



Getting Started

Installation

Install Lumen via Composer:

```
composer create-project --prefer-dist  
laravel/lumen <project-name>
```

Navigate to your project directory:

```
cd <project-name>
```

Run the local development server:

```
php -S localhost:8000 -t public
```

Basic Configuration

.env file: Configure environment-specific settings, such as database connections and API keys. Example:

```
APP_DEBUG=true  
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=lumen_db  
DB_USERNAME=homestead  
DB_PASSWORD=secret
```

bootstrap/app.php: Configure middleware, service providers, and exception handling.

```
$app->withFacades();  
$app->register(App\Providers\AppServiceProvider::class);
```

Enable Facades: Uncomment `$app->withFacades();` to enable facades.

```
$app->withFacades();
```

Enable Eloquent: Uncomment `$app->withEloquent();` to enable Eloquent ORM.

```
$app->withEloquent();
```

Routing

Define routes in `routes/web.php` or `routes/api.php`.

Example:

```
$router->get('/', function () { return 'Hello, Lumen!'; });
```

Available HTTP methods: `get`, `post`, `put`, `patch`, `delete`, `options`.

Route parameters:

```
$router->get('/user/{id}', function ($id) {  
    return 'User ID: ' . $id; });
```

Controllers & Middleware

Controllers

Generate a controller:

```
php artisan make:controller <ControllerName>
```

Example Controller:

```
namespace App\Http\Controllers;  
  
use Illuminate\Http\Request;  
  
class UserController extends Controller  
{  
    public function show($id)  
    {  
        return 'User ID: ' . $id;  
    }  
}
```

Register the controller route:

```
$router->get('/user/{id}',  
'UserController@show');
```

Middleware

Creating Middleware:	Generate middleware: <pre>php artisan make:middleware <MiddlewareName></pre>
Example Middleware:	<pre>namespace App\Http\Middleware; use Closure; class ExampleMiddleware { public function handle(\$request, Closure \$next) { // Perform action return \$next(\$request); } }</pre>
Registering Middleware:	Register in <code>bootstrap/app.php</code> : <pre>\$app->routeMiddleware(['example' => App\Http\Middleware\ExampleMid dleware::class]);</pre>
Applying Middleware:	Apply to a route: <pre>\$router->get('/admin', ['middleware' => 'example', function () { return 'Admin Area'; }]);</pre>

Request & Response

Accessing Request Data:	<pre>use Illuminate\Http\Request; \$router->post('/submit', function (Request \$request) { \$name = \$request->input('name'); return 'Hello, ' . \$name; });</pre>
Returning JSON Responses:	<pre>return response()->json(['message' => 'Success', 'data' => []], 200);</pre>
Setting Headers:	<pre>return response('Hello World')- >header('Content-Type', 'text/plain');</pre>

Database & Models

Database Configuration

Configure database settings in <code>.env</code> file:
<pre>DB_CONNECTION=mysql DB_HOST=127.0.0.1 DB_PORT=3306 DB_DATABASE=lumen_db DB_USERNAME=homestead DB_PASSWORD=secret</pre>
Ensure Eloquent is enabled in <code>bootstrap/app.php</code> :
<pre>\$app->withEloquent();</pre>

Eloquent Models

Creating a Model:	<pre>namespace App; use Illuminate\Database\Eloquent\Mo del; class User extends Model { protected \$fillable = ['name', 'email']; protected \$table = 'users'; }</pre>
Using a Model:	<pre>use App\User; \$router->get('/users', function () { \$users = User::all(); return response()- >json(\$users); });</pre>
Model Attributes:	<ul style="list-style-type: none"><code>\$fillable</code> - attributes that can be mass-assigned.<code>\$table</code> - the table associated with the model.<code>\$primaryKey</code> - the primary key column.<code>\$timestamps</code> - whether to maintain timestamps (<code>created_at</code>, <code>updated_at</code>).

Database Migrations

Create a migration:
<pre>php artisan make:migration create_users_table</pre>
Example migration file:
<pre>use Illuminate\Database\Migrations\Migration; use Illuminate\Database\Schema\Blueprint; use Illuminate\Support\Facades\Schema; class CreateUsersTable extends Migration { public function up() { Schema::create('users', function (Blueprint \$table) { \$table->id(); \$table->string('name'); \$table->string('email')->unique(); \$table->timestamps(); }); } public function down() { Schema::dropIfExists('users'); } }</pre>
Run migrations:
<pre>php artisan migrate</pre>

Authentication & Authorization

Authentication

Lumen provides basic authentication scaffolding. Consider using Passport or JWT for more robust authentication.

Using basic authentication middleware:

```
$router->get('/profile', ['middleware' =>
'auth', function () {
    return 'User Profile';
}]);
```

Authorization (Policies)

Creating a Policy:

```
namespace App\Policies;

use App\User;
use App\Post;

class PostPolicy
{
    public function
    update(User $user, Post
$post)
    {
        return $user->id ===
$post->user_id;
    }
}
```

Registering a Policy: Register the policy in `AuthServiceProvider`:

```
protected $policies = [
    App\Post::class =>
App\Policies\PostPolicy::clas
s,
];
```

Using a Policy:

```
use
Illuminate\Support\Facades\Gate;

$router->put('/post/{id}',
function ($id) {
    $post =
App\Post::findOrFail($id);
    if
(Gate::allows('update',
$post)) {
        // Update the post
    }
});
```

Events & Listeners

Define events and listeners to decouple components.

Example Event:

```
namespace App\Events;

use App\User;
use Illuminate\Queue\SerializesModels;

class UserRegistered
{
    use SerializesModels;

    public $user;

    public function __construct(User $user)
    {
        $this->user = $user;
    }
}
```

Example Listener:

```
namespace App\Listeners;

use App\Events\UserRegistered;

class SendWelcomeEmail
{
    public function handle(UserRegistered
$event)
    {
        // Send welcome email to $event->user
    }
}
```

Register events and listeners in

`app/Providers/EventServiceProvider.php`.