LARAVEL Pizzahouse Project

Route::get('/', function () {

    return view('welcome');

});

Route::get('/pizzas', function () {

    // return view('pizzas');  //return view

    // return "Pizas";     //return string

    // return ['name'=> 'veg pizza', 'base' => 'classic' ];   //return in json format

    //passing data to views

    // return view('pizzas',['type' => 'Hawaiian', 'base' => 'classic']);   //pass data as sec parameter to view ,lateron fron db

    // $pizza = [

    //     'type' => 'Hawaiian',

    //     'base' => 'classic',

    //     'price' => 20

    // ];

    // return view ('pizzas',$pizza);

    $pizzas = [

        ['type' => 'Hawaiian', 'base' => 'classic','price' => 20],

        ['type' => 'Veg garden', 'base' => 'cheesy crust','price' => 16],

        ['type' => 'Tandoori Chicken', 'base' => 'garlic crust','price' => 22]

    ];

    // query parameters in url pizzahouse.com/?name=maria

    $name = request('name');

    return view ('pizzas', [

        'pizzas' => $pizzas,

        'name' => $name,

        'age' => request('age')

    ]);

});

Laravel 6 Tutorial for Beginners #8

https://www.youtube.com/watch?v=LpJqxx5pNUk

**Making and including CSS file**

For styles, create a css folder in public and add main.css file there..add styles part from layouts/layout.css in main.css. Include this file in layout.blade.php as:

<link href =”/css/main.css” rel=”stylesheet”>

For including an image, make img folder in public and put image there

In Welcome.blade.php, add:

<img src="/img/logo\_trans.png" alt=" Pizza house" />

Laravel 6 Tutorial for Beginners #9 - Query Parameters

http://localhost:8000/pizzas?name=maria

?name=maria is query parameter

Route::get('/pizzas', function () {

 $pizzas = [

        ['type' => 'Hawaiian', 'base' => 'classic','price' => 20],

        ['type' => 'Veg garden', 'base' => 'cheesy crust','price' => 16],

        ['type' => 'Tandoori Chicken', 'base' => 'garlic crust','price' => 22]

    ];

    // query parameters in url pizzahouse.com/?name=maria

    $name = request('name');

    return view ('pizzas', [

        'pizzas' => $pizzas,

        'name' => $name

    ]);

Pizzas.blade.php

 <p>{{$name}} </p>

<http://localhost:8000/pizzas?name=mari>a will display maria in webpage

for passing two query parameters,

<http://localhost:8000/pizzas?name=maria&age=40>

web.blade.php

$name = request('name');

    return view ('pizzas', [

        'pizzas' => $pizzas,

        'name' => $name,

        'age' => request('age')

    ]);

Pizzas.blade.php

 <p>{{$name}} </p>

    <p>{{$age}} </p>

# Laravel 6 Tutorial for Beginners #10 - Route Parameters (wildcards)

Pizzahouse.com/pizzas/1

Pizzahouse.com/pizzas/maria

Pizzahouse.com/pizzas/{id}

Route::get('/pizzas/{id}', function ($id) {

//use id to query db for a record

return view('details', ['id' => $id]);

}

Details.blade.php

<!-- import layout styles file from layouts folder -->

@extends ('layouts.layout')

<!-- define a new section for all style content -->

@section('content')

<div class="flex-center position-ref full-height">

    <div class="content">

        <div class="title m-b-md">

            Pizza List - {{$id}}

        </div>

    </div>

</div>

@endsection

# Laravel 6 Tutorial for Beginners #11 - Controllers

Controller is special class that contain various route handler functions that we use in routes file instead of directly defining functionality in routes. Make code cleaner and understandable. We can create one controller for a group of routes e.g. a pizza controller for pizza route, a user controller for user routes

Route::get('/pizzas', function () {

    $pizzas = [

        ['type' => 'Hawaiian', 'base' => 'classic','price' => 20],

        ['type' => 'Veg garden', 'base' => 'cheesy crust','price' => 16],

        ['type' => 'Tandoori Chicken', 'base' => 'garlic crust','price' => 22]

    ];

    // query parameters in url pizzahouse.com/?name=maria

    $name = request('name');

    return view ('pizzas', [

        'pizzas' => $pizzas,

        'name' => $name,

        'age' => request('age')

    ]);

});

Route::get('/pizzas/{id}', function ($id) {

    //use id to query db for a record

    return view('details', ['id' => $id]);

});

PizzaController will handle these two routes.

In terminal, command

Php artisan make:controller PizzaController

A controller file is automatically created in App-http-contrllers

Move web.bundle.php roués handling to PizzaController under diff functions

class PizzaController extends Controller

{

    public function index(){

         $pizzas = [

        ['type' => 'Hawaiian', 'base' => 'classic','price' => 20],

        ['type' => 'Veg garden', 'base' => 'cheesy crust','price' => 16],

        ['type' => 'Tandoori Chicken', 'base' => 'garlic crust','price' => 22]

    ];

    // query parameters in url pizzahouse.com/?name=maria

    $name = request('name');

    return view ('pizzas', [

        'pizzas' => $pizzas,

        'name' => $name,

        'age' => request('age')

    ]);

    }

    public function show($id){

        return view('details', ['id' => $id]);

    }

}

In web.php

// PizzaConroller action index called in route

Route::get('/pizzas', 'PizzaController@index');

