.

Generally, the config goes into your .env file.



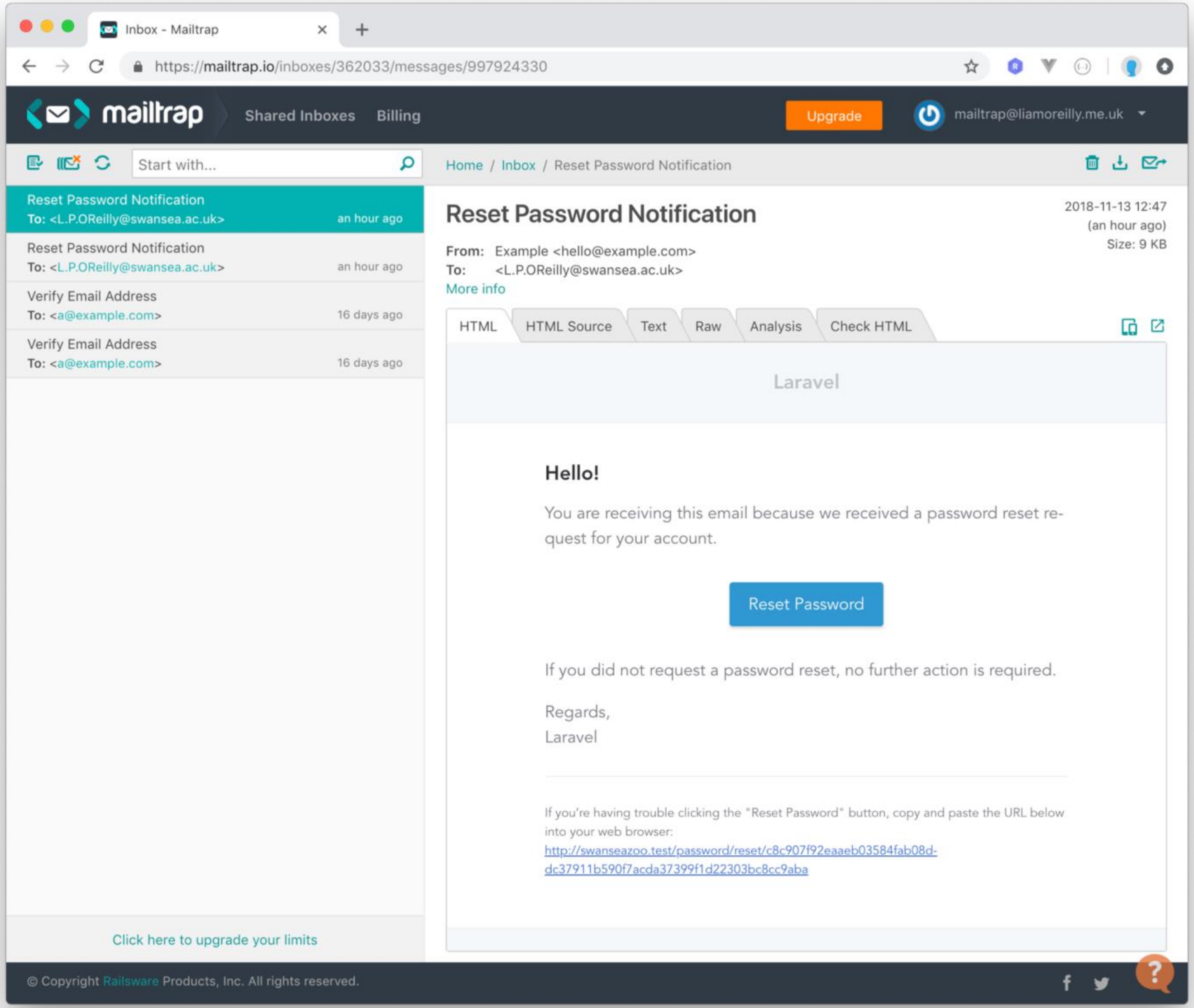
```
MAIL_DRIVER=smtp

MAIL_HOST=smtp.mailtrap.io

MAIL_PORT=2525

MAIL_USERNAME=e2cce7147674c8

MAIL_PASSWORD=...

MAIL_ENCRYPTION=null
```

But you might also look in config/email.php

Mailtrap

mailtrap.io

- Webservice that looks like SMTP server.
- You configure your server to send email through it.
- But it doesn't actually send mail, it traps it and allows you to view it.
- Great for testing – you can't acidentially spam people.



Inbox - Mailtrap
https://mailtrap.io/inboxes/362033/messages/997924330
mailtrap
Shared Inboxes
Billing
Upgrade
mailtrap@liamoreilly.me.uk
Start with...
Home / Inbox / Reset Password Notification
HTML
HTML Source
Text
Raw
Analysis
Check HTML
Reset Password
Click here to upgrade your limits

Reset Password Notification
To: <L.P.OReilly@swansea.ac.uk> an hour ago

Reset Password Notification
To: <L.P.OReilly@swansea.ac.uk> an hour ago

Verify Email Address
To: <a@example.com> 16 days ago

Verify Email Address
To: <a@example.com> 16 days ago

Reset Password Notification

From: Example <hello@example.com>
To: <L.P.OReilly@swansea.ac.uk>
More info

2018-11-13 12:47
(an hour ago)
Size: 9 KB

Laravel

Hello!

You are receiving this email because we received a password reset re-quest for your account.

If you did not request a password reset, no further action is required.

Regards,
Laravel

If you're having trouble clicking the "Reset Password" button, copy and paste the URL below into your web browser:
http://swanseazoo.test/password/reset/c8c907f92eaaeb03584fab08d-dc37911b590f7acda37399f1d22303bc8cc9aba

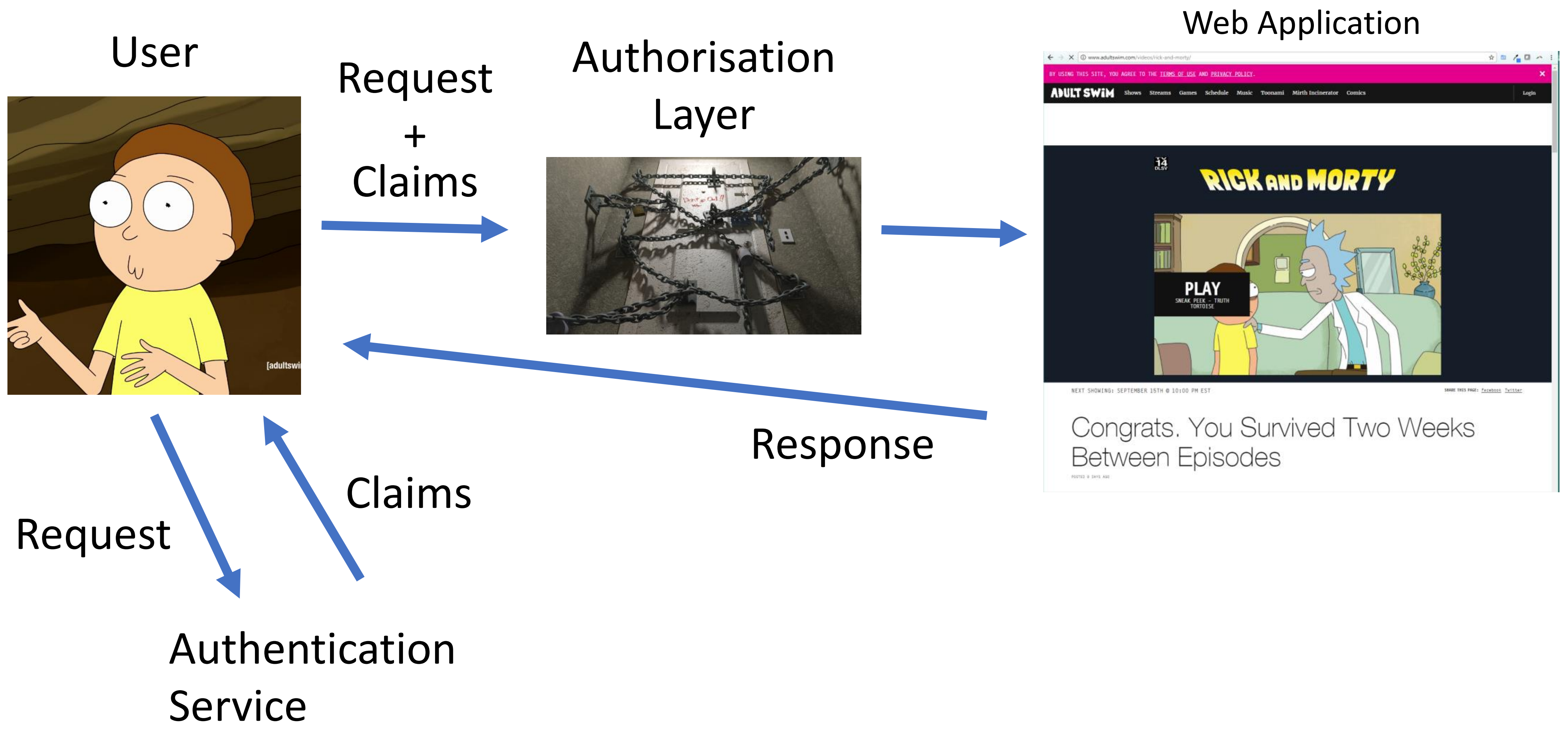
© Copyright Railsware Products, Inc. All rights reserved.

