Laravel 超入門

自己紹介

滝澤 正大

Webアプリ開発.
iOS／Androidアプリ開発.

出身：群馬県

趣味：写真撮影

所属：デジタル・ヒュージ・テクノロジー

WEBアプリケーション
静的Webから動的Webへ

- WWWの誕生．静的ページの利用
- CGIの誕生．動的ページの利用

Laravel
特徴

- Webアプリケーションフレームワーク
- 人気
- セットアップが簡単
- MVCモデル
- 比較的学習コストが低い
- サーバサイドのFWながらVue.jsやBootstrapとの連携も可能

Laravel
特徴

- Webアプリケーションフレームワーク
- 人気
- セットアップが簡単
- MVCモデル
- 比較的学習コストが低い
- サーバサイドのFWながらVue.jsやBootstrapとの連携も可能

Webアプリケーションフレームワーク
Webアプリケーションフレームワークとは

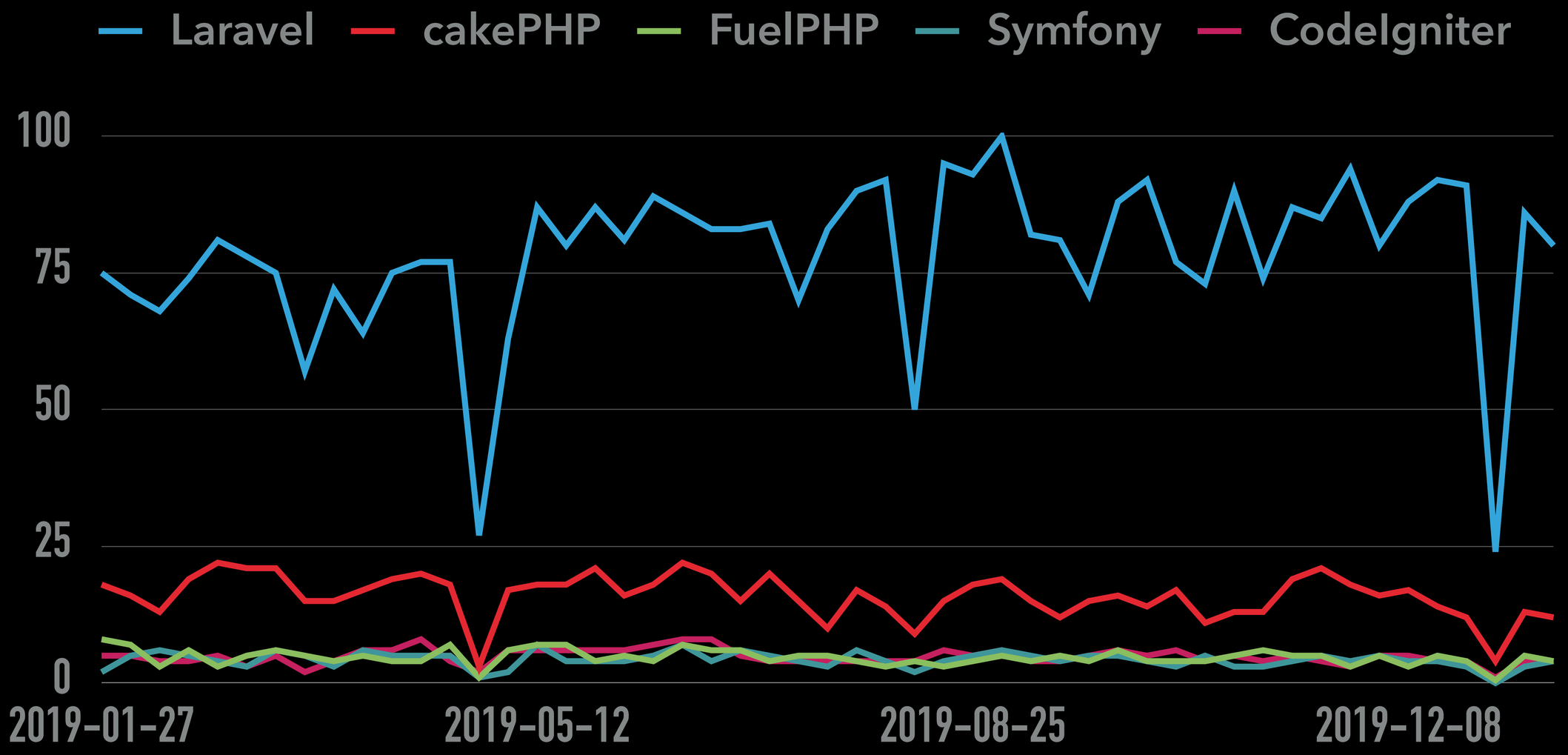
- Webサービス開発をサポートするフレームワーク
- サーバサイドで利用頻度の高い機能を簡単に実装
- コンポーネントの再利用性を高めてくれるものもある

