

use namespace 関係

namespace・・・参照対象に入れたい際に使うパス通し用のやつ。

use・・・参照対象になったファイル内にある関数やクラスなどを使う為に使うもの。

ついでに参照元ファイルに namespace が一見無さそうに見えた場合はクラスの継承元で namespace が使われているかも。

ファイル階層が一つしかない場合は、trait を使っている場合が多い気がする。

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。

よく使う use 関係

-デフォルトで作成されているもの-

* migration ファイル関係

=====

-use Illuminate\Support\Facades\Schema;-

make:migration を使ってマイグレーションファイルを作成した時に自動で付属される use 階層。

例

create()・・・DB内にテーブルを作成する関数。第一引数はテーブル名・第二引数以降はカラムやレコードを作成する為の関数を記入していく。

dropIfExists()・・・対象 table を削除し作り直す?関数。

make:rollback を使う際に使われる。

=====

-use Illuminate\Database\Schema\Blueprint;-

create テーブルを作る際などに使う関数の内の一部を使いたい時のもの。

例

bigIncrements()・・・主キー設定時に使う関数

timestamp()・・・table内にcreated_at,updated_atカラムを作成する関数。

rememberToken();

=====

* seeder ファイル関係

=====

-use Illuminate\Database\Seeder;-

=====

*controller 関係

=====

namespace App\Http\Controllers;

use Illuminate\Http\Request;

=====

まだよくわかってない。

-use Illuminate\Database\Migrations\Migration;-

-後からよく追加する物-

=====

* seeder ファイル関係

-use Carbon\Carbon;-

Carbon クラスやそれに関係するメソッドを使う為の use 階層

主に create_at や update_at を

now()

-use Illuminate\Support\Facades\DB;-

作成済みの DB 関係に直接アクセスしたりする関数や DB クラスを

使う為の use 階層

table() 実際に作成済みの table に直接アクセスする為の関数。

これに insert()を使ったりして data を直接入れたりする。

=====

-その他-

=====

-use Illuminate\Database\Schema\ColumnDefinition-

この namespace 元ファイルを参照している関数は dbal をインストールしないと
いけない。

composer require doctrine/dbal を使ってインストール

=====

cd /Applications/MAMP/htdocs/typeingPractice_app

php artisan serve

cd /Applications/MAMP/Library/bin/

./mysql -u root -p;

名前空間

php5.3 からできた仕組みでクラスのディレクトリと同じ様に、違う名前空間で同じ名前のクラスを定義できる様になる。

名前空間なしでは PHP の標準クラスと同じなまえのくらすはつukれないため、違う名前にするしかなく、既存のクラス名にプレフィックス(接頭辞)をつけ長くした物が多かった。

大人数で開発をする場合、同じクラス名をるけてしまい名前衝突してしまう場合がある為、コミュニケーションをとりながら開発をし、プレフィックス(接頭辞)をつけ名前を長くしていた。そのため、クラス名が無駄に長くなっていた。そのままの名前を使うことができる。

namespace . . . 使い回す予定ある関数などをもつクラスがあるファイルに使う。そのファイルの一つ前のフォルダ名から置いておく。

別のファイルでも同じ関数を使いたい場合に使う宣言方法.

例

大元の方の宣言方法

```
<?php
namespace Auth;
class Login
{
}
```

ファイル側の宣言

フルパスでの使い方

```
$login = new \Auth\Login();
```

あらかじめ宣言する方法

```
use Auth\Login as AuthLogin;
use Db\Login as DbLogin;
$login = new Login;
```

use RegistersUsers; <- 階層の記入が無く、ファイル名のみが記入されている場合 trast 関数を使う事が多い?

* trast 関数 . . . クラス定義するほどのものではない物をまとめておく。

return View::make('profile'); <-ファザード

return view('profile');<-ヘルパ関数

コマンド実行時に吐かれるエラーは重要度順に出てくる。