

GABRIELE ROMANATO

Menu

Implementare un URL Shortener con Laravel

Gli URL shortener sono strumenti utili che trasformano lunghi indirizzi web in versioni più brevi e facili da condividere. In questo articolo, vedremo come implementare un URL shortener utilizzando Laravel, un potente framework PHP.

Per prima cosa, crea un nuovo progetto Laravel utilizzando Composer. Apri il terminale e digita:

```
composer create-project --prefer-dist laravel/laravel url-shortener
```

Configura il database aprendo il file `.env` e aggiornando le seguenti variabili con i dettagli del tuo database:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=url_shortener
DB_USERNAME=username
DB_PASSWORD=password
```

Assicurati di aver creato un database chiamato `url_shortener`.

Ora, creiamo una migrazione per la tabella `urls` che conterrà gli URL originali e quelli accorciati. Esegui il comando:

```
php artisan make:migration create_urls_table
```

Questo comando genererà un file di migrazione in `database/migrations`. Apri questo file e modifica il metodo `up` come segue:

```
public function up()
{
    Schema::create('urls', function (Blueprint $table) {
```

```
        $table->id();
        $table->string('original_url');
        $table->string('shortened_url')->unique();
        $table->timestamps();
    });
}
```

Esegui la migrazione per creare la tabella:

```
php artisan migrate
```

Crea un modello per interagire con la tabella `urls`. Esegui il comando:

```
php artisan make:model Url
```

Apri il file `app/Models/Url.php` e aggiungi il seguente codice per consentire l'assegnazione di massa dei campi `original_url` e `shortened_url`:

```
protected $fillable = ['original_url', 'shortened_url'];
```

Creiamo un controller per gestire la logica del nostro URL shortener. Esegui il comando:

```
php artisan make:controller UrlController
```

Apri `app/Http/Controllers/UrlController.php` e aggiungi i metodi per creare e risolvere gli URL accorciati:

```
use App\Models\Url;
use Illuminate\Http\Request;
use Illuminate\Support\Str;

class UrlController extends Controller
{
    public function store(Request $request)
    {
```

```

$request->validate([
    'original_url' => 'required|url'
]);

$shortened_url = Str::random(6);
Url::create([
    'original_url' => $request->original_url,
    'shortened_url' => $shortened_url
]);

return response()->json(['shortened_url' => url($shortened_url)]);
}

public function show($shortened_url)
{
    $url = Url::where('shortened_url', $shortened_url)->firstOrFail();

    return redirect($url->original_url);
}
}

```

Definiamo le rotte per creare e risolvere gli URL accorciati. Apri `routes/web.php` e aggiungi le seguenti rotte:

```

use App\Http\Controllers\UrlController;

Route::post('/shorten', [UrlController::class, 'store']);
Route::get('/{shortened_url}', [UrlController::class, 'show']);

```

Per semplicità, possiamo creare una semplice vista per inserire l'URL originale e ottenere l'URL accorciato. Apri `resources/views/welcome.blade.php` e aggiungi il seguente codice:

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>URL Shortener</title>
    <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/tailwindcss/2.2.19/tailwind.min
.css">
</head>
<body class="bg-gray-100">
    <div class="container mx-auto p-4">
        <h1 class="text-2xl font-bold mb-4">URL Shortener</h1>

