-laravel 基本コマンド-

composer create-project laravel/laravel=5.8 sample_app --prefer-dist

雛形作成

(本来は
laravel new (app 名);
後でパス通す。)

マイグレーションファイル作成

サーバー起動コマンド
php artisan serve

基本コマンド
php artisan make:migration (テーブル名)_table

database/migration ファイルに作成される。

php artisan make:migration create_(テーブル名)_table —create=(新規 table 名)

—create=(新規 table 名)は新規テーブル名につけるオプション
(オプション内容をちゃんとみる。)

php artisan migrate;
設計書にしたがってファイル作成.
(作成後は use -> show table で確認)

ログイン機能一覧作成
php artisan make:auth

-ロールバック用コマンド-
php artisan migrate:rollback;

オプション
—step=(戻りたいステップ数)
例 3 場面分戻りたい時
php artisan migrate:rollback —step=3;

フォルダ/ワークスペース全体

- Ctrl + Shift + F　フォルダ/ワークスペース全体を検索
- 特定のフォルダ配下を検索したい場合はファイルエクスプローラでフォルダを右クリックして「フォルダー内を検索」

- Ctrl + Shift + H　フォルダ/ワークスペース全体を置換
- F4 / Shift + F4　次の/前の検索結果にジャンプ
- shift+commond+↓ 全選択

検索関係

- Ctrl + F　エディタ内の検索
- Ctrl + H　エディタ内の置換

ファイル名からファイルを開く	Quick Open	cmd + P	
コマンドの実行	Show All Commands	cmd + shift + P	
定義、参照を見る&ジャンプ	Peek References	shift + F12	ctrl + '
シンボルからファイルを開く	Open symbol by name	cmd + T	
ファイル内のシンボルに移動	Go to Symbol	cmd + shift + O	

-基礎知識まとめ-
migratetable 関係
-up メソッド-
新規テーブルの設計図
Schema クラスの専用関数 create を使う

-down メソッド-
ロールバック用設計図
Schema クラスの専用関数 dropIfExits を使って今あるテーブルを削除。

php artisan migrate 時に作成される
migrate テーブルは migrate ファイルの内容の確認になる。

php artisan migrate をした後は
down メソッドが機能するかを確認する為に
php artisan migrate:rollback を使う。

larabel はデフォルトで create_users,create_password ができている。

-命名規則関係-

table 名は複数形

model 名は単数形(というより model 名のみ単数形かもしれない)

習慣にするべき事

新規 table 作成後は

-use (対象の table 名);で対象 table を選択。

例
use app_sample

-show tables;で table 一覧を見る。

laravel だとオリアプで作成した function.php の様な関数を纏めたファイルは trait で纏められている。

trait 内の関数を使いたい場合は宣言名+use で別ファイルで使える。

trait･・・ php5.4 以降で実装された機能。
class でインスタンス化して使いたいほどではないもの(使いたくてもインスタンス化は出来ない。)
や関数を纏めておく関数。

seeder･・・ ダミーデータ作成の機能
database/seeder
php artisan make:seeder

例えば user テーブルにダミーデータを作りたい場合は
php artisan make:seeder UsersTableSeeder
みたいにする。

ざっくりとした流れ
php artisan make:seeder コマンドを打った後に
DataBases/Seeder の中に
Seeder クラスを継承した seeder 関係のクラスと run メソッドが用意されたファイルが生成される。

その中で事前に用意されている DB クラス内にある table 関数を使ってダミーデータを入れたいテーブルを取得する。
(DB クラスは用意されているけど、use を使ってのパス通しをしていないの通す。)

DB クラスのディレクトリは

use Illuminate\Support\Facades\DB;

Seeder ファイルをつくった時はこれもやっとく。
use Carbon\Carbon;
createtime スタンプを使う際等につかう関数を使う為にこれを使う。

書き終わったら
php artisan db:seed —class=(実行対象の seeder ファイル);

php artisan db:seed —class=UserTableSeeder;

macOS を JIS キーボードで使っている場合、バックスラッシュ (\ ) の入力は
- option キーを押しながら、¥キーを押す。(Mac mini 等で Windows 用のキーボードを接続している場合は Alt キー)

: コロン

その後 insert 関数(SQL の insert と同じ)を使って連想配列の形でキー側にカラム名。バリュー側に内容を書く。

* password カラムの内容には bcrypt 関数を使ってハッシュ化させるのを忘れずに

FW は基本エスケープが挟まれる。

controller 関係
命名規則あり(アッパーキャメルケース)
php artisan make:controller (作りたいコントローラー名)

例
php artisan make:controller DrillsController

メソッド名に new を使う事は php7 以前は❌

多言語化対応用の関数
翻訳用のフォルダを別に用意する事でブラウザ側の設定をいじるだけで色々な言語に表示内容を変更できる。
{{__(‘ ’)}}

翻訳用フォルダの設定
(php でも json でも可.)
{
　英語:日本語 で分ける。

```
    "Drill Register": "練習登録",
    "title": "タイトル",
    "Category": "カテゴリ-",
    "problem": "問題"
}
```

設定後 config\app.php で
local:ja;
デフォルト言語を設定。

‘fallback_locale’ => ja
選択した言語の翻訳ファイルがない場合のやつ。

v

old()
入力情報を保持しておく関数

html5 から使える様になった html 上で入力情報の確認をする属性。

autocomplete="email"

ヘルパーメソッド laravel で用意されている便利メソッド
例
action=“{{route(‘drills.new’}}”
“{{route(‘’}}”
view ファイルで用意されているヘルパーメソッド
これが付加されているボタン等を押せば route で設定されている->name()を元に route を走らせることができる。