PIXEL & LINE

Laravel in Action
BSU 2015-09-15

Nathan Norton
nate@pixelandline.com

About Me

- Full Stack Web Developer, 5+ years
  - “If your company calls you a full stack developer, they don’t know how deep the stack is, and neither do you” - Coder’s Proverb
- Expertise/Buzz words:
  - PHP, Composer, ORM, Doctrine, Symfony, Silex, Laravel, OOP, Design Patterns, SOLID, MVC, TDD, PHPUnit, BDD, DDD, Build Automation, Jenkins, Git, Mercurial, Apache HTTPD, nginx, MySQL, NoSQL, MongoDB, CouchDB, memcached, Redis, RabbitMQ, beanstalkd, HTML5, CSS3, Bootstrap, Responsive design, IE Death, Javascript, NodeJS, Coffeescript, ES6, jQuery, AngularJS, Backbone.js, React, Asterisk, Lua, Perl, Python, Java, C/C++
- Enjoys:
  - Beer

About Pixel & Line

- Creative Agency
- Web development, mobile, development, and design
- Clients/projects include Snocru, Yale, Rutgers, UCSF, Wizard Den
- Every employee can write code
- PHP/Laravel, node, AngularJS, iOS/Android
- “It sucks ten times less to work at Pixel & Line than anywhere else I’ve worked” - Zack, iOS developer

Laravel

- Born in 2011 by Taylor Otwell
- MVC framework in PHP
- 83,000+ sites
- Convention over configuration
- Attempts to make working with PHP a joy
- Inspired by Ruby on Rails, ASP.NET, Symfony, and Sinatra
- Latest version 5.1, finally LTS

Laravel Features

- Eloquent ORM
- Artisan command runner
- Blade Templating engine
- Flexible routing
- Easy environment-based configuration
- Sensible migrations
- Testable
- Caching system
- IoC container for easy dependency injection
- Uses Symfony components
- Web documentation is great, tons of supplement

Laravel Impact

- PSR-2 coding standards
  - http://www.php-fig.org/psr/psr-2
- Use of Composer
  - Package manager for PHP
  - No more PEAR/PECL
  - https://getcomposer.org/
- Homestead
  - Development environment that “just works”
- Adoption of nginx
  - The community moves away from Apache HTTPD

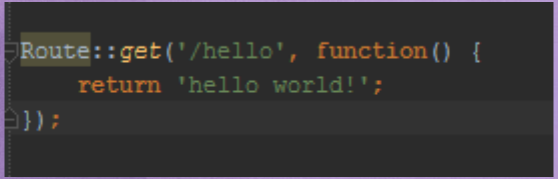
KEEP
CALM
LETS GET
THIS PARTY
STARTED

Laravel Homestead

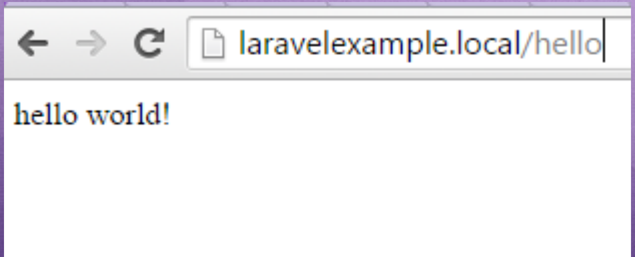
- Development environment that “just works”
- Uses Vagrant and Virtualbox
- No more “it worked on my machine”
- Edit files on Windows/Mac, test instantly on webserver
- Includes nginx, PHP 5.6, Composer, MySQL, PostgreSQL, Redis, Memcached, and Node on Ubuntu
- Keeps developers developing
- http://laravel.com/docs/5.1/homestead

Getting Started

- Clone project (inside the VM)
    - *composer create-project laravel/laravel laravelExample --prefer-dist*
- Add some routes
    - Edit app/Http/routes.php
- Enjoy!



