Francesco Lettera

LARAVEL

THE PHP FRAMEWORK

UNA GUIDA SEMPLICE PER REALIZZARE APPLICAZIONI DI QUALITÀ

Laravel 5 in pratica

Una guida semplice per realizzare applicazioni di qualità

Francesco Lettera

Questo libro è in vendita presso http://leanpub.com/laravel5inpratica

Questa versione è stata pubblicata il 2018-11-02

Indice

Autenticazione bella e pronta

Laravel fornisce, out of the box, un sistema di autenticazione corredato di registrazione. Vediamo come implementarlo velocemente.

Autenticazione nella versione 5.2

Nella versione 5.2 di Laravel l'autenticazione viene trattata diversamente. In appendice troverai quello cerchi, altrimenti continua pure la lettura di questo capitolo.

In *database* troveremo la cartella *migrations*. A cosa serve?

Migration

Per creare le tabelle del nostro database, Laravel utilizza il suo migration system. Questo sistema permette la gestione completa delle tabelle. Ma non entriamo nei dettagli. Per il progetto è necessario creare due tabelle

In *database/migrations* troveremo due file:

```
- 2014_10_12_000000_create_users_table.php
- 2014_10_12_100000_create_password_resets_table.php
```

Apriamo il primo file. All'interno del metodo `up()` saranno presenti tutte le modifiche necessarie alla tabella. Il metodo `down` ci permetterà di effettuare un `rollback` della migration: cioè nel caso della tabella `users`, questa sarà cancellata.

Il secondo file, invece, creerà la tabella *password_resets*. Il funzionamento è simile al file precedente. Ma come fare per creare materialmente le due tabelle? Il primo step è configurare i dati d'accesso al db. In base al setup effettuato (MAMP, WAMP, Homestead, ecc) creiamo un db (es. laravel). Subito dopo apriamo il file .env presente nella root della nostra applicazione e indichiamo le credenziali d'accesso al db nelle opportune variabili.

```
// .env
...
DB_HOST=localhost
DB_DATABASE=laravel
DB_USERNAME=root
DB_PASSWORD=root
...
```

Da riga di comando (posizioniamoci sempre nella root della nostra applicazione) scriviamo:

```
php artisan migrate
```

E voilà, il nostro database avrà dunque la tabella *users* e *password_resets* con i suoi campi. Siamo pronti per testare l'autenticazione fornita di default da Laravel.

Cos'è .env?

Bella domanda. Per adesso è sufficiente sapere che si avvicina molto ad un file di configurazione.

Nel corso del libro approfondiremo l'argomento.

Registrazione, login e logout

Tocca al file di route. Apriamolo e indichiamo le "strade" per l'autenticazione:

```
// app/Http/routes.php

// home
Route::get('/home', function(){
    return view('home.main');
});

// Autenticazione
Route::get('auth/login', 'Auth\AuthController@getLogin');
Route::post('auth/login', 'Auth\AuthController@postLogin');
Route::get('auth/logout', 'Auth\AuthController@getLogout');

// Registrazione
Route::get('auth/register', 'Auth\AuthController@getRegister');
Route::post('auth/register', 'Auth\AuthController@postRegister');
```

Benissimo, mancano però le view. Siamo liberi di crearle come vogliamo purchè siano rispettati i campi di input, naturalmente. Ecco un suggerimento (preso dalla documentazione ufficiale). Il percorso dove posizionare il file è *resources/views/auth/login.blade.php* :

Questo è il form di autenticazione:

```
<form method="POST" action="/auth/login">
    {!! csrf_field() !!}

    <div>
        Email
        <input type="email" name="email" value="{{ old('email') }}">
    </div>

    <div>
        Password
        <input type="password" name="password" id="password">
    </div>

    <div>
        <input type="checkbox" name="remember"> Remember Me
    </div>

    <div>
        <button type="submit">Login</button>
    </div>
</form>
```

Questo, invece, quello di registrazione (percorso *resources/views/auth/register.blade.php*):

```
<form method="POST" action="/auth/register">
    {!! csrf_field() !!}

    <div>
        Name
        <input type="text" name="name" value="{{ old('name') }}">
    </div>

    <div>
        Email
        <input type="email" name="email" value="{{ old('email') }}">
    </div>

```

```
14      <div>
15          Password
16          <input type="password" name="password">
17      </div>
18
19      <div>
20          Confirm Password
21          <input type="password" name="password_confirmation">
22      </div>
23
24      <div>
25          <button type="submit">Register</button>
26      </div>
27  </form>
```

Infine l'homepage verso la quale si sarà rediretti dopo il login o la registrazione:

```
1  <!-- resources/views/home/main.blade.php -->
2
3  <h1>Homepage</h1>
4
5  <p>Benvenuto!</p>
```

Andiamo su

```
1  http://localhost:8888/auth/register
```

e registriamo l'utente (Sì, lo so, non è formattato. Ma nei capitoli successivi renderemo tutto più piacevole). Subito dopo risulteremo loggati nella nostra applicazione (e saremo, quindi, redirezionati verso "/home"). Il db sarà popolato dei dati della nostra registrazione. Se effettuiamo il logout:

http://localhost:8888/auth/logout

possiamo testare il login:

```
1  http://localhost:8888/auth/login
```

Insomma un ottimo sistema di autenticazione base. Nei capitoli successivi vedremo come customizzare l'autenticazione e come presentarla con tutti i crismi.