Securing applications using user identity

- A simple first step in making a web application more secure is requiring a user to log in
- In the vast majority of cases you will want different users to have different levels of access
- Advanced applications might have a radically different user experience depending on user role

Authentication and Authorisation

- These are two different concepts
- Authentication is concerned with verifying users are who they say they are
- Authorisation is concerned with verifying what a user is able to do in your application


User
Request
+
Claims
Authorisation
Layer
Web Application
Response
Request
Claims
Authentication
Service
Congrats. You Survived Two Weeks Between Episodes

Laravel Provides Authentication Out of the Box

- Laravel comes with great authentication out of the box.
  - Requires almost no work.
  - It will scaffold out an authentication system that you can customise as much as you like.
  - It creates the:
    - Routes, controllers and views
  - It provides ability to login, logout, register, reset password
    - Can also verify email address,

- You get all full authentication system in seconds.

- This uses middleware, which we will discuss in the next session.

Artisan Command

```
artisan help make:auth
```

```
Description:
  Scaffold basic login and registration views and routes



Usage:
  make:auth [options]



Options:
      --views              Only scaffold the authentication views
      --force              Overwrite existing views by default
  -h, --help               Display this help message
  -q, --quiet              Do not output any message
  -V, --version            Display this application version
      --ansi               Force ANSI output
      --no-ansi            Disable ANSI output
  -n, --no-interaction     Do not ask any interactive question
...
```

Running artisan make:auth

- We scaffold the authentication:

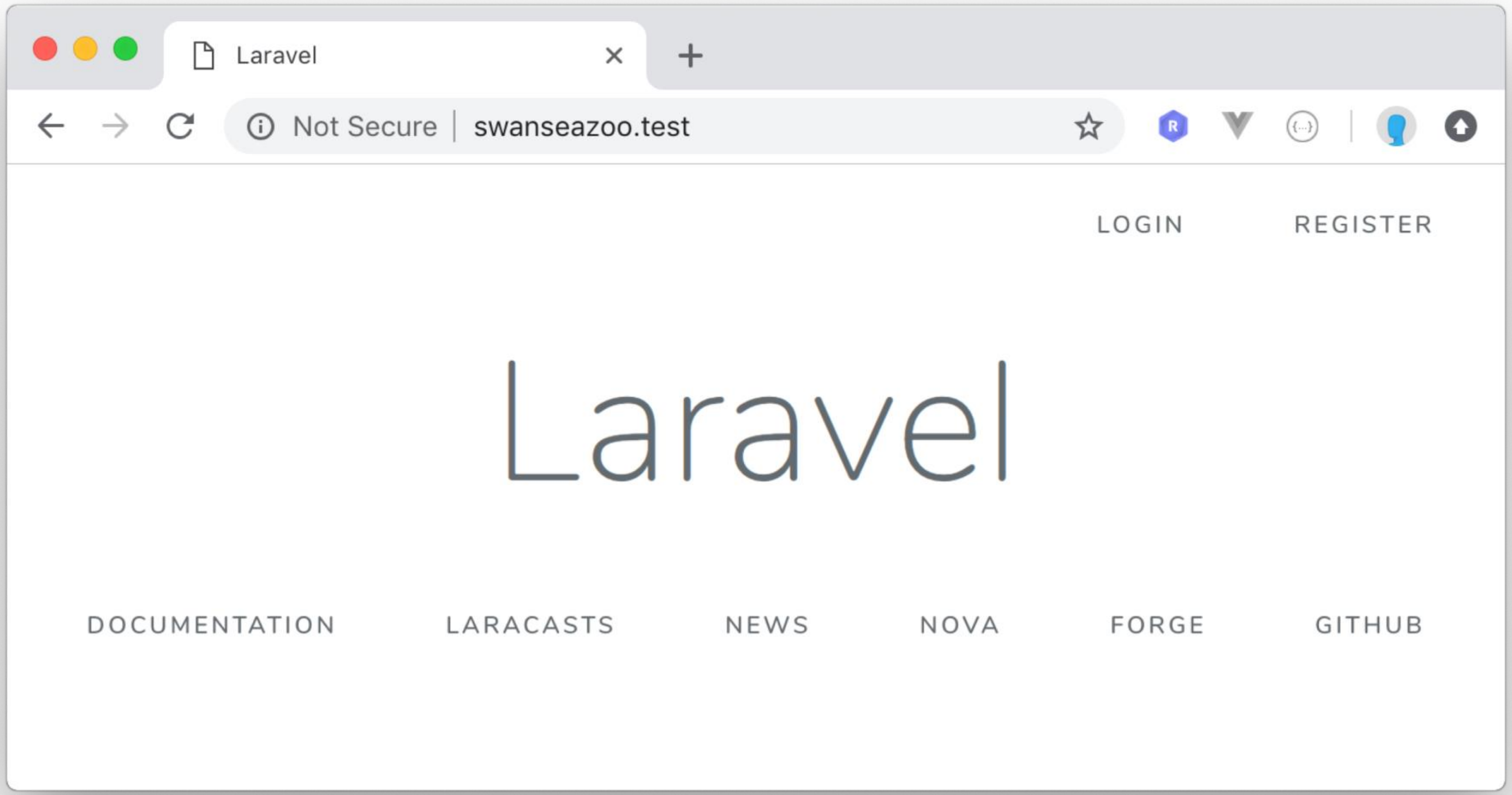
```
artisan make:auth
```

```
Authentication scaffolding generated successfully.
```

- This will create many files. If it is going to replace a file it will ask you.
- For swanseazoo it will try to replace the layout file
  Resources/views/layouts/app.blade.php
  So we will just move our one out of the way and merge them later.
- The produced controllers assume that you use the provided user migration file.

