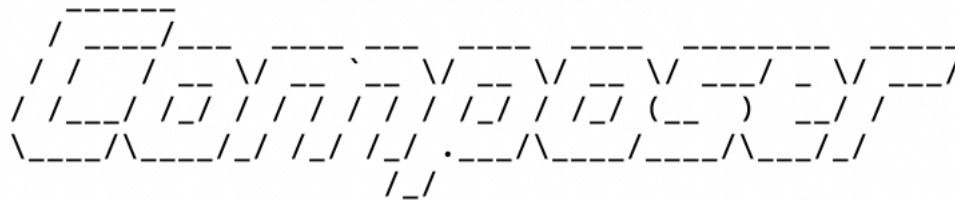


# Laravel

## composer



Composer version 2.3.7 2022-06-06 16:43:28

### Usage:

command [options] [arguments]

コマンドプロンプト or ターミナルで実行する。

### ①インストール

Windows の場合：

Composer のサイトからインストーラをダウンロード  
(<https://getcomposer.org/>)

mac の場合：

curl -sS <https://getcomposer.org/installer> | php を実行  
上記実行後、composer -v とし、左記のような表示が出ればインストール完了です。（先を確認）

### ②プロジェクト作成

composer create-project laravel/laravel プロジェクト名  
--prefer-dist を実行

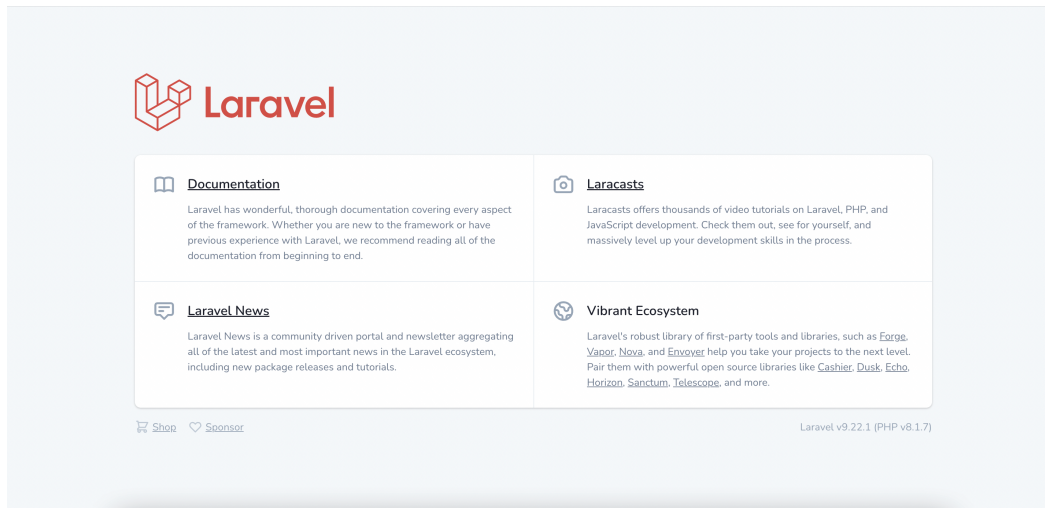
例) composer create-project laravel/laravel sample  
--prefer-dist

(MAMP or Xampp の htdocs 内で実行する事)

Composer とは PHP のライブラリやパッケージを管理するライブラリ依存管理ツールです。プロジェクトに必要なものを設定ファイルに追加しておくことで自動的にインストールしてくれる機能があります。