```

```

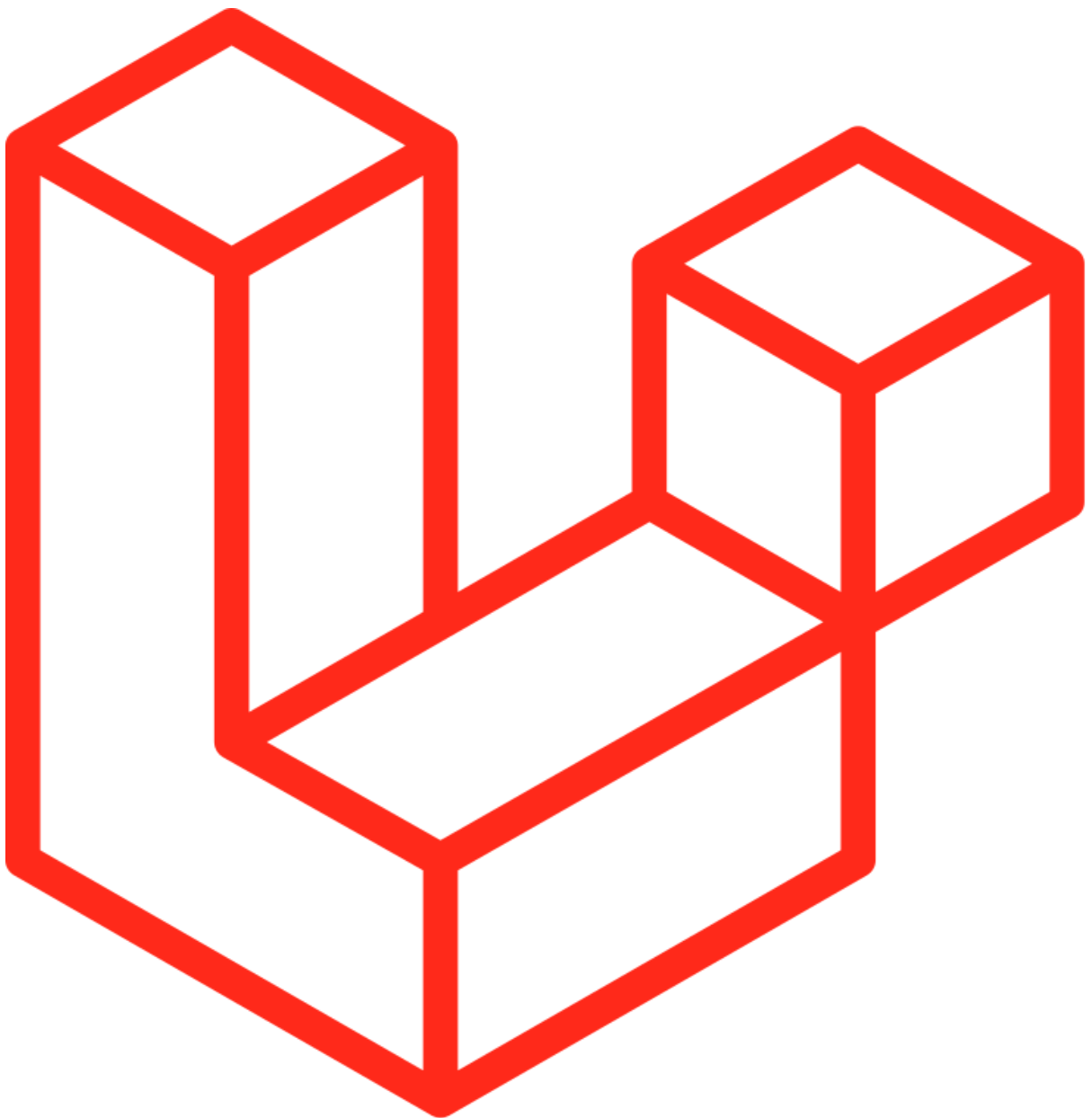
        <form id="url-form">
            <input type="text" id="original_url" class="border p-2 w-full"
placeholder="Enter your URL">
            <button type="submit" class="bg-blue-500 text-white p-2 mt-
2">Shorten</button>
        </form>
        <div id="result" class="mt-4"></div>
    </div>
    <script>
        document.getElementById('url-form').addEventListener('submit', async
(e) => {
            e.preventDefault();
            const originalUrl =
document.getElementById('original_url').value;
            const response = await fetch('/shorten', {
                method: 'POST',
                headers: {
                    'Content-Type': 'application/json',
                    'X-CSRF-TOKEN': '{{ csrf_token() }}'
                },
                body: JSON.stringify({ original_url: originalUrl })
            });
            const data = await response.json();
            document.getElementById('result').innerHTML = `<p>Shortened URL:
<a href="${data.shortened_url}" target="_blank">${data.shortened_url}</a>
</p>`;
        });
    </script>
</body>
</html>

```

Conclusione

Abbiamo creato un semplice URL shortener utilizzando Laravel. Questo progetto può essere ulteriormente esteso aggiungendo funzionalità come il tracking delle visite, la scadenza degli URL accorciati e l'autenticazione degli utenti. Laravel, con la sua potente architettura e gli strumenti integrati, rende lo sviluppo di applicazioni web complesso ma allo stesso tempo gratificante.

