## Getting Started with Laravel: A Beginner-Friendly Guide

Laravel is a tool that helps you build websites and web applications easily. It is based on PHP, a programming language used to create dynamic web pages. This guide will explain everything step-by-step, using simple words, so you can follow along even if you’re new.

**What is Laravel?**

Laravel is a framework. Think of it like a box of tools and instructions that helps you build a website faster and better. For example:

* Instead of writing long and repetitive code, Laravel gives you shortcuts.
* It organizes your project so your code is clean and easy to manage.
* It includes built-in tools for common tasks like user login, database management, and sending emails.

**What Do You Need to Start?**

Before using Laravel, make sure you have:

1. **PHP Installed**:
   * PHP is the programming language Laravel is built on. You need version 8.0 or higher.
2. **Composer**:
   * Composer is a tool that helps you install Laravel and manage its dependencies (extra tools Laravel needs to work).
3. **A Text Editor**:
   * Use a program like Visual Studio Code to write your code.
4. **A Database**:
   * A database stores your website's information, like user accounts or product details. MySQL is commonly used with Laravel.
5. **Web Server**:
   * Laravel includes a built-in server to help you test your website locally (on your computer).

**How to Install Laravel**

**Step 1: Install Composer**

Composer is like an app store for Laravel. To install it:

1. Open your terminal (Command Prompt on Windows).
2. Type this command and press Enter:
3. curl -sS https://getcomposer.org/installer | php

mv composer.phar /usr/local/bin/composer

This downloads Composer to your computer.

**Step 2: Create a New Laravel Project**

Use Composer to create a new Laravel project. In your terminal, type:

composer create-project --prefer-dist laravel/laravel my-project

Replace my-project with the name of your project.

**Step 3: Start Laravel**

Go into your project folder and start Laravel’s built-in server:

cd my-project

php artisan serve

This starts a web server. Open your browser and go to http://localhost:8000. You’ll see the Laravel welcome page.

**How Laravel Works (Step-by-Step)**

Here are the key terms you’ll encounter and what they mean:

**1. Routes:**

* Routes are like roadmaps. They tell Laravel what to do when someone visits a specific web address.
* Example in plain PHP: Route::get('/tasks', [TaskController::class, 'index']);
* Route::post('/tasks', [TaskController::class, 'store']);
* In Laravel, you define routes in the *routes/web.php file:*

**2. Controllers:**

* Controllers handle the logic of your application. They decide what data to show and what happens when users interact with your website.
* Example in plain PHP:

*<?php*

*namespace App\Http\Controllers;*

*use App\Models\Task;*

*use Illuminate\Http\Request;*

*class TaskController extends Controller{*

*public function index(){*

*$tasks = Task::latest()->get();// Gets tasks ordered by newest first*

*return view('index', compact('tasks')); }*

*public function store(Request $request){*

*$request->validate(['name' => 'required|string|max:255'  // Basic validation ]);*

*Task::create($request->only('name'));  // Safer than passing all request data*

*return redirect('/tasks');}}*

* In Laravel, you create a controller using run :*php artisan make:controller TaskController*
* ***Creating model run*** *: php artisan make:model Task -m*
* ***Then Run :*** *php artisan migrate*
* ***Creating tables****:php artisan session:table*
* *php artisan migrate*

Then, define logic in the controller and link it to a route:

**3. Views:**

* Views are the HTML pages your users see. They separate the design (HTML) from the logic (PHP).
* Example in plain PHP:*echo '<h1>Welcome to my website!</h1>';*
* In Laravel, you create views in the *resources/views folder:*

Link it to a route:Route::get('/tasks', [TaskController::class, 'index']);

**Example:***<h1>To-Do List</h1>*

*<form method="POST" action="/tasks">*

*@csrf*

*<input type="text" name="name" placeholder="New Task" required>*

*<button type="submit">Add</button>*

*</form>*

*<ul>@foreach ($tasks as $task)*

*<li>{{ $task->name }}</li>*

*@endforeach*

*</ul>*

**4. Database:**

* Laravel uses a database to store information, like user data or product details.
* In plain PHP:
* $conn = new mysqli($servername, $username, $password, $dbname);
* $sql = "INSERT INTO tasks (name) VALUES ('Learn PHP')";

$conn->query($sql);

* In Laravel, you configure the database in the .env file:
* DB\_CONNECTION=mysql
* DB\_HOST=127.0.0.1
* DB\_PORT=3306
* DB\_DATABASE=your\_database\_name
* DB\_USERNAME=your\_username

DB\_PASSWORD=your\_password

**5. Migrations:**

* Migrations are like version control for your database. They help you create or modify database tables easily.
* Example in Laravel:

*php artisan make:migration create\_tasks\_table*

Add table columns:

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

*$table->id();*

$table->string('name');

$table->timestamps();

});

Run the migration:

*php artisan migrate*

**Building a Simple To-Do App**

Let’s create a simple to-do list to understand Laravel better.

**Step 1: Define Routes**

Add routes in routes/web.php:

*Route::get('/tasks', [TaskController::class, 'index']);*

*Route::post('/tasks', [TaskController::class, 'store']);*

**Step 2: Create a Controller**

Run this command:

*php artisan make:controller TaskController*

*In* ***app/Http/Controllers/TaskController.php,***

*add:*

***use App\Models\Task;***

*public function index()*

*{*

*$tasks = Task::latest()->get();  // Gets tasks ordered by newest first*

*return view('index', compact('tasks'));*

*}*

*public function store(Request $request)*

*{*

*$request->validate([*

*'name' => 'required|string|max:255'  // Basic validation*

*]);*

*Task::create($request->only('name'));  // Safer than passing all request data*

*return redirect('/tasks');*

*}*

**Step 3 create Task**

**app/model/Task.php**

1.Run in new Terminal:***php artisan make:model Task -m***

1. ***Open the migration file (in database/migrations/)***

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

*$table->id();*

*$table->string('name');// This matches what your controller expects*

*$table->boolean('completed')->default(false);// Optional: for tracking completion status*

*$table->timestamps();// Creates created\_at and updated\_at columns});*

**4.Run the Migration**

*php artisan migrate*

**Step 4: Update Your Task Model**

Edit **app/Models/Task.php**to include fillable fields:

*<?php*

*namespace App\Models;*

*use Illuminate\Database\Eloquent\Factories\HasFactory;*

*use Illuminate\Database\Eloquent\Model;*

*class Task extends Model*

*{use HasFactory;*

*protected $fillable = ['name',*

*'completed'  // Include this if you added it to your migration*

*];}*

**Step 4: Create a View**

Create resources/views/tasks/index.blade.php:

<!DOCTYPE html>

<html>

<head>

<title>To-Do List</title>

</head>

<body>

<h1>To-Do List</h1>

<form method="POST" action="/tasks">

@csrf

<input type="text" name="name" placeholder="New Task" required>

<button type="submit">Add</button>

</form>

<ul>

@foreach ($tasks as $task)

<li>{{ $task->name }}</li>

@endforeach

</ul>

</body>

</html>

**Learning Tips**

1. **Start Small**: Build small projects to understand the basics.
2. **Use Laravel Docs**: The [official documentation](https://laravel.com/docs) is clear and helpful.
3. **Practice Often**: The more you code, the better you’ll understand.

Now you’re ready to start building with Laravel! Take it one step at a time, and enjoy coding!