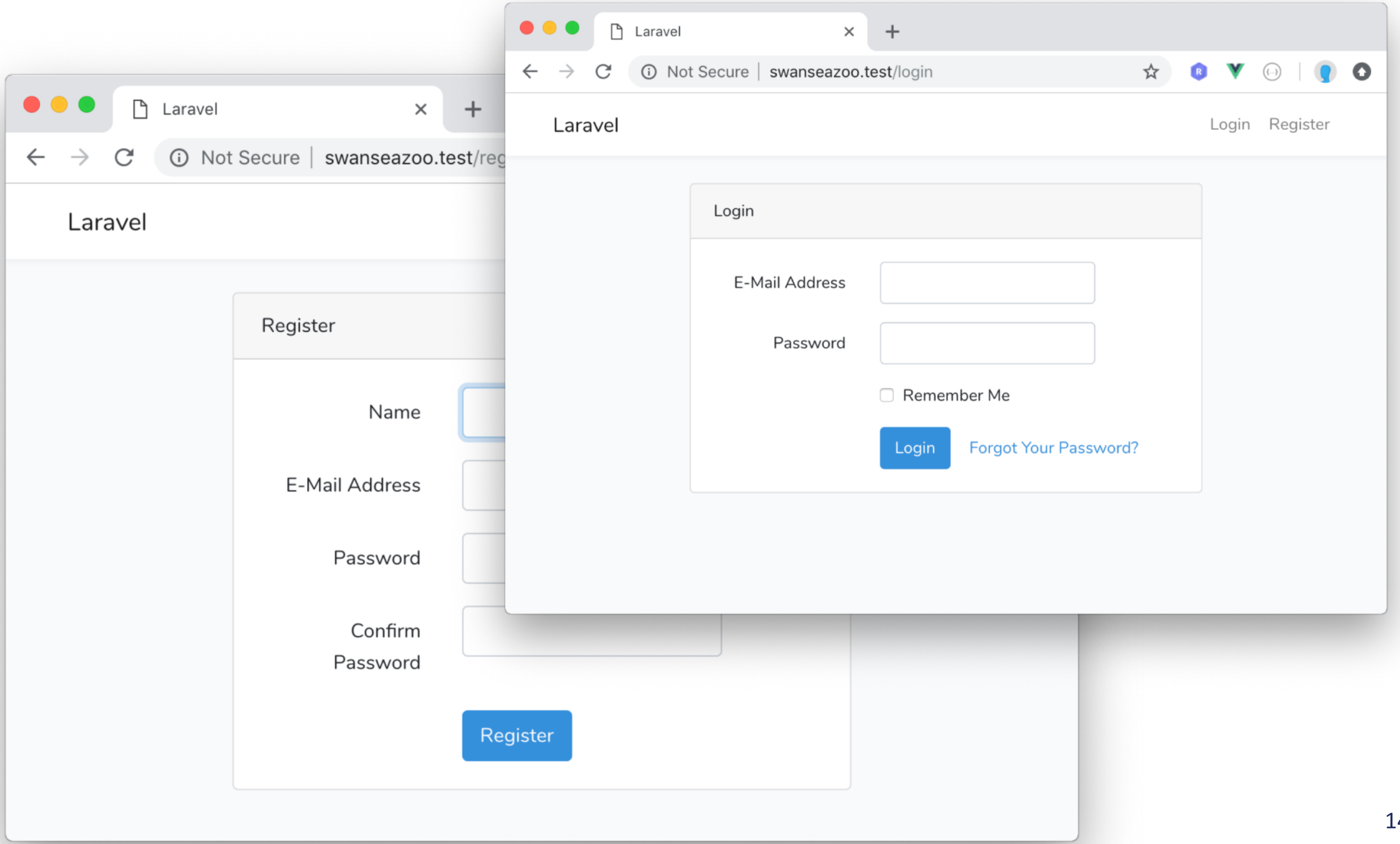
What Do We Get

We get a Navbar (in Bootstrap) with login and register buttons



What Do We Get (Cont.)

We get fully working login and register pages.



Laravel

Not Secure | swanseazoo.test/reg

Laravel

Register

Name

E-Mail Address

Password

Confirm Password

Register

Laravel

Not Secure | swanseazoo.test/login

Laravel

Login Register

Login

E-Mail Address

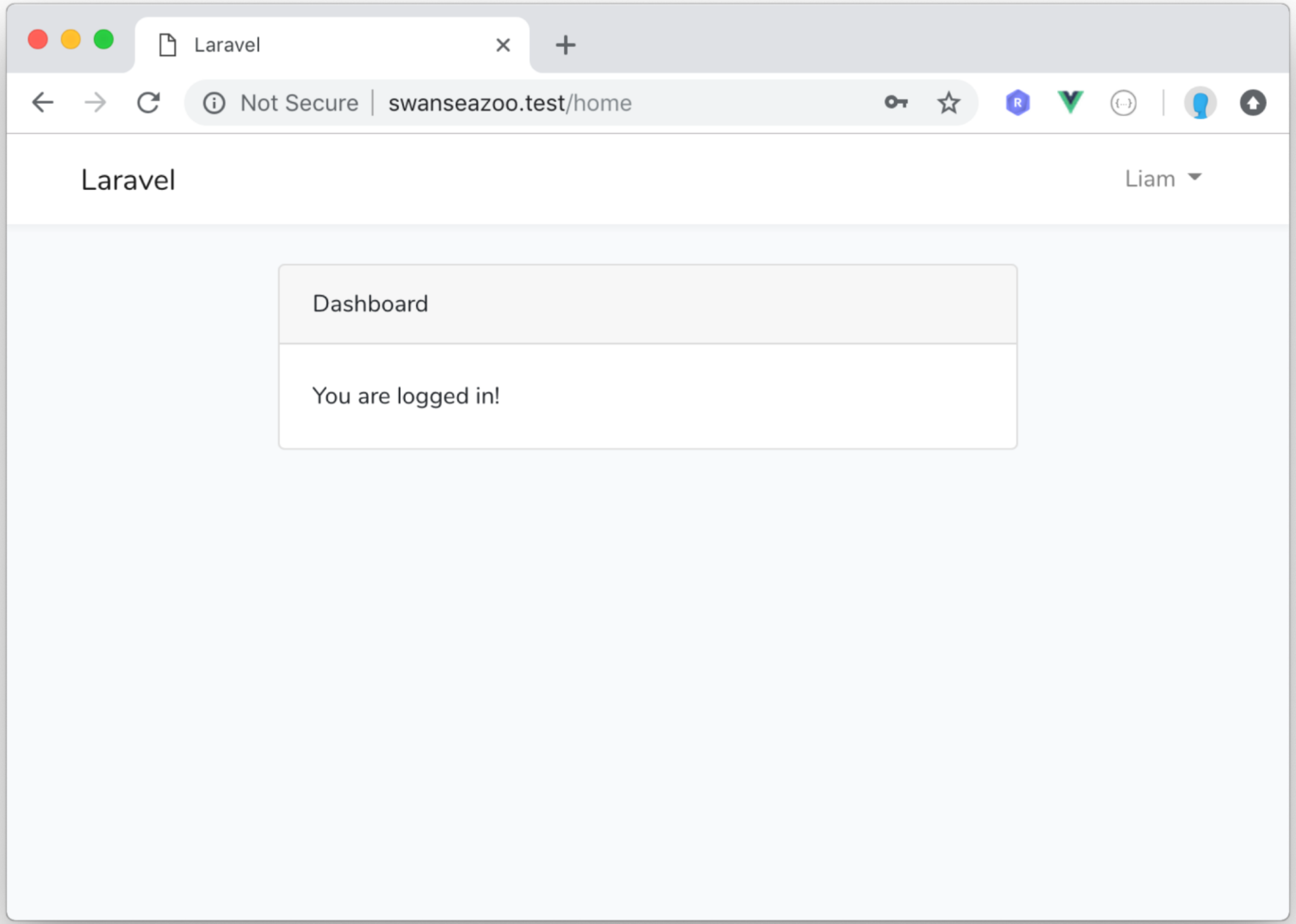
Password

Remember Me

Login Forgot Your Password?