Laravel
特徴

- Webアプリケーションフレームワーク
- 人気
- セットアップが簡単
- MVCモデル
- 比較的学習コストが低い
- サーバサイドのFWながらVue.jsやBootstrapとの連携も可能

Laravel

Google Trends



Laravel
特徴

- Webアプリケーションフレームワーク
- 人気
- セットアップが簡単
- MVCモデル
- 比較的学習コストが低い
- サーバサイドのFWながらVue.jsやBootstrapとの連携も可能

Laravel
セットアップ

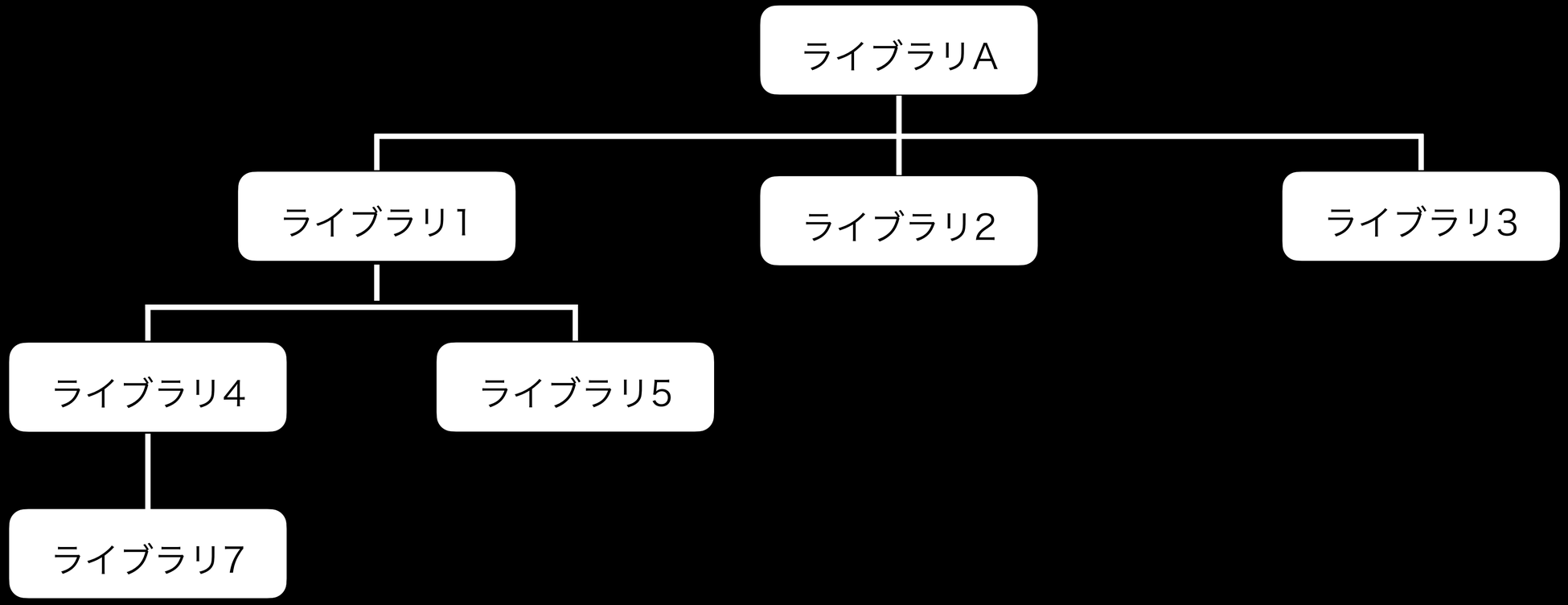
```
$ composer global require laravel/installer

$ laravel new laravel-project
```

Composer

役割

パッケージ管理システム



Laravel
セットアップ

```
$ composer global require laravel/installer

$ laravel new laravel-project
```