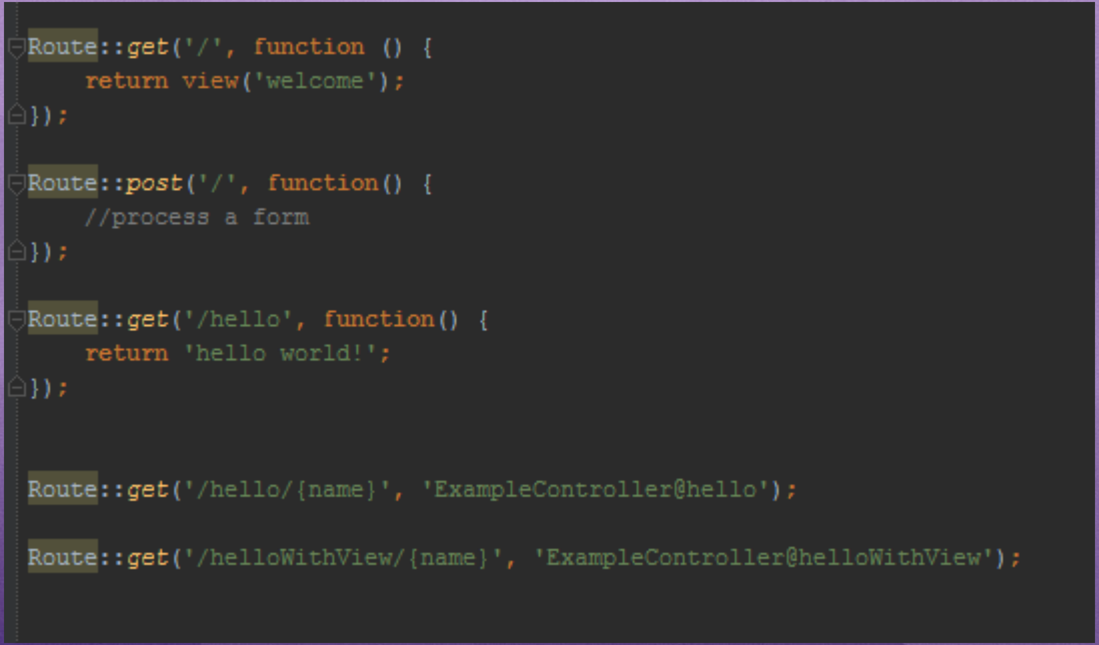
```
Route::get('/hello', function() {
    return 'hello world!';
});
```



```
hello world!
```

Routes

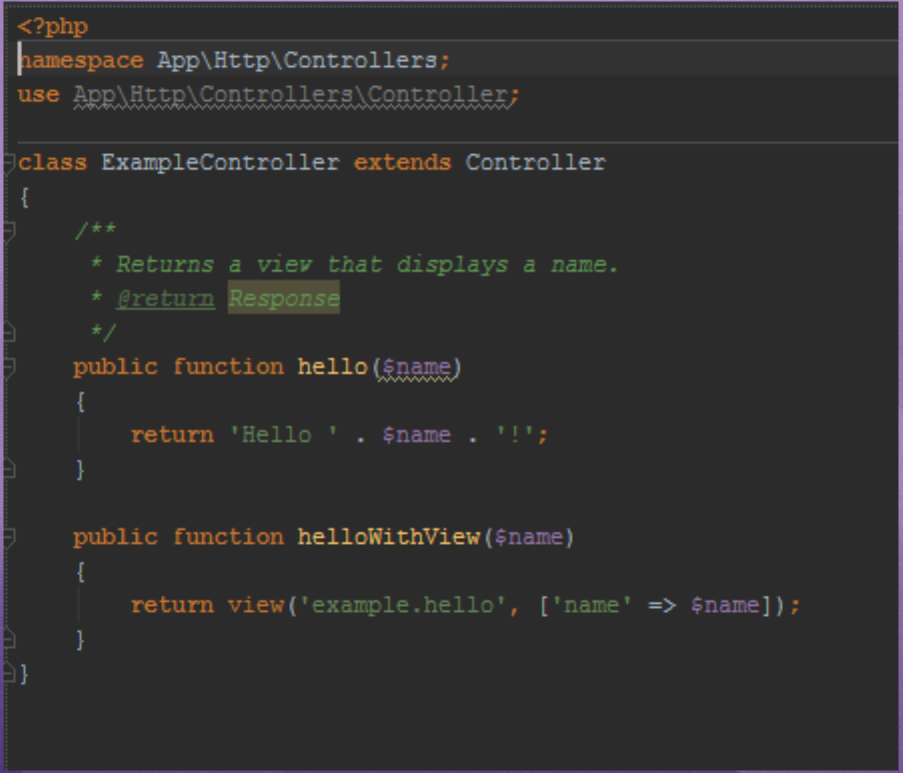
- Allow you to map URL’s to specific blocks of code
- app/Http/routes.php
- Route:: with HTTP verb
- Anonymous function
- OR controller



```php
Route::get('/', function () {
    return view('welcome');
});

Route::post('/', function() {
    //process a form
});

Route::get('/hello', function() {
    return 'hello world!';
});


Route::get('/hello/{name}', 'ExampleController@hello');

Route::get('/helloWithView/{name}', 'ExampleController@helloWithView');
```