What Do We Get (Cont.)

A homepage (dashboard) which you can only access if logged in.



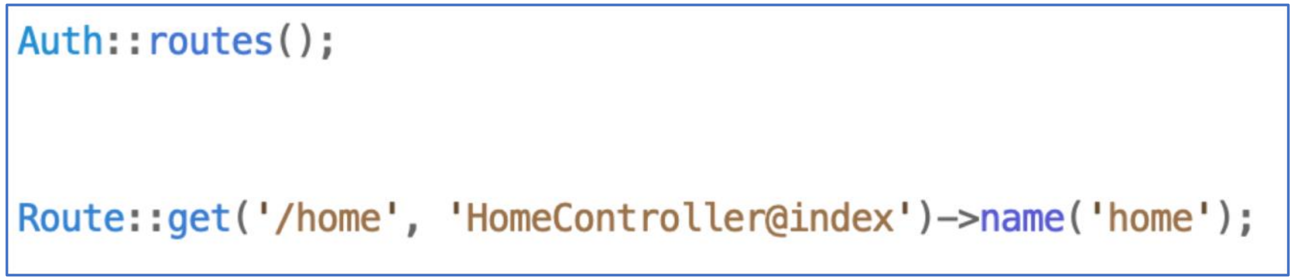
Dashboard

You are logged in!

Authentication: Routes

- Wow that was easy. But how does it work.
  - Let’s find out.

- First Let’s look at the added routes (in routes/web.php)



```
Auth::routes();


Route::get('/home', 'HomeController@index')->name('home');
```

  - 2 lines have been added.
    - The first adds all the routes to do with authentication (see next slides).
    - The second adds a ‘home’ page which you are redirected to once logged in.

Routes: Login, Logout

```
artisan route:list
```

Display the registered routes

```
+----------+------------------------+------------------+-----------------------------------------------------------+--------------+
| Method   | URI                    | Name             | Action                                                    | Middleware   |
+----------+------------------------+------------------+-----------------------------------------------------------+--------------+
| ...                                                  App\Http\Controllers                                                      |
| GET|HEAD | login                  | login            | ...\Auth\LoginController@showLoginForm                    | web,guest    |
| POST     | login                  |                  | ...\Auth\LoginController@login                            | web,guest    |
| POST     | logout                 | logout           | ...\Auth\LoginController@logout                           | web          |
| POST     | password/email         | password.email   | ...\Auth\ForgotPasswordController@sendResetLinkEmail      | web,guest    |
| GET|HEAD | password/reset         | password.request | ...\Auth\ForgotPasswordController@showLinkRequestForm     | web,guest    |
| POST     | password/reset         | password.update  | ...\Auth\ResetPasswordController@reset                    | web,guest    |
| GET|HEAD | password/reset/{token} | password.reset   | ...\Auth\ResetPasswordController@showResetForm            | web,guest    |
| GET|HEAD | register               | register         | ...\Auth\RegisterController@showRegistrationForm          | web,guest    |
| POST     | register               |                  | ...\Auth\RegisterController@register                      | web,guest    |
+----------+------------------------+------------------+-----------------------------------------------------------+--------------+
```

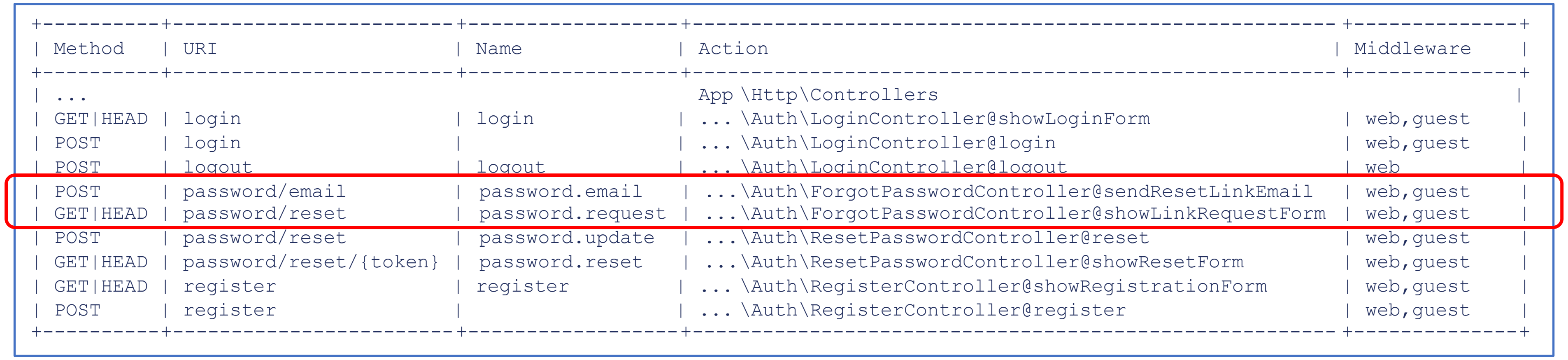
- That's a lot of routes
- Lets look at each in turn:
  - GET login – show the login form.
  - POST login – login form posts to here to actually process the login request.
  - POST logout – logout of site. The logout button in navbar will post to here.

Routes: Registering

```
+----------+------------------------+------------------+-------------------------------------------------------+--------------+
| Method   | URI                    | Name             | Action                                                | Middleware   |
+----------+------------------------+------------------+-------------------------------------------------------+--------------+
| ...                                                  App\Http\Controllers                                                   |
| GET|HEAD | login                  | login            | ...\Auth\LoginController@showLoginForm                | web,guest    |
| POST     | login                  |                  | ...\Auth\LoginController@login                        | web,guest    |
| POST     | logout                 | logout           | ...\Auth\LoginController@logout                       | web          |
| POST     | password/email         | password.email   | ...\Auth\ForgotPasswordController@sendResetLinkEmail  | web,guest    |
| GET|HEAD | password/reset         | password.request | ...\Auth\ForgotPasswordController@showLinkRequestForm | web,guest    |
| POST     | password/reset         | password.update  | ...\Auth\ResetPasswordController@reset                | web,guest    |
| GET|HEAD | password/reset/{token} | password.reset   | ...\Auth\ResetPasswordController@showResetForm        | web,guest    |
| GET|HEAD | register               | register         | ...\Auth\RegisterController@showRegistrationForm      | web,guest    |
| POST     | register               |                  | ...\Auth\RegisterController@register                  | web,guest    |
+----------+------------------------+------------------+-------------------------------------------------------+--------------+
```

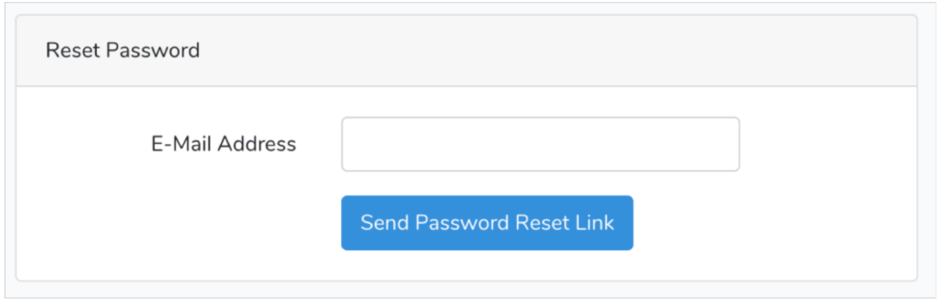
- GET register – display the register form.
- POST register – the register form posts to here to actual process the register request.