# Laravel

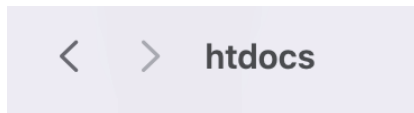
## 動作確認



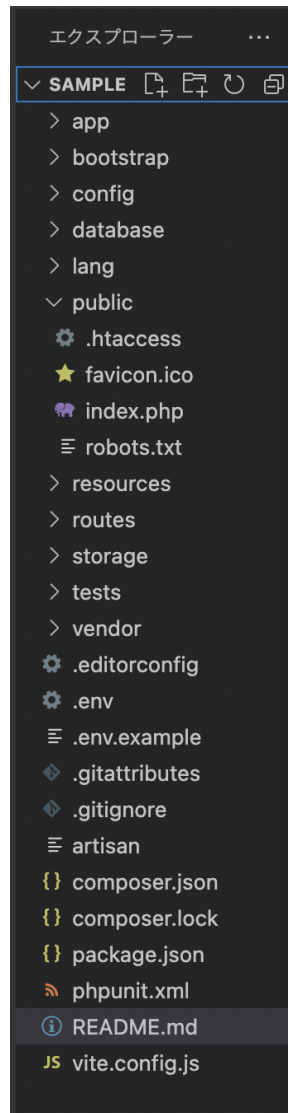
- ① `cd` プロジェクト名 (例) `cd sample` を実行し、プロジェクトディレクトリに移動する。
- ② MAMP or Xampp を起動する。
- ③ `php artisan serve` を実行する。
- ④ ブラウザで `http://127.0.0.1:8000` にアクセスし、左の画面が表示されるかを確認する。

# Laravel

## はじめに



sample

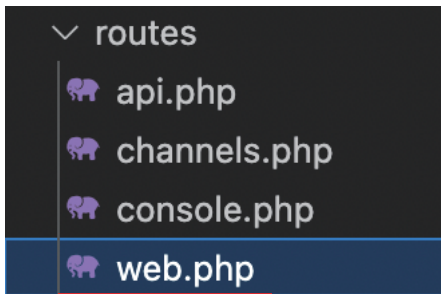


それでは Laravel で MVC を作成して処理の流れを把握していきましょう。

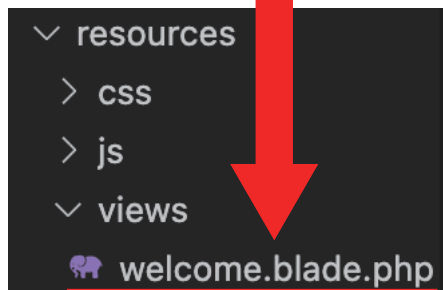
前ページで作成した Laravel プロジェクトをエディタで開きます。

# Laravel

## ルーティング



```
16 Route::get('/', function () {  
17     return view('welcome');  
18 });
```



Laravel でアクセスを制御するためにはルーティングという設定を行います。

WEB に関するルーティングは routes ディレクトリ配下の web.php を編集します。

デフォルトでは / にアクセス、つまり `http://127.0.0.1:8000` にアクセスすると、welcome ページに遷移するように設定されています。

# Laravel

## ルーティング 2

### ▼ SAMPLE

#### ▼ app

##### > Console

##### > Exceptions

#### ▼ Http

##### ▼ Controllers

🐘 Controller.php

🐘 PlayersController.php

```
routes > 🐘 web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\PlayersController;
5
6  /*
7  |-----
8  | Web Routes
9  |-----
10 |
11 | Here is where you can register web routes for your application. These
12 | routes are loaded by the RouteServiceProvider within a group which
13 | contains the "web" middleware group. Now create something great!
14 |
15 */
16
17 // Route::get('/', function () {
18 //     return view('welcome');
19 // });
20
21 Route::get('/', [PlayersController::class, 'index']);
22 Route::get('/index', [PlayersController::class, 'index']);
```

追加

ここでは直接ビューではなく、コントローラーを通す設定にしてみます。

まず、ここで使用するコントローラーを作成します。

ターミナルまたはコマンドプロンプトで ここまで作成したファイルに移動し、php artisan make:controller ○○

Controller 例 )php artisan make:controller

PlayersController を実行します。

Controller created successfully. と表示されれば成功です。

App>Http>Controller 配下に作成したコントローラーが作られます。

次に、routes ディレクトリ配下の web.php を編集し、/ もしくは /index にアクセスされた場合、PlayersController の index メソッドを呼び出すようにします。

# Laravel



## ルーティング 3

```
app > Http > Controllers > 🐘 PlayersController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class PlayersController extends Controller
8  {
9      public function index() {
10         echo 'success';
11     }
12 }
```

