

Vroomeco

Installation

- installer php (v8)
- installer [Composer](#)
- vérifier que `extension=fileinfo` dans `C:\tools\php82\php.ini` (si windows) est bien décommenté (enlever le ;)
- pareil pour `extension=pdo_mysql` (si vous êtes sur Ubuntu, à s'assurer que le driver `php_mysql` est bien installé)
- même manipulation pour `extension=zip`.
- Lancer le script d'installation dans le dossier vroomeco:
(NOTE: essayez de le lancer à chaque pull d'ailleurs)

A faire lors d'un clone github :

```
composer install          | Installe tout les packages laravel  
yarn                      | Installe tout les packages nodes
```

Crée le `.env` a partire de `.env.exemple`

```
DB_CONNECTION=mysql  
DB_HOST=127.0.0.1  
DB_PORT=3306  
DB_DATABASE=laravel (à modifier)  
DB_USERNAME=root  
DB_PASSWORD=
```

Puis faire la commande suivante :

```
php artisan migrate      | Crée le env et migres tout sur la Base de Donner  
php artisan key:generate | génère une clé d'application  
php artisan serve      | lance le server laravel  
yarn dev               | lancer react js (à lancer dans un terminal à part)
```

En production : `yarn build`

En développement : `yarn dev` (permet de charger a chaque modification)

route vers les dossier utiles

Name	Route	utilisation
React JS	<code>/ressource/js</code>	
route front	<code>/route/web.php</code>	
route back	<code>/route/api.php</code>	
route Controller	<code>/app/Http/Controllers</code>	A ne pas toucher sauf si vous faite le back

Route

Route	Description	Api
<code>/</code>	route vers la page d'accueil	✗
<code>/home</code>	route vers la page d'accueil après authentification	✗
<code>/login</code>	route vers la page de login	✗
<code>/register</code>	route vers la page de register	✗
<code>/logout</code>	route vers la page de deconnexion	✗
<code>/traject</code>	route vers la page traject	✗

Donnée par default:

faire la commande :

```
php artisan db:seed --class=UsersTableSeeder
```

vous aurez ensuite ces information la de remplie pour l'utilisateur de test:

```

DB::table('users')->insert([
    'name' => 'John Doe',
    'username' => 'johndoe',
    'email' => 'johndoe@example.com',
    'phone_number' => '123456789',
    'password' => Hash::make('password'),
    'biographie' => 'Lorem ipsum dolor sit amet, consectetur adipiscing
elit.',
    'created_at' => now(),
    'updated_at' => now(),
]);

```

Diagram de la bdd

```

---
title: Diagramme de la Base de Donnée
---
erDiagram
    USERS {
        Id INT
        Name STRING
        Username STRING
        Email STRING
        Email_verified_at DATETIME
        Phone_number STRING
        Password STRING
        Created_at DATETIME
        Updated_at DATETIME
        token STRING
        Biographie TINYTEXT
        Roles STRING
    }

    TRAJECT {
        Traject_ID INT
        uuid INT
        Created_By INT
        Vehicule INT
        Ville_depart STRING
        Ville_arriver STRING
        Date_traject DATE
        Estimated_time TIME
        Distance INT
    }

    USERS_TRAJECTS {
        User_traject_ID INT
        User_Id INT
    }

```

```

    Traject_uuid INT
    Confirmed_at DATE
}

CAR {
    Id INT
    User_id INT
    Marque STRING
    Model STRING
    Type STRING
    Colors STRING
    Annee DATE
    Immatriculation STRING
}

PERSONAL_ACCESS_TOKENS {
    id INT
    tokenable_id INT
    tokenable_type STRING
    name STRING
    token STRING
    abilities TEXT
    last_used_at TIMESTAMP
    expires_at TIMESTAMP
    created_at TIMESTAMP
    updated_at TIMESTAMP
}

USERS ||--o{ TRAJECT : "Created_By (User_Id)"
TRAJECT ||--o{ USERS_TRAJECTS : "Traject_uuid"
USERS ||--o{ USERS_TRAJECTS : "User_Id"
USERS ||--o{ CAR : "User_id"
PERSONAL_ACCESS_TOKENS || --o{ USERS : token

```