Routes: Forgot Password – Send Email



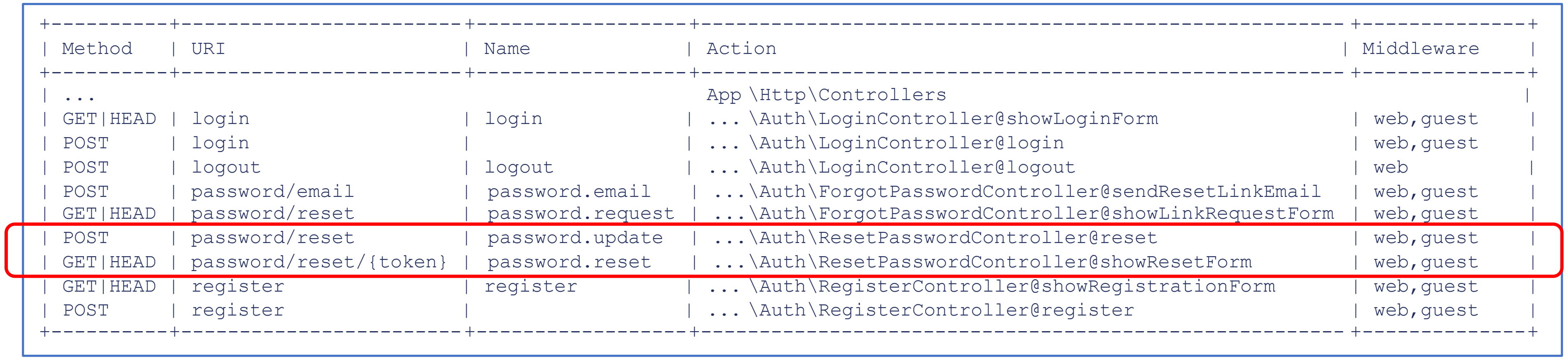
```
+----------+-------------------------+-----------------+------------------------------------------------------------+--------------+
| Method   | URI                     | Name            | Action                                                     | Middleware   |
+----------+-------------------------+-----------------+------------------------------------------------------------+--------------+
| ...                                                   App \Http\Controllers                                                     |
| GET|HEAD | login                   | login           | ... \Auth\LoginController@showLoginForm                    | web,guest    |
| POST     | login                   |                 | ... \Auth\LoginController@login                            | web,guest    |
| POST     | logout                  | logout          | ... \Auth\LoginController@logout                           | web          |
| POST     | password/email          | password.email   | ...\Auth\ForgotPasswordController@sendResetLinkEmail     | web,guest    |
| GET|HEAD | password/reset          | password.request | ...\Auth\ForgotPasswordController@showLinkRequestForm    | web,guest    |
| POST     | password/reset          | password.update  | ...\Auth\ResetPasswordController@reset                   | web,guest    |
| GET|HEAD | password/reset/{token}  | password.reset   | ...\Auth\ResetPasswordController@showResetForm           | web,guest    |
| GET|HEAD | register                | register        | ... \Auth\RegisterController@showRegistrationForm          | web,guest    |
| POST     | register                |                 | ... \Auth\RegisterController@register                      | web,guest    |
+----------+-------------------------+-----------------+------------------------------------------------------------+--------------+
```

- GET password/reset – display the reset (I forgot my password) form.



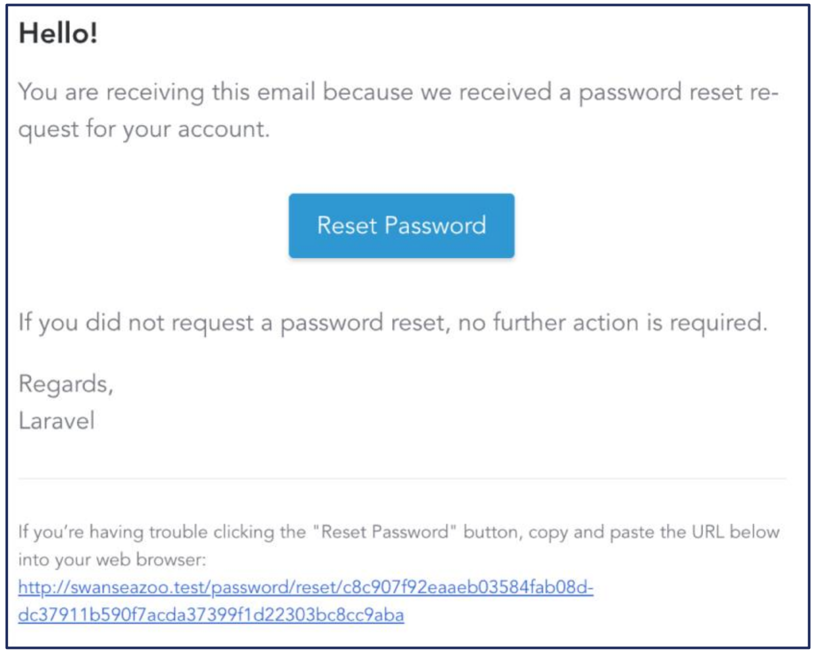
- POST password/email - reset form posts to here. This causes an email with a reset link to password/reset/{token} to be sent.

Routes: Forgot Password: Reset Password



```
+----------+------------------------+-----------------+------------------------------------------------------------+--------------+
| Method   | URI                    | Name            | Action                                                     | Middleware   |
+----------+------------------------+-----------------+------------------------------------------------------------+--------------+
| ...                                                 App \Http\Controllers                                                       |
| GET|HEAD | login                  | login           | ... \Auth\LoginController@showLoginForm                     | web,guest    |
| POST     | login                  |                 | ... \Auth\LoginController@login                             | web,guest    |
| POST     | logout                 | logout          | ... \Auth\LoginController@logout                            | web          |
| POST     | password/email         | password.email   | ...\Auth\ForgotPasswordController@sendResetLinkEmail        | web,guest    |
| GET|HEAD | password/reset         | password.request | ...\Auth\ForgotPasswordController@showLinkRequestForm       | web,guest    |
| POST     | password/reset         | password.update  | ...\Auth\ResetPasswordController@reset                      | web,guest    |
| GET|HEAD | password/reset/{token} | password.reset   | ...\Auth\ResetPasswordController@showResetForm              | web,guest    |
| GET|HEAD | register               | register        | ... \Auth\RegisterController@showRegistrationForm           | web,guest    |
| POST     | register               |                 | ... \Auth\RegisterController@register                       | web,guest    |
+----------+------------------------+-----------------+------------------------------------------------------------+--------------+
```

! GET password/reset{token} - This is the link in the email.



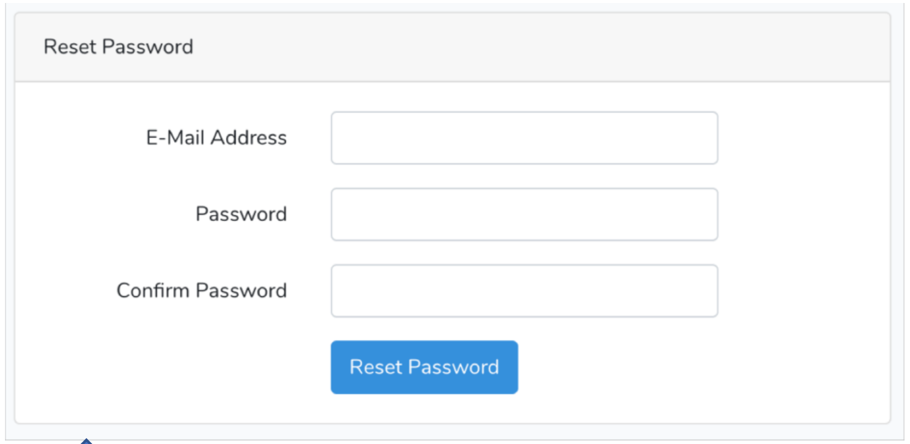
Hello!

