

Piccolo manuale **Laravel (7.x)**

Questo manuale vi aiuterà a rinfrescarvi la memoria nei momenti di buio più totale.

1. Prerequisiti

IMPORTANTE!!!

Non Utilizzate WAMP, XAMP, MAMP, QUACK o qualsiasi altro, munitevi di una Linux Box e assicuratevi di aver installato i seguenti pacchetti/software:

- Git
- Zip
- PHP > 7.2.5
- Composer
- npm

Git e Zip sono 2 comandi che generalmente vengono installati con qualsiasi ambiente, per maggiori dettagli sui requisiti visitate il link: <https://laravel.com/docs/7.x> poi fate click sulla voce di menù "Documentation" ed infine su # Installation > Server Requirements (che trovate poco più sopra del centro della pagina).

ATTENZIONE Nel caso non aveste PHP 7.2 installato ma aveste PHP 7.1 non ci sono particolari problemi, la differenza è che verrebbe istanziato un progetto Laravel all'ultima versione precedente (5.x) che per quanto vedremo noi, non ha differenze importanti dalla versione 7 tranne nella parte di gestione dell'autenticazione.

Installare Composer

Composer è un gestore di pacchetti per PHP e ci aiuta a gestire le librerie e per installarlo, la Documentazione di Laravel fornisce le indicazioni per tutti i S.O.

Per installare composer su ubuntu, usare il comando:

```
sudo apt-get install composer
```

Seguire la guida: <https://laravel.com/docs/> (e poi proseguite su Installation), potete trovare la sua documentazione completa per cli su: <https://getcomposer.org/doc/03-cli.md>

I comandi principali sono:

```
composer require <nomepackage> [vincolo][versione]
```

Aggiunge nuove dipendenze aggiungendo librerie al file composer.json

```
Es: composer require laravel/laravel ^6.0
```

```
x.y.z    -> Esattamente la versione x.y.z
```

```
>=x.y.z  -> Versione maggiore o uguale di x.y.z
```

```
^x.y.z   -> Versione vicina a x.y.z (limite alla major version)
```

```
~x.y.z   -> Versione vicina a x.y.z (limite alla release)
```

composer update

Aggiorna tutte le librerie all'ultima versione disponibile compatibile con quanto indicato in composer.json e al termine crea/aggiorna composer.lock (che contiene le versioni precise di quanto installato).

composer install

Installa le versioni specificate in composer.lock (che quindi deve già essere creato con un primo composer update).

composer clear-cache

Pulisce la cache di composer.

2. Getting Started!

Se durante la creazione di un nuovo progetto ottenete un errore, controllate nella sezione 5 (Troubleshooting) se è elencato il vostro caso

Creazione di un progetto Laravel (7.x)

1. Creare un nuovo progetto usando uno dei seguenti comandi a seconda se si vuole o meno voler specificare una versione precisa di Laravel:

- a. `composer create-project laravel/laravel --prefer-dist <nome_progetto>`
- b. `composer create-project laravel/laravel=7.0 --prefer-dist <nome_progetto>`

2. Configurare il progetto:

- a. `cd <nome_progetto>`
- b. `cp .env.example .env`
- c. `php artisan key:generate`
- d. `php artisan config:cache`

3. (Opzionale) Se il vostro progetto prevede l'utilizzo di un Database creare un nuovo database con MySQL:

```
CREATE DATABASE <dbname>;  
GRANT ALL PRIVILEGES ON <dbname>.* TO 'username'@'localhost'  
IDENTIFIED BY 'password';
```

Configurare il file `.env` del nuovo progetto con il driver e i parametri di connessione al database

4. (Opzionale MA DA FARE COME PRIMA COSA SE RICHIESTO) Se il vostro progetto prevede diversi utenti eseguite i seguenti passaggi:

Per laravel 7.x:

- a. `composer require laravel/ui --dev`
- b. `php artisan ui vue --auth`
- c. `npm install`
- d. `npm run dev`

Maggiori dettagli: <https://laravel.com/docs/7.x/authentication#authentication-quickstart>

Se avete bisogno di una gestione degli utenti più evoluta (per esempio con i ruoli utente) allora vi consiglio di seguire le indicazioni presenti in [questa discussione su Stack Over Flow](#)

5. Eseguire il server locale con il seguente comando:

`php artisan serve`

A questo punto visitando il link <http://127.0.0.1:8000>, troverete la pagina predefinita “welcome” del vostro progetto

3. Convenzioni sui nomi

Tabelle: Nome in **minuscolo**, al **plurale** con forma “snake” (_ per separare le parole) es. users, tags, ...