Applicazioni Correlate

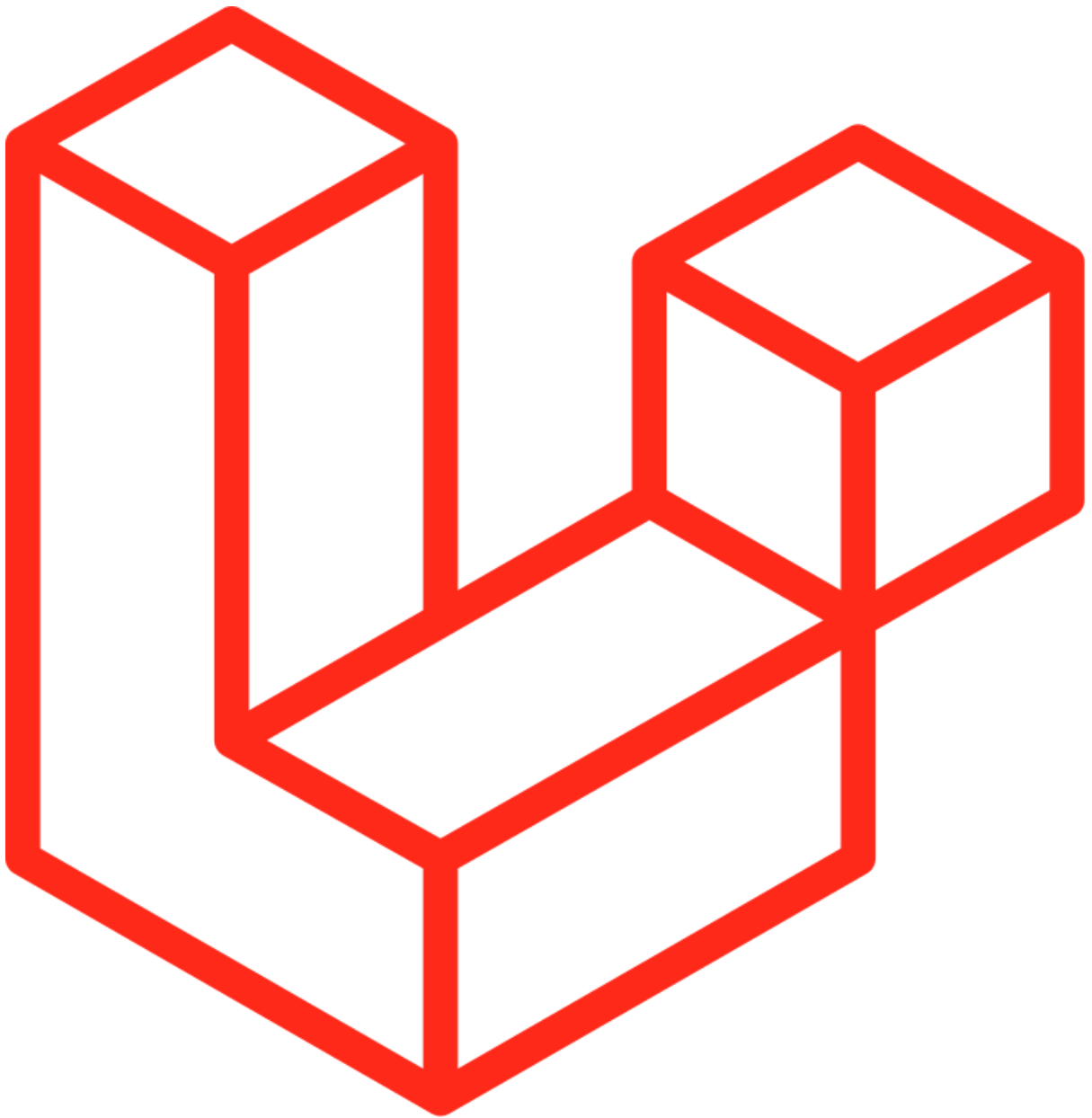


•

Laravel Placeholder Image

Applicazione in Laravel per creare immagini segnaposto.

Docker Docker Compose Composer PHP Laravel JavaScript

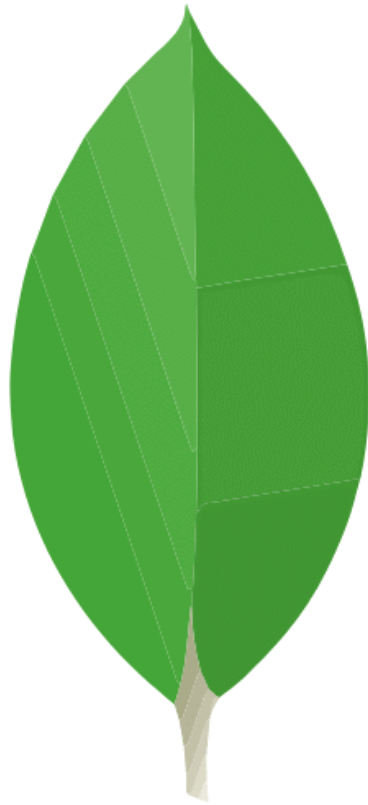


•

Laravel Shopping Cart

Applicazione che fa parte del progetto Laravel E-commerce e implementa la gestione del carrello in Laravel.

Docker Docker Compose Composer PHP Laravel



-

PHP MongoDB App

Applicazione basata su MongoDB con il driver PHP ufficiale.
DockerDocker ComposeComposerPHPMongoDB