You are receiving this email because we received a password reset request for your account.

If you did not request a password reset, no further action is required.

Regards,
Laravel

If you're having trouble clicking the "Reset Password" button, copy and paste the URL below into your web browser: http://swanseazoo.test/password/reset/c8c907f92eaaeb03584fab08d-dc37911b590f7acda37399f1d22303bc8cc9aba



Reset Password

E-Mail Address	
Password	
Confirm Password	

The token will be stored in the database and is unique for the user

Routes: Forgot Password: Reset Password (Cont.)

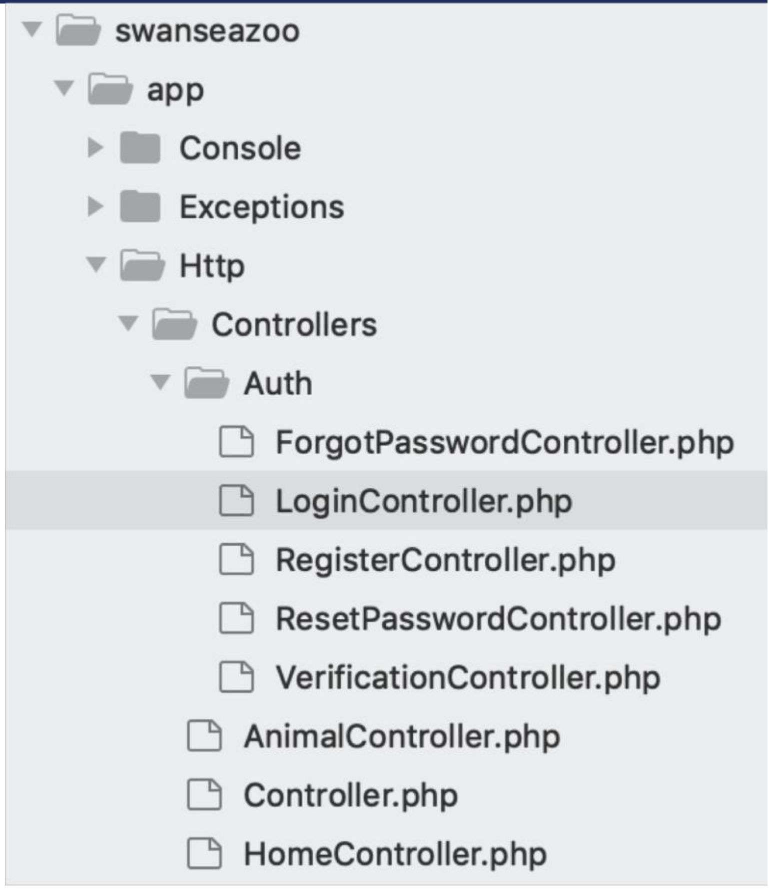
```
+----------+-------------------------+-----------------+----------------------------------------------------------+-------------+
| Method   | URI                     | Name            | Action                                                   | Middleware  |
+----------+-------------------------+-----------------+----------------------------------------------------------+-------------+
| ...                                                     App\Http\Controllers                                                   |
| GET|HEAD | login                   | login           | ...\Auth\LoginController@showLoginForm                   | web,guest   |
| POST     | login                   |                 | ...\Auth\LoginController@login                           | web,guest   |
| POST     | logout                  | logout          | ...\Auth\LoginController@logout                          | web         |
| POST     | password/email          | password.email  | ...\Auth\ForgotPasswordController@sendResetLinkEmail     | web,guest   |
| GET|HEAD | password/reset          | password.request | ...\Auth\ForgotPasswordController@showLinkRequestForm   | web,guest   |
| POST     | password/reset          | password.update | ...\Auth\ResetPasswordController@reset                   | web,guest   |
| GET|HEAD | password/reset/{token}  | password.reset  | ...\Auth\ResetPasswordController@showResetForm           | web,guest   |
| GET|HEAD | register                | register        | ...\Auth\RegisterController@showRegistrationForm         | web,guest   |
| POST     | register                |                 | ...\Auth\RegisterController@register                     | web,guest   |
+----------+-------------------------+-----------------+----------------------------------------------------------+-------------+
```

- POST password/reset – Form (previous slide) posts to here to actually reset the details.

- Okay – that was a lot of routes. But it is all done for you.

Authentication Controller

- All the authentication routes are serviced by various contollers in the Auth folder.
- You can guess what most of them do.
- But lets explore just one of them to get the idea.

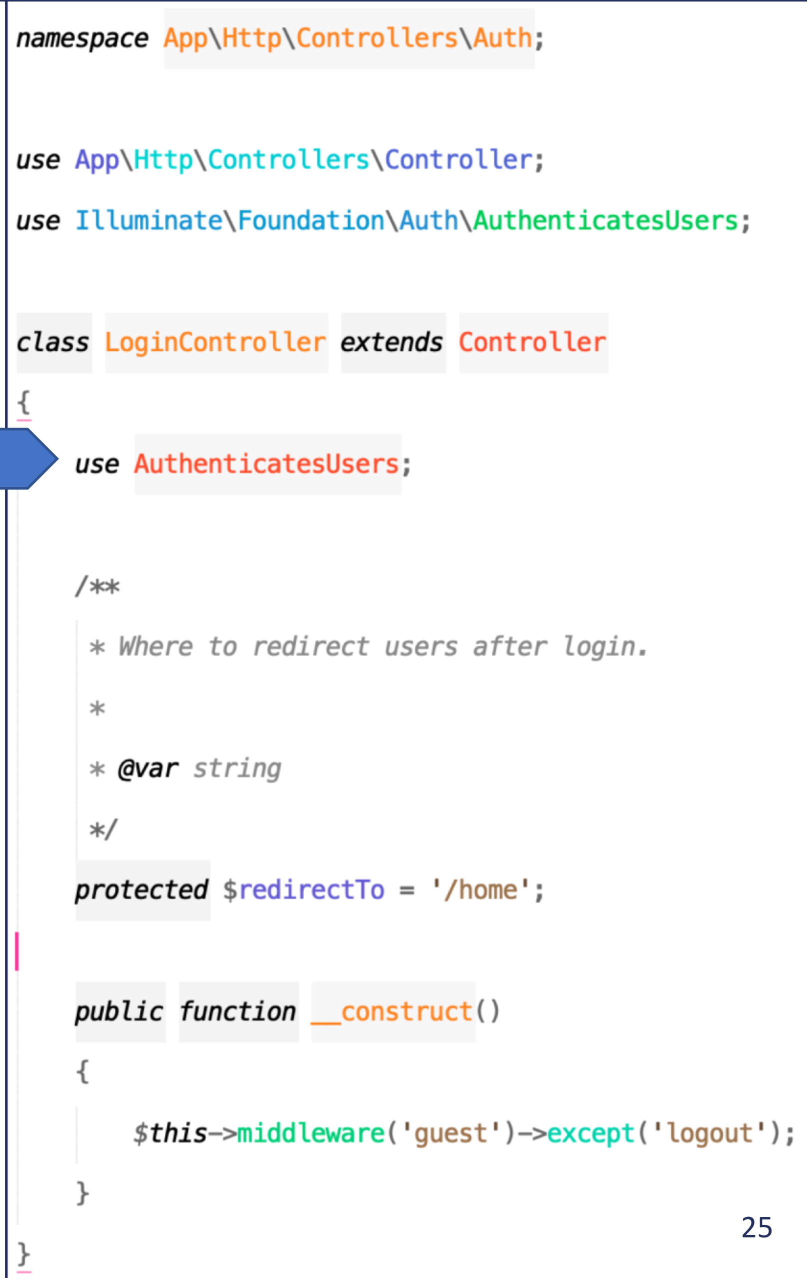


```
▼ swanseazoo
  ▼ app
    ▶ Console
    ▶ Exceptions
    ▼ Http
      ▼ Controllers
        ▼ Auth
            ForgotPasswordController.php
            LoginController.php
            RegisterController.php
            ResetPasswordController.php
            VerificationController.php
          AnimalController.php
          Controller.php
          HomeController.php
```

LoginController

- Wow it is small.
  - This is misleading.