Controllers

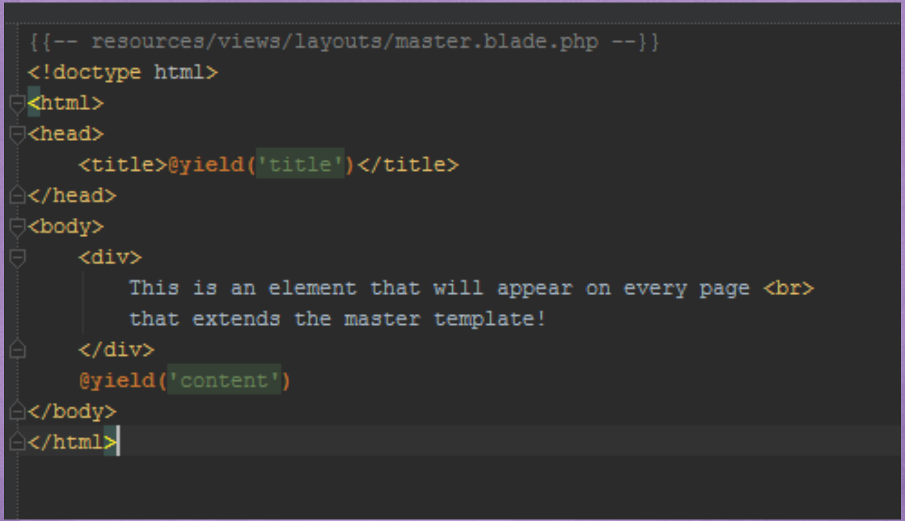
- Organizes resources into logical and semantic units
- app/Http/Controllers/
- php artisan make:controller OtherExampleController
- Can return plain text, Response object, or view



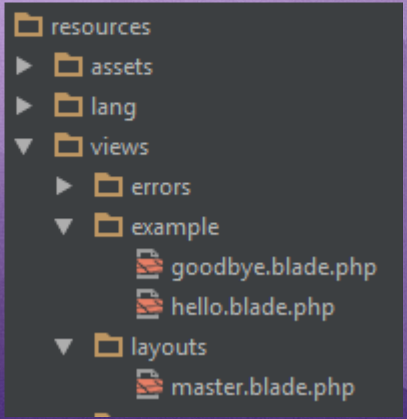
```php
<?php
namespace App\Http\Controllers;
use App\Http\Controllers\Controller;

class ExampleController extends Controller
{
    /**
     * Returns a view that displays a name.
     * @return Response
     */
    public function hello($name)
    {
        return 'Hello ' . $name . '!';
    }

    public function helloWithView($name)
    {
        return view('example.hello', ['name' => $name]);
    }
}
```

Views - Blade Templating

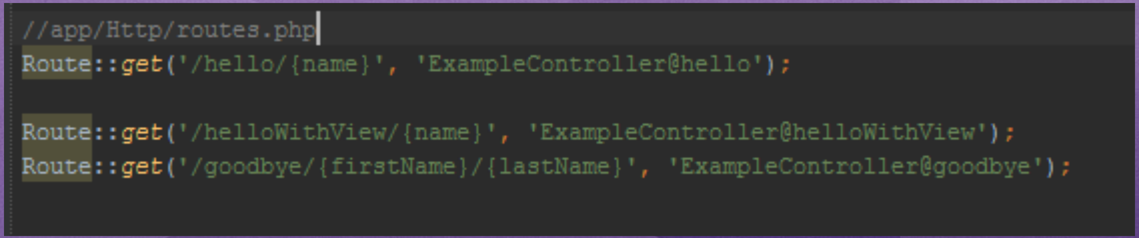
- Separates presentation from logic
- Borrows from other templating engines (Handlebars, Mustache)
- Allows you to break views into small, modular pieces
- resources/views



```
{{-- resources/views/layouts/master.blade.php --}}
<!doctype html>
<html>
<head>
    <title>@yield('title')</title>
</head>
<body>
    <div>
        This is an element that will appear on every page <br>
        that extends the master template!
    </div>
    @yield('content')
</body>
</html>
```