前項設定後、コントローラーの index メソッドに適当に echo するプログラムを記述します。

ブラウザの URL バーに `http://http://127.0.0.1:8000` と入力し、左記画面が表示されれば O K です。

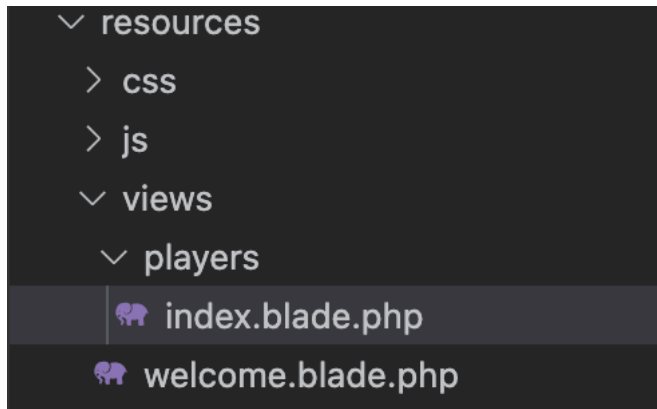
← → ↻ ⓘ 127.0.0.1:8000

 free pics  Gmail  YouTube 

success

# Laravel

## コントローラーとビュー



続いてコントローラーとビューを連携します。

ここからはコントローラーは処理、ビューは表示という役割を意識して作成していきます。

まずは resources/views 配下にビューを作成しますが、players/index.blade.php という構成とします（中身は後で記述）。

Laravel のビューでは blade というテンプレートエンジンを使用します。

# Laravel

## コントローラーとビュー 2

```
app > Http > Controllers > 🐘 PlayersController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  class PlayersController extends Controller
8  {
9      public function index() {
10         $params = [
11             'test' => 'これはテストです。',
12             'sample' => 'これはサンプルです。'
13         ];
14
15         return view('players.index', compact('params'));
16     }
17 }
18
```

ビューのパスを記述      ビューに連携するデータを記述  
「,」で区切り、「\$」は不要

続いてコントローラーを記述していきます。

ここではコントローラー内で定義した配列データをビューに連携するプログラムを作成します。



# Laravel

## コントローラーとビュー 3

```
resources > views > players > 🐘 index.blade.php
```

```
1  {{ $params['test'] }}<br>
2  {{ $params['sample'] }}
```



127.0.0.1:8000



free pics



Gmail



YouTube

これはテストです。

これはサンプルです。

続いてビューを記述していきます。

先ほどコントローラーから連携された配列データを左記のように記述するだけで画面に表示することができます。

Blade テンプレートの詳細についてはここでは触れませんので自分で使い方を調べてみましょう。（「Laravel Blade 使い方」で検索）

# Laravel

## モデル

```
app > Http > Controllers > 🐘 PlayersController.php
1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6  use App\Models\Player;
7
8  class PlayersController extends Controller
9  {
10     public function index() {
11         $players = Player::all();
12         return view('players.index', compact('players'));
13     }
14 }
```

続いてモデルを使用し、D B からデータを取得してみましょう。

ここでは例として SQL 基礎で使った worldcup のデータを使用します。

まず、php artisan make:model Player を実行し、Player.php を Models 配下に作成します。

次に、Controller を編集します。

左記のようにモデル記述を追加し、取得したデータをビューに連携します。

フレームワークでのモデルではクエリビルダという S Q L 発行するメソッドが既に用意されています。

記述例以外の様々なメソッドを活用できるよう Laravel ドキュメントを確認する癖をつけましょう。

# Laravel

## モデル 2

```
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=laravel
15 DB_USERNAME=root
16 DB_PASSWORD=
```



次に .env ファイルをデータベースの設定を左記のように変更・追記します。

```
11 DB_CONNECTION=mysql
12 DB_HOST=127.0.0.1
13 DB_PORT=3306
14 DB_DATABASE=worldcup
15 DB_USERNAME=root
16 DB_PASSWORD=root
17 DB_SOCKET=/Applications/MAMP/tmp/mysql/mysql.sock
```