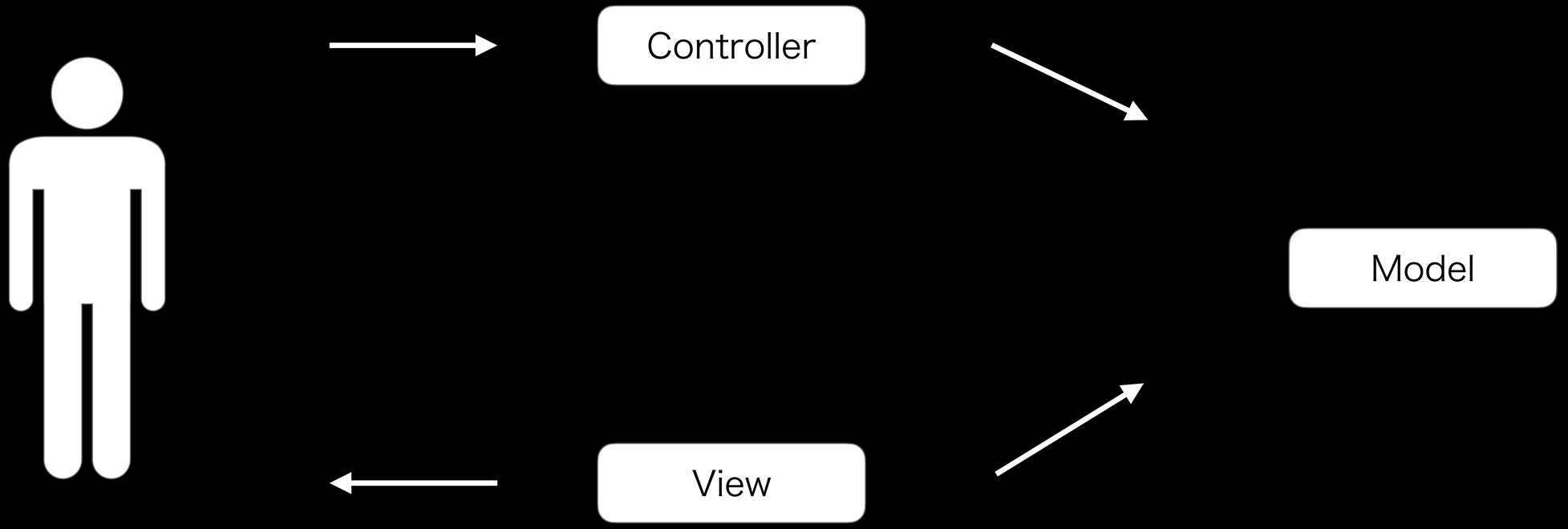
Laravel
特徴

- Webアプリケーションフレームワーク
- 人気
- セットアップが簡単
- MVCモデル
- 比較的学習コストが低い
- サーバサイドのFWながらVue.jsやBootstrapとの連携も可能

MVCモデル
MVCの歴史と特徴

- GUIをもつアプリケーションのアーキテクチャ
- 1970年年代．SmalltalkのGUIアプリケーションのために考案

MVCモデル
MVCの特徴と歴史

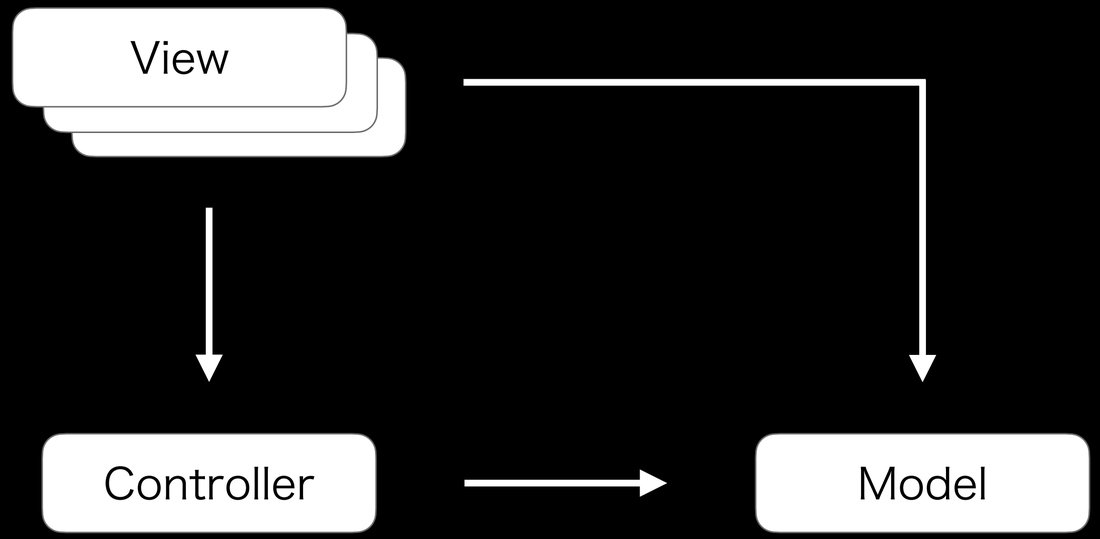


MVCモデル
MVCの歴史と特徴

- GUIをもつアプリケーションのアーキテクチャ
- 1970年年代．SmalltalkのGUIアプリケーションのために考案

MVCモデル
原初MVC