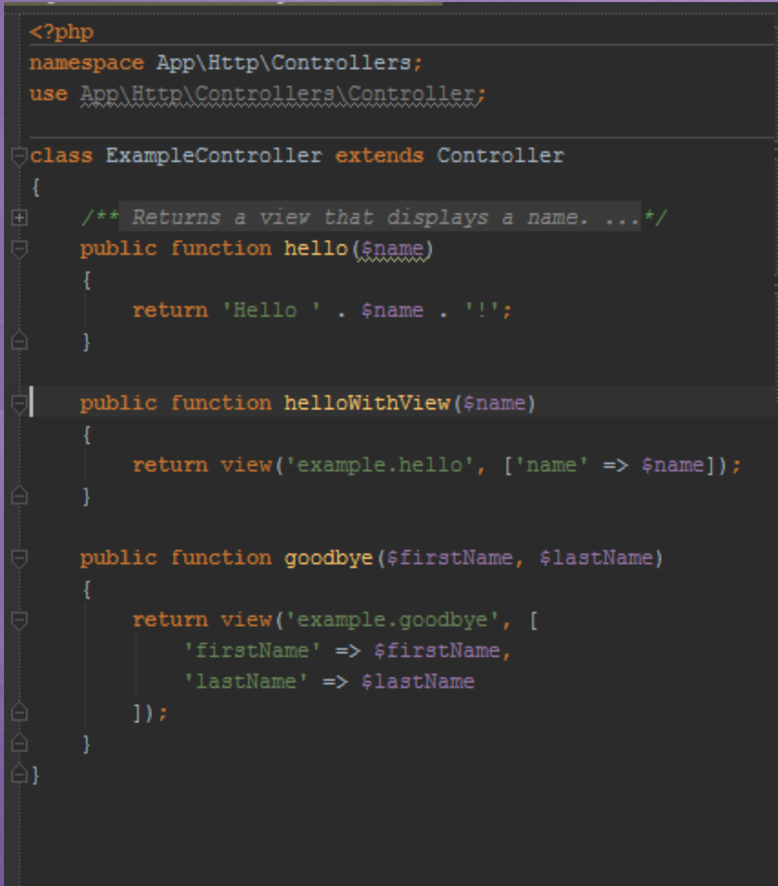


```
resources
  assets
  lang
  views
    errors
    example
      goodbye.blade.php
      hello.blade.php
    layouts
      master.blade.php
```

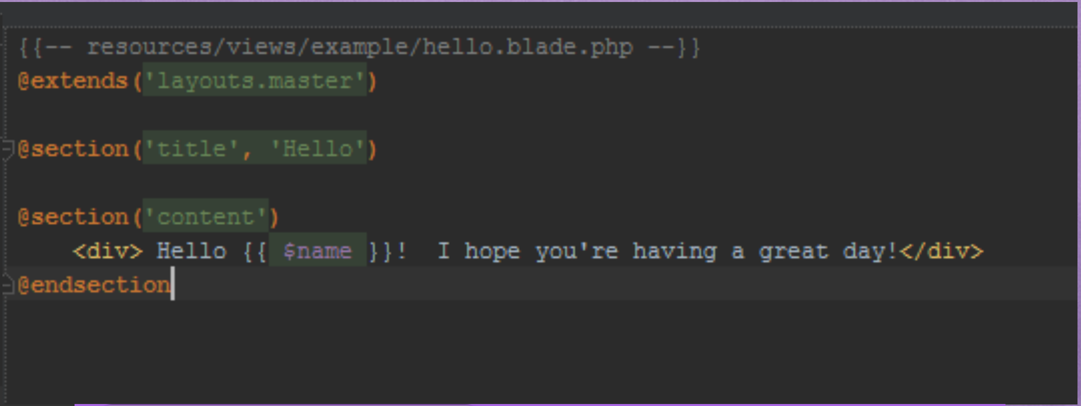


```
//app/Http/routes.php
Route::get('/hello/{name}', 'ExampleController@hello');

Route::get('/helloWithView/{name}', 'ExampleController@helloWithView');
Route::get('/goodbye/{firstName}/{lastName}', 'ExampleController@goodbye');
```