- That use line actually causes the class to pull in trait. Similar to inheritance.
- We can change where ~~we~~ the controller redirects to.
- We will look at the middleware in the next session. But the idea here is that:
  - all methods in this controller can only be vested by guests (non-logged in users)
  - Except for logout (you need to be logged in to use that)



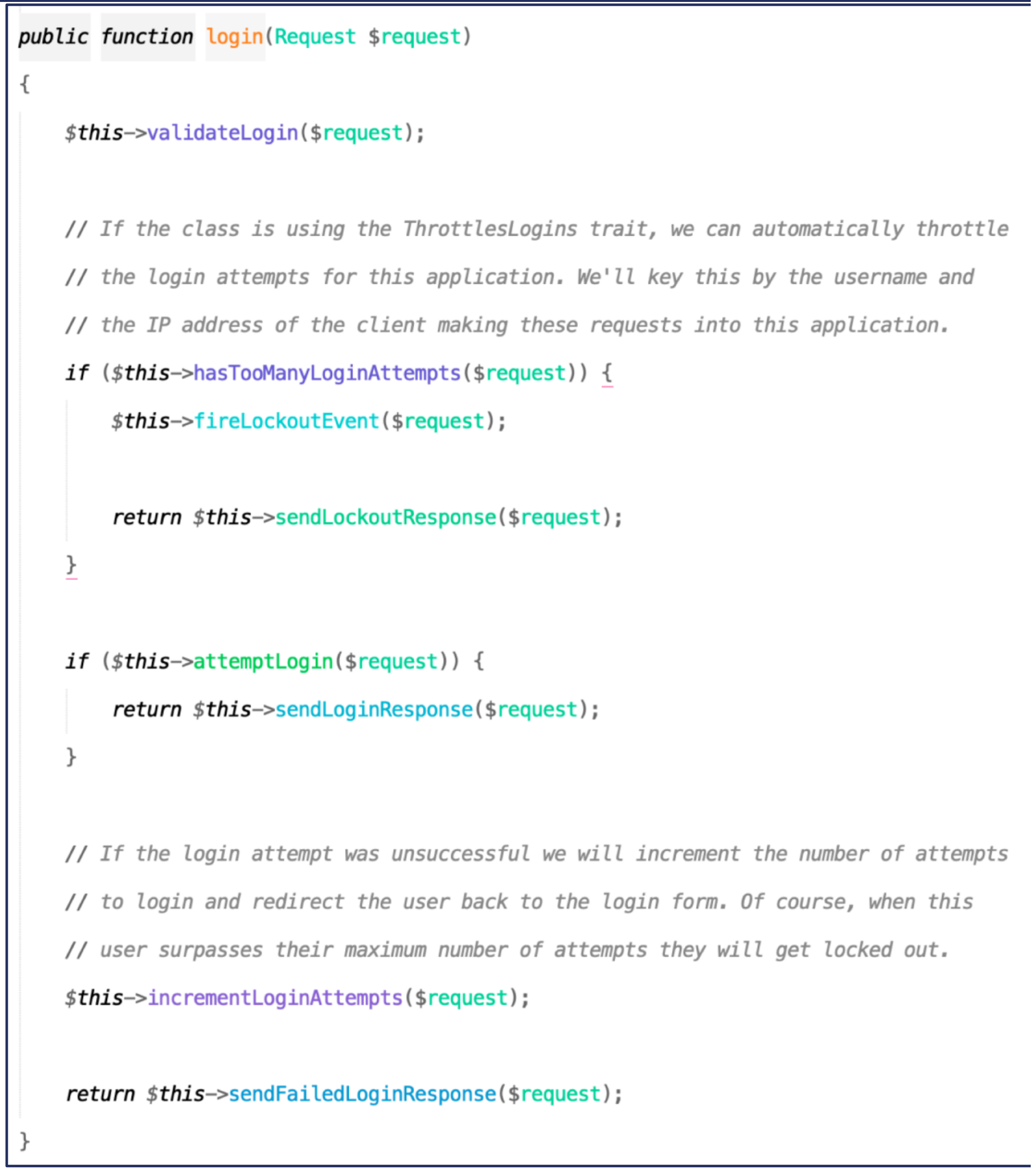
```php
namespace App\Http\Controllers\Auth;

use App\Http\Controllers\Controller;
use Illuminate\Foundation\Auth\AuthenticatesUsers;

class LoginController extends Controller
{
    use AuthenticatesUsers;

    /**
     * Where to redirect users after login.
     *
     * @var string
     */
    protected $redirectTo = '/home';

    public function __construct()
    {
        $this->middleware('guest')->except('logout');
    }
}
```

LoginController (Cont.)

Let’s look at the code for the trait /vendor/laravel/framework/src/Illuminate/Foundation/Auth/AuthenticatesUsers.php

Basically the login method does the following:

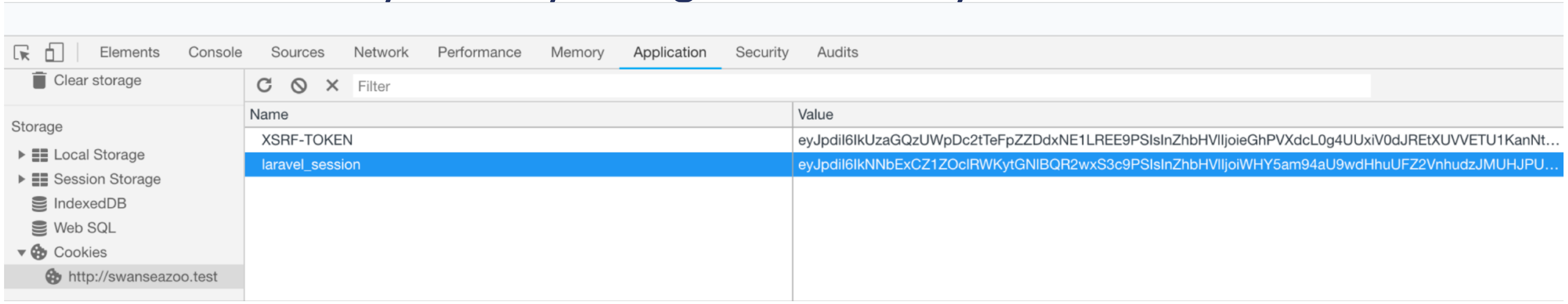
- It validates the login form data.
- If there have been too many attempts then you get locked out.
- Attempt to login
  - Match credentials.
- Redirect depending on result.



```
public function login(Request $request)
{
    $this->validateLogin($request);

    // If the class is using the ThrottlesLogins trait, we can automatically throttle
    // the login attempts for this application. We'll key this by the username and
    // the IP address of the client making these requests into this application.
    if ($this->hasTooManyLoginAttempts($request)) {
        $this->fireLockoutEvent($request);

        return $this->sendLockoutResponse($request);
    }

    if ($this->attemptLogin($request)) {
        return $this->sendLoginResponse($request);
    }

    // If the login attempt was unsuccessful we will increment the number of attempts
    // to login and redirect the user back to the login form. Of course, when this
    // user surpasses their maximum number of attempts they will get locked out.
    $this->incrementLoginAttempts($request);

    return $this->sendFailedLoginResponse($request);
}
```

Session IDs

- So how does this all work in terms of HTTP?
- Laravel will ask your browser to store a cookie storing a session key when you login sucessfully.