Colonne: Nome in **minuscolo** al **singolare** con forma “snake” (_ per separare le parole) es. first_name, last_name, birth_date...

Modelli: Al **singolare** con forma “cammellizzata” (prima lettera in maiuscolo per separare le parole) es. User, Tag, ...

Controllers: Al **singolare** con forma “cammellizzata” (prima lettera in maiuscolo per separare le parole) es. UserController, TagController, HelloController ...

4. Comandi base

Se durante l'esecuzione dei comandi ottenere un errore, controllate nella sezione 4 (Troubleshooting) se è elencato il vostro caso

Generatori

```
// Controller
php artisan make:controller <nome_controller_singolare>Controller

// Controller completo di tutte le azioni
php artisan make:controller <nome_controller_singolare>Controller
--resource

// Model
php artisan make:model <nome_model_singolare>

// Model (con migrazione)
php artisan make:model <nome_model_singolare> -m

// Model (con migrazione e controller)
php artisan make:model <nome_model_singolare> -m -c --resource
```

Migrazioni

```
// Creare una migrazione
php artisan make:migration create_<nome_tabella>_table

php artisan make:migration <sintesi_modifica>_to_<nome_tabella>_table

// Eseguire le migrazioni
php artisan migrate

// Eseguire il rollback dell'ultima migrazione
php artisan migrate:rollback [--step=n]

// Eseguire il reset del database, cancella tutte le tabelle e riesegue
le migrazioni
php artisan migrate:fresh
```

Seeder

```
// Creare un seeder
php artisan make:seeder <nome_model_plurale>TableSeeder

// Eseguire tutti i seed
```

```
php artisan db:seed
```

```
// Eseguire il seed solo di una specifica classe
```

```
php artisan db:seed --class=<nome_model_plurale>TableSeeder
```

5. Troubleshooting

Errori durante la creazione del progetto su Ubuntu

Installate i seguenti pacchetti con i comandi:

```
sudo apt-get install php-mbstring
sudo apt-get install php-xml
```

Errori durante la creazione del progetto

Può capitare che durante la creazione del progetto il comando restituisca un errore simile a:

```
"Script php artisan clear-compiled handling the pre-update-cmd event
returned with error code 255"
```

Per risolvere questo errore cancellate la dir del progetto ed eseguite il seguente comando per cancellare la cache di composer:

```
composer clear-cache
```

A questo punto ripartite dal punto 1a o 1b.

Errori nella lettura del config

Se ricevete degli errori che su delle configurazioni che non rispettano quello che avete inserito nel file .env allora dovete eseguire la pulizia della cache del file di config:

```
php artisan config:clear
```

Il codice JS o CSS non esegue quello che volete o non trova funzioni che avete appena scritto

Eseguire la pulizia della cache delle view:

```
php artisan view:clear
```

Errori che sembrano venire da codice "vecchio" che avete rimosso

Eseguire la pulizia della cache:

```
php artisan cache:clear
```

Errori durante l'esecuzione delle migrazioni

Può capitare di fare un po' di casini con il database e le migrazioni, ad esempio dover cambiare il nome del database o cambiare l'ordine delle migrazioni o voler inserire una migrazione fra due già esistenti.

Laravel ha un sistema di caching per migliorare le performance che potrebbe mettere i bastoni fra le ruote...

1. Cancellare i file dentro alla cartella `bootstrap/cache/`
2. Eseguire il reset delle migrazioni: **php artisan migrate:reset**
3. Se necessario cancellare tutte le tabelle e il database
4. Ricreare le migrazioni da capo nell'ordine corretto con il comando: **php artisan make:migration...**
5. Eseguire il comando: **composer dump-autoload**
6. Eseguire di nuovo le migrazioni: **php artisan migrate**

Errore SQL 42000 durante le migrazioni

Durante l'esecuzione delle migrazioni (generalmente alla prima) appare un errore riguardante una *"Key too long; max length is ..."*

Per risolverlo editate il file `AppServiceProvider.php` e aggiungete in testa:

```
use Illuminate\Support\Facades\Schema;
```

e nella funzione `boot()` inserite la seguente riga di codice:

```
Schema::defaultStringLength(191);
```

A questo punto cancellate le tabelle e rieseguite le migrazioni