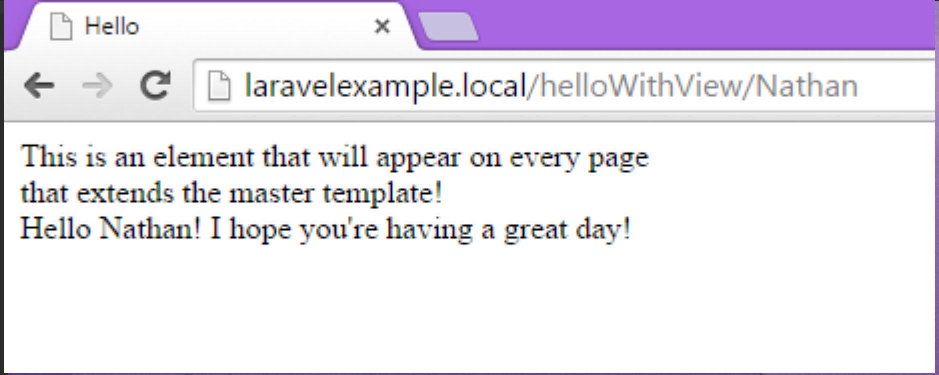
Views - Blade Templating (continued)



```php
<?php
namespace App\Http\Controllers;
use App\Http\Controllers\Controller;

class ExampleController extends Controller
{
    /** Returns a view that displays a name. ...*/
    public function hello($name)
    {
        return 'Hello ' . $name . '!';
    }

    public function helloWithView($name)
    {
        return view('example.hello', ['name' => $name]);
    }

    public function goodbye($firstName, $lastName)
    {
        return view('example.goodbye', [
            'firstName' => $firstName,
            'lastName' => $lastName
        ]);
    }
}
```

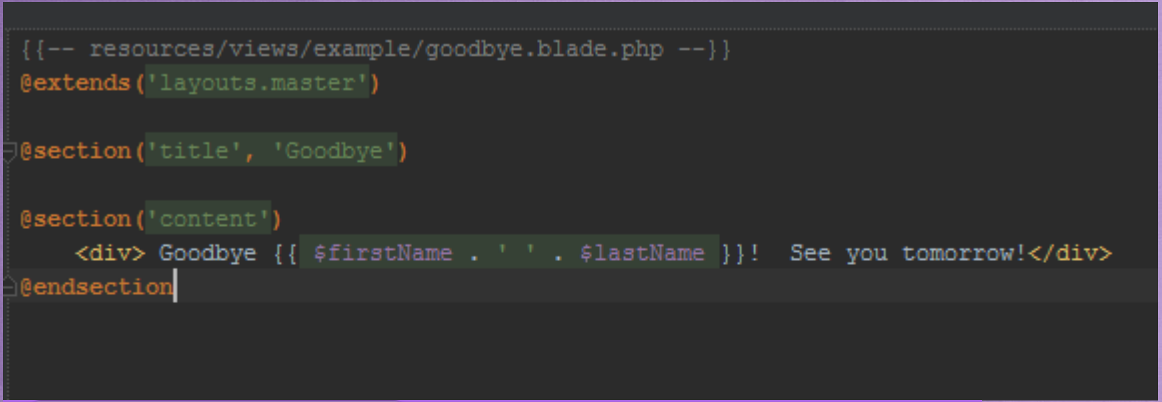


```
{{-- resources/views/example/hello.blade.php --}}
@extends('layouts.master')

@section('title', 'Hello')

@section('content')
    <div> Hello {{ $name }}!  I hope you're having a great day!</div>
@endsection
```