# Laravel

## モデル 3

resources > views > players > index.blade.php > ...

```
1  <table>
2      <tr>
3          <th>ID</th>
4          <th>背番号</th>
5          <th>ポジション</th>
6          <th>所属</th>
7          <th>名前</th>
8          <th>誕生日</th>
9          <th>身長</th>
10         <th>体重</th>
11     </tr>
12     @foreach($players as $player)
13         <tr>
14             <th>{{ $player->id }}</th>
15             <th>{{ $player->uniform_num }}</th>
16             <th>{{ $player->position }}</th>
17             <th>{{ $player->name }}</th>
18             <th>{{ $player->club }}</th>
19             <th>{{ $player->birth }}</th>
20             <th>{{ $player->height }}</th>
21             <th>{{ $player->weight }}</th>
22         </tr>
23     @endforeach
24 </table>
```

最後に、ビューにモデルを使用し取得したデータを表示してみます。

ここでは配列データではなく、オブジェクトデータとして渡されているため、左記のようにループを使用します。

# Laravel

## モデル 4

ID	背番号	ポジション	所属	名前	誕生日	身長	体重
1	12	MF	ジュリオセザール	トロント (カナダ)	1979-09-03	186	79
2	3	MF	ジェフェルソン	ボタフォゴ	1983-01-23	188	80
3	22	MF	ビクトル	アトレチコ・ミネイロ (ブラジル)	1983-01-21	193	84
4	23	DF	マイコン	ローマ (イタリア)	1981-07-26	184	77
5	14	DF	マックスウェル	パリサンジェルマン (フランス)	1981-08-27	176	73
6	2	DF	アウベス	バルセロナ (スペイン)	1983-05-06	173	64
7	13	DF	ダンチ	Bミュンヘン (ドイツ)	1983-10-18	188	87
8	3	DF	チアゴシウバ	パリサンジェルマン (フランス)	1984-09-22	183	79
9	15	DF	エンリケ	ナポリ (イタリア)	1986-10-14	187	73
10	4	DF	ダビドルイス	チェルシー (イングランド)	1987-04-22	189	84
11	6	DF	マルセロ	Rマドリード (スペイン)	1988-05-12	172	73
12	5	MF	フェルナンジーニョ	マンチェスターC (イングランド)	1985-05-04	175	67
13	18	MF	エルナネス	インテルミラノ (イタリア)	1985-05-29	181	76
14	16	MF	ラミレス	チェルシー (イングランド)	1987-03-24	180	73
15	17	MF	ルイスグスタボ	ヴォルフスブルク (ドイツ)	1987-07-23	187	80
16	8	MF	パウリーニョ	トットナム (イングランド)	1988-07-25	182	71
17	19	MF	ビリアン	チェルシー (イングランド)	1988-08-09	174	72
18	11	MF	オスカル	チェルシー (イングランド)	1991-09-09	180	66
19	9	FW	フレジ	フルミネンセ (ブラジル)	1983-10-03	185	90
20	7	FW	フッキ	ゼニト (ロシア)	1986-07-25	180	85
21	21	FW	ジョー	アトレチコ・ミネイロ (ブラジル)	1987-03-20	189	78
22	10	FW	ネイマール	バルセロナ (スペイン)	1992-02-05	175	64
23	20	FW	ベルナルジ	シャフタル・ドネツク (クロアチア)	1992-09-08	163	60
24	12	GK	コロナ	クルスアスル (メキシコ)	1981-01-26	178	72
25	23	GK	タラベラ	トルーカ (メキシコ)	1982-09-18	188	85

ブラウザからアクセスすると D B のデータが表示できていることが確認できます。

このようにフレームワークでは既に用意されている機能を理解して活用する必要があります。

# Laravel

## おまけ

- 公式ドキュメント（基本はここを見て理解できるようにしましょう）
- <https://readouble.com/>
- Laravel でログインをサクッと作成してみましょう。
  - 検索キーワード例：「Laravel Jetstream」（Laravel8 以降）
- 以下を参考に Blade テンプレートでレイアウトを共通化してみましょう。
  - 検索キーワード例：「Laravel Blade」「Laravel レイアウト」
- GET や POST したデータをコントローラーで取得してみましょう。
  - 検索キーワード例：「Laravel GET 取得」「Laravel POST 取得」
- Laravel における配列やオブジェクトデータの確認方法を理解しましょう。
  - 検索キーワード例：「Laravel dd」「Laravel dump」