# Laravel 6 Tutorial for Beginners #12 - Connecting to MySQL

<http://localhost/phpmyadmin>

username:root

password: empty

create new database-pizzahouse (jut dropt it in loacalhot)or create new database from terminal

copy mysql path: C:\wamp64\bin\mysql\mysql5.7.26\bin

in search bar near windows icon, search env- goto environment vars for system-environment vars-path

new and paste C:\wamp64\bin\mysql\mysql5.7.26\bin

goto in terminal C:\wamp64\bin\mysql\mysql5.7.26\bin>mysql -u root

mysql> create database pizzahouse;

Query OK, 1 row affected (0.01 sec)

In .env file

DB\_CONNECTION=mysql

DB\_HOST=127.0.0.1

DB\_PORT=3306

DB\_DATABASE=pizzahouse

DB\_USERNAME=root

DB\_PASSWORD=

# Laravel 6 Tutorial for Beginners #13 - Migration Basics

PS C:\wamp64\www\laravel\pizzahouse> php artisan make:migration create\_pizzas\_table

class CreatePizzasTable extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('pizzas', function (Blueprint $table) {

            $table->increments('id');

            $table->timestamps();

            $table->string('type');

            $table->string('base');

            $table->string('name');

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('pizzas');

    }

}

To run this migration:

PS C:\wamp64\www\laravel\pizzahouse> php artisan migrate

<?php

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

class CreatePizzasTable extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('pizzas', function (Blueprint $table) {

            $table->increments('id');

            $table->timestamps();

            $table->string('type');

            $table->string('base');

            $table->string('name');

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('pizzas');

    }

}

Migration-users

<?php

use Illuminate\Support\Facades\Schema;

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'. 250)->unique();

            $table->string('password');

            $table->rememberToken();

            $table->timestamps();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('users');

    }

}

Migration-password reset

<?php

use Illuminate\Support\Facades\Schema;

use Illuminate\Database\Schema\Blueprint;

use Illuminate\Database\Migrations\Migration;

class CreatePasswordResetsTable extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::create('password\_resets', function (Blueprint $table) {

            $table->string('email')->index();

            $table->string('token');

            $table->timestamp('created\_at')->nullable();

        });

    }

    /\*\*

     \* Reverse the migrations.

     \*

     \* @return void

     \*/

    public function down()

    {

        Schema::dropIfExists('password\_resets');

    }

}

As migrate was not working, deleted users and password migrations and then run again

PS C:\wamp64\www\laravel\pizzahouse> php artisan migrate

Migration table created successfully.

Migrating: 2022\_02\_02\_213038\_create\_pizzas\_table

Migrated: 2022\_02\_02\_213038\_create\_pizzas\_table

you can use php artisan migrate:fresh this will ignore the current user table and recreate the tables from fresh.

# Laravel 6 Tutorial for Beginners #14 - More on Migrations

<https://stackoverflow.com/questions/36492917/php-artisan-migrate-not-creating-new-table>

php artisan enter

will show diff help in commands..can acheck migrate part

PS C:\wamp64\www\laravel\pizzahouse> php artisan migrate:status

| Ran? | Migration |

+------+---------------------------------------+

| Y | 2022\_02\_02\_213038\_create\_pizzas\_table |

PS C:\wamp64\www\laravel\pizzahouse> php artisan migrate:rollback

Rolling back: 2022\_02\_02\_213038\_create\_pizzas\_table

Rolled back: 2022\_02\_02\_213038\_create\_pizzas\_table

PS C:\wamp64\www\laravel\pizzahouse> php artisan migrate:status

+------+---------------------------------------+

| Ran? | Migration |

+------+---------------------------------------+

| N | 2022\_02\_02\_213038\_create\_pizzas\_table |

+------+---------------------------------------+

PS C:\wamp64\www\laravel\pizzahouse> php artisan migrate

Migrating: 2022\_02\_02\_213038\_create\_pizzas\_table

Migrated: 2022\_02\_02\_213038\_create\_pizzas\_table

To add a new column, two options:

1. add that column in migration file and then migrate rollback and then migrate again
2. best and safe option is to create new migration

PS C:\wamp64\www\laravel\pizzahouse> php artisan make:migration add\_price\_to\_pizzas\_table

Created Migration: 2022\_02\_02\_225555\_add\_price\_to\_pizzas\_table

In this new migration, add function:

class AddPriceToPizzasTable extends Migration

{

    /\*\*

     \* Run the migrations.

     \*

     \* @return void

     \*/

    public function up()

    {

        Schema::table('pizzas', function (Blueprint $table) {

            $table->integer('price');

        });

    }

Now run migrate again(if didn’t work, drop tables from phpmyadmin, rollback and then migrate)

Rollback will only rollback the recent one, if use rollback again, earlier ones will be rolled back

# Laravel 6 Tutorial for Beginners #15 - Eloquent Models

Model is code representation for specific tables in database

e.g. find(), save(), destroy() using models on database

PS C:\wamp64\www\laravel\pizzahouse> php artisan make:model Pizza

Model created successfully.

This modela utromatically connected to pizzas table. If we want to connect it with some other table, thn inside Pizza.php model, write

Protected $table = “some\_table”;

Now in our web.bundle.php ,its using index fn of controller which has hard-coded data. What if we want to get this data from db..we have to use model.

In PizzaContrloller.php

use Illuminate\Http\Request;

use App\Pizza;

in PizzaController.php

 $pizzas = Pizza::all();

return view('pizzas', ['pizzas' => $pizzas] );

in pizzas.blade.php

 @foreach($pizzas as $pizza)

            <div>

                {{$pizza->name }} {{ $pizza-> type}} - {{$pizza-> base}}

            </div>

        @endforeach