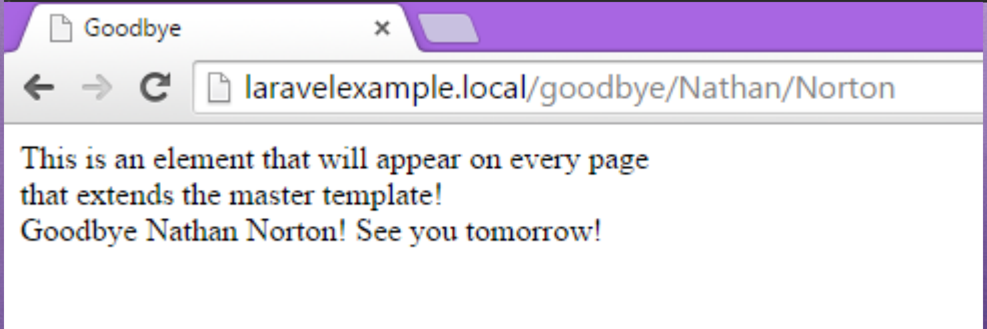


This is an element that will appear on every page
that extends the master template!
Hello Nathan! I hope you're having a great day!

Views - Blade Templating (continued)



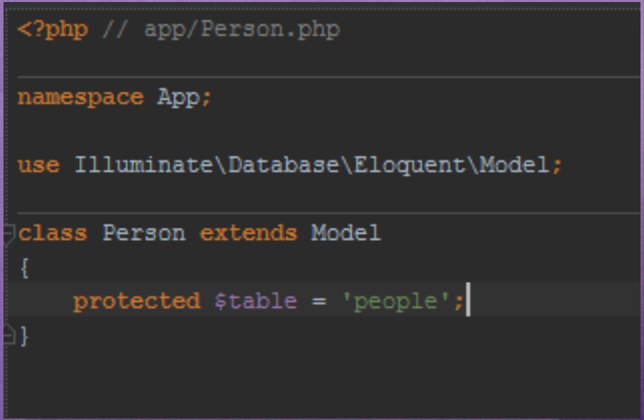
```
{{-- resources/views/example/goodbye.blade.php --}}
@extends('layouts.master')

@section('title', 'Goodbye')

@section('content')
    <div> Goodbye {{ $firstName . ' ' . $lastName }}!  See you tomorrow!</div>
@endsection
```



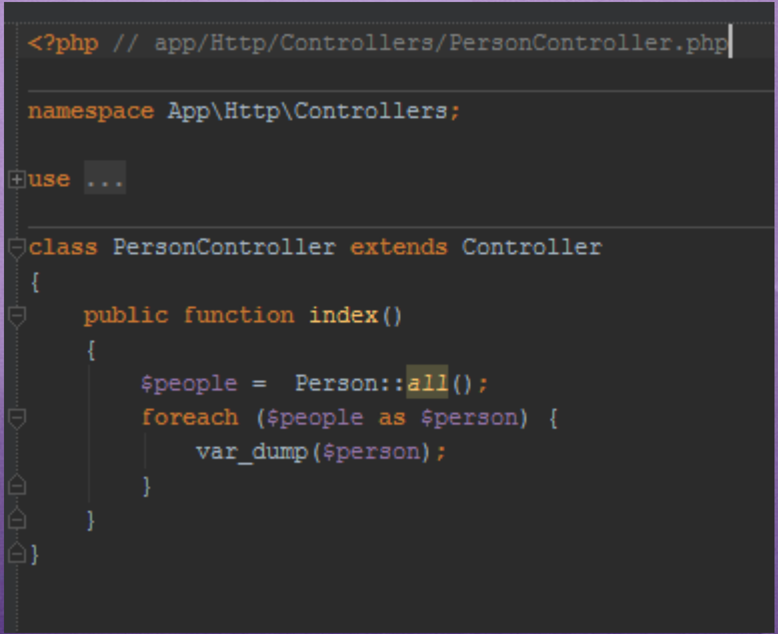
This is an element that will appear on every page
that extends the master template!
Goodbye Nathan Norton! See you tomorrow!

ORM - Eloquent

- ActiveRecord Pattern
- Adapts to schema
- Agnostic to MySQL/PostgreSQL
- php artisan make:model Person



```php
<?php // app/Person.php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Person extends Model
{
    protected $table = 'people';
}
```



