Contents

[LESSON 1: PHP INSTALLATION 2](#_Toc185295575)

[a. Installing multiple php versions 2](#_Toc185295576)

[LESSON 2: LARAVEL CONTROLLERS 2](#_Toc185295577)

[a. Types of Controllers in Laravel: 3](#_Toc185295578)

[**Basic Controllers:** These are simple classes with methods that handle specific HTTP requests. They are suitable for handling individual actions or simple use cases. **Artisan Command** php artisan make:controller AboutController **Controller code** <?php 3](#_Toc185295579)

[**Resource Controllers:** These controllers are designed to handle all CRUD (Create, Read, Update, Delete) operations for a specific resource. They provide a convenient way to define RESTful APIs. **Artisan Command** php artisan make:controller PostController --resource **Controller code** <?php 3](#_Toc185295580)

[**Form Request Controllers:** These controllers are used to validate incoming form requests and provide a more structured approach to input validation. 4](#_Toc185295581)

# LESSON 1: PHP INSTALLATION

## Installing multiple php versions

1. Download the php version you prefer
2. Unzip the binary file
3. Copy the unzipped directory/folder to a safe place (preferably Drive C)
4. Open the unzipped php directory and locate to the php executable file
5. Copy the path and add to Environment variables (**System Variables**)
6. When all the above are done, you can easily switch your php versions

# LESSON 2: LARAVEL CONTROLLERS

In Laravel, controllers are classes that handle the logic of incoming HTTP requests and generate appropriate responses.

**Key Roles of Controllers in Laravel:**

1. **Handling HTTP Requests:** Controllers receive incoming HTTP requests (GET, POST, PUT, DELETE, PATCH etc.) and process them according to their defined logic.
2. **Processing Request Data:** They extract and validate data from the request, such as form input, query parameters, or request body.
3. **Interacting with Models:** Controllers often interact with models to retrieve or manipulate data from the database.
4. **Generating Responses:** Based on the processed data and business logic, controllers generate appropriate responses, which can be HTML views, JSON data, redirects, or other formats.
5. **Routing Requests:** Controllers are linked to specific routes defined in the routes directory. When a request matches a route, the corresponding controller method is executed.

## Types of Controllers in Laravel:

### Basic Controllers: These are simple classes with methods that handle specific HTTP requests. They are suitable for handling individual actions or simple use cases.

#### Artisan Command php artisan make:controller AboutController

#### Controller code <?php

namespace App\Http\Controllers;

use Illuminate\Http\Request;

class AboutController extends Controller

{

public function index()

{

return view('about');

}

}

#### Route Route::get('/about', [AboutController::class, 'index']);

### **Resource Controllers:** These controllers are designed to handle all CRUD (Create, Read, Update, Delete) operations for a specific resource. They provide a convenient way to define RESTful APIs.

#### Artisan Command php artisan make:controller PostController –resource

#### Controller code

<?php

namespace App\Http\Controllers;

use App\Models\Post;

use Illuminate\Http\Request;

class PostController extends Controller

{

/\*\*

\* Display a listing of the resource.

\*/

public function index()

{

$posts = Post::all();

return view('posts.index', compact('posts'));

}

// ... other CRUD methods (create, store, show, edit, update, destroy)

}

#### Route

Route::resource('posts', PostController::class);

### **Form Request Controllers:** These controllers are used to validate incoming form requests and provide a more structured approach to input validation.

#### Artisan Command

php artisan make:request StorePostRequest

#### Controller Code

<?php

namespace App\Http\Requests;

use Illuminate\Foundation\Http\FormRequest;

class StorePostRequest extends FormRequest

{

/\*\*

\* Determine if the user is authorized to make this request.

\*/

public function authorize(): bool

{

return true;

}

/\*\*

\* Get the validation rules that apply to the request.

\*/

public function rules(): array

{

return [

'title' => 'required|string|max:255',

'body' => 'required|string',

];

}

}

#### Usage in controller

public function store(StorePostRequest $request)

{

Post::create($request->validated());

return redirect()->route('posts.index');

}