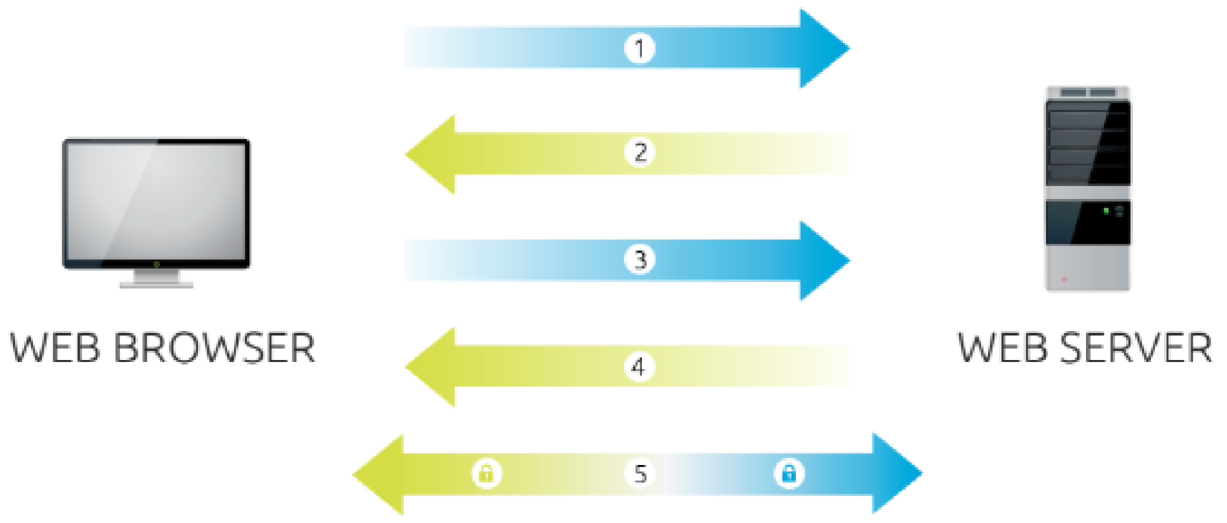
Name	Value
XSRF-TOKEN	eyJpdiI6IkUzaGQzUWpDc2tTeFpZZDdxNE1LREE9PSIsInZhbHVlIjoieGhPVXdcL0g4UUxiV0dJREtXUVVETU1KanNt…
laravel_session	eyJpdiI6IkNNbExCZ1ZOclRWKytGNlBQR2wxS3c9PSIsInZhbHVlIjoiWHY5am94aU9wdHhuUFZ2VnhudzJMUHJPU…

- Each request will contain this, so Laravel can match you up.
  - You had better use SSL/TLS to keep this secret.

- When you logout Laravel will invalidate the session key.
  - No longer accept it.

Secure Sockets Layer (SSL) – Technically now called Transport Layer Security

- Now we are moving towards sending sensitive data using web protocols we need to add some encryption
- SSL is a standard security technology for establishing an encrypted link between a server and client
- Normally HTTP is performed in plain text and is venerable to man-in-the-middle attacks (an attacker relays and alters the communication from client to server or vise versa)
- HTTPS tells the browser to encrypt the HTTP using SSL it connects to the server and starts an SSL handshake…



1. **Browser** connects to a web server (website) secured with SSL (https). Browser requests that the server identify itself.

2. **Server** sends a copy of its SSL Certificate, including the server's public key.

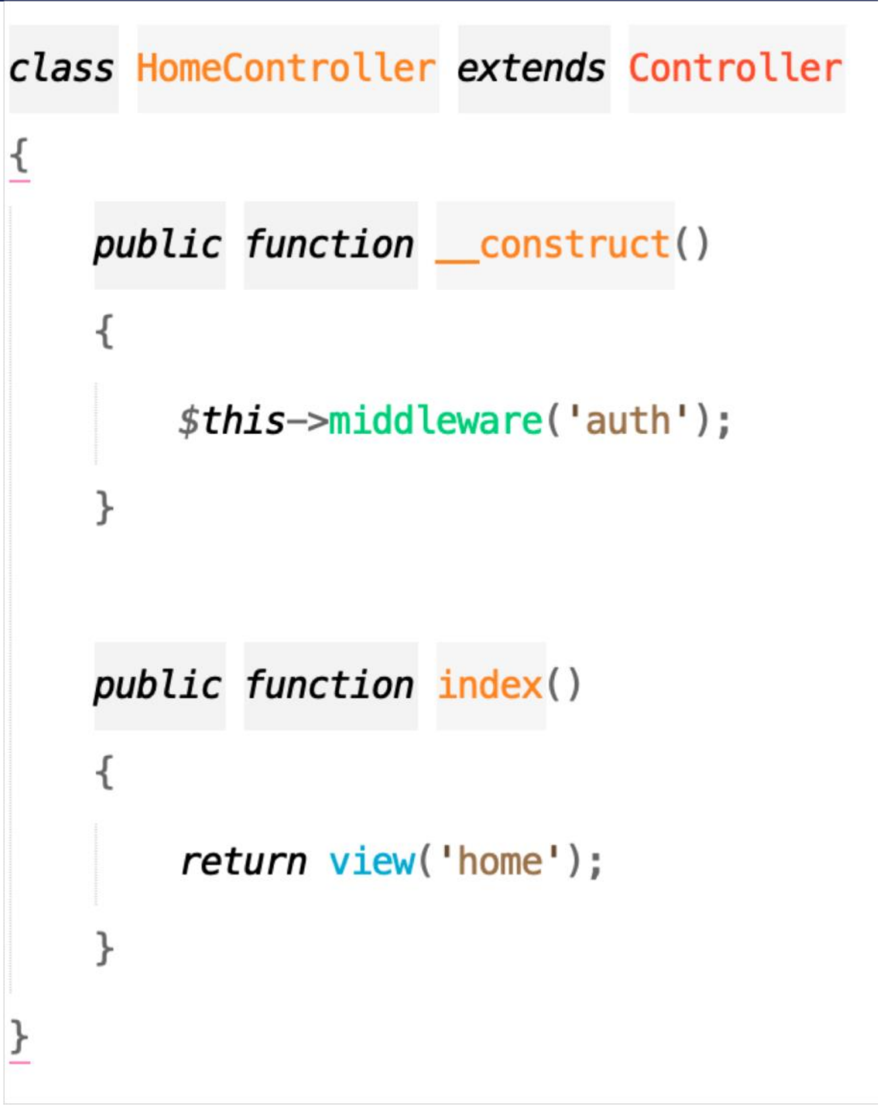
3. **Browser** checks the certificate root against a list of trusted CAs and that the certificate is unexpired, unrevoked, and that its common name is valid for the website that it is connecting to. If the browser trusts the certificate, it creates, encrypts, and sends back a symmetric session key using the server's public key.

4. **Server** decrypts the symmetric session key using its private key and sends back an acknowledgement encrypted with the session key to start the encrypted session.

5. **Server** and **Browser** now encrypt all transmitted data with the session key.

Protecting Routes

- Okay, so we can log in. How do we protect routes.
- HomeContoller’s constructor registers some middleware. This basically forces all methods in this class to run only if the user is authenticated.
- If not, then they get redirected to login.
- More on middleware in the next session.



```php
class HomeController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
    }

    public function index()
    {
        return view('home');
    }
}
```

Customising

- You can go and customise any of the scaffolding code.
  - It is there as a starting point.
- Example:
  - Laravel assumes you will login with an email address.
  - May be you want to use a username
  - Will have to update the auth controllers and views; and also the migration.
    - Really, its not that hard.