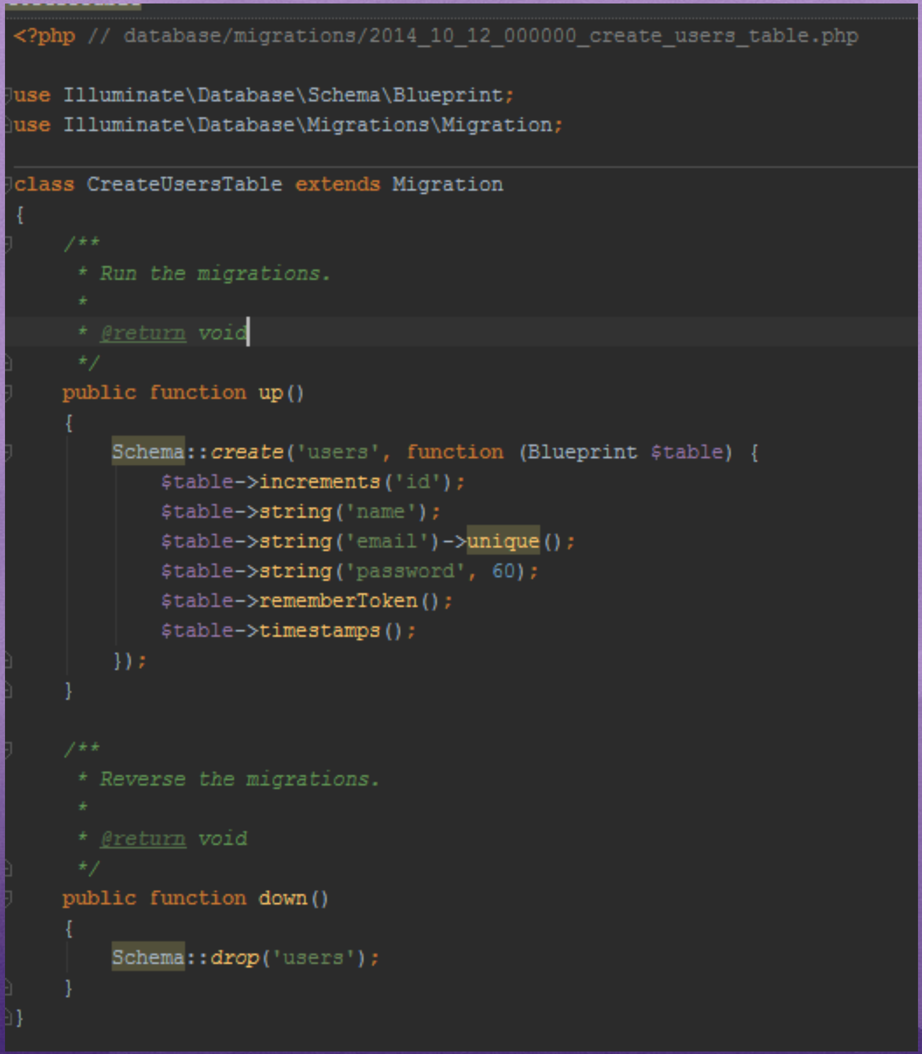
```php
<?php // app/Http/Controllers/PersonController.php

namespace App\Http\Controllers;

use ...

class PersonController extends Controller
{
    public function index()
    {
        $people =  Person::all();
        foreach ($people as $person) {
            var_dump($person);
        }
    }
}
```

Commands - Artisan

- Convenience functions
    - Controllers
    - Migrations
    - Models
    - Resources
    - Cache
    - Queues/Workers
- php artisan
- Custom commands
- http://laravel.com/docs/5.1/artisan

Migrations

- Manage changes in state of databases
- php artisan make:migration create_users_table
- Just define action for “up” and “down”
- php artisan migrate
- php artisan rollback



```php
<?php // database/migrations/2014_10_12_000000_create_users_table.php

use Illuminate\Database\Schema\Blueprint;
use Illuminate\Database\Migrations\Migration;

class CreateUsersTable extends Migration
{
    /**
     * Run the migrations.
     *
     * @return void
     */
    public function up()
    {
        Schema::create('users', function (Blueprint $table) {
            $table->increments('id');
            $table->string('name');
            $table->string('email')->unique();
            $table->string('password', 60);
            $table->rememberToken();
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     *
     * @return void
     */
    public function down()
    {
        Schema::drop('users');
    }
}
```

Extra Laravel Resources

- The official Laravel docs
  - Ensure version that corresponds with one you’re using
  - Plenty of examples
- Laracasts
  - 5-15 minutes video tutorials
  - Updated daily-ish
- Laravel.io
  - Laravel forums
- Reddit
  - http://reddit.com/r/laravel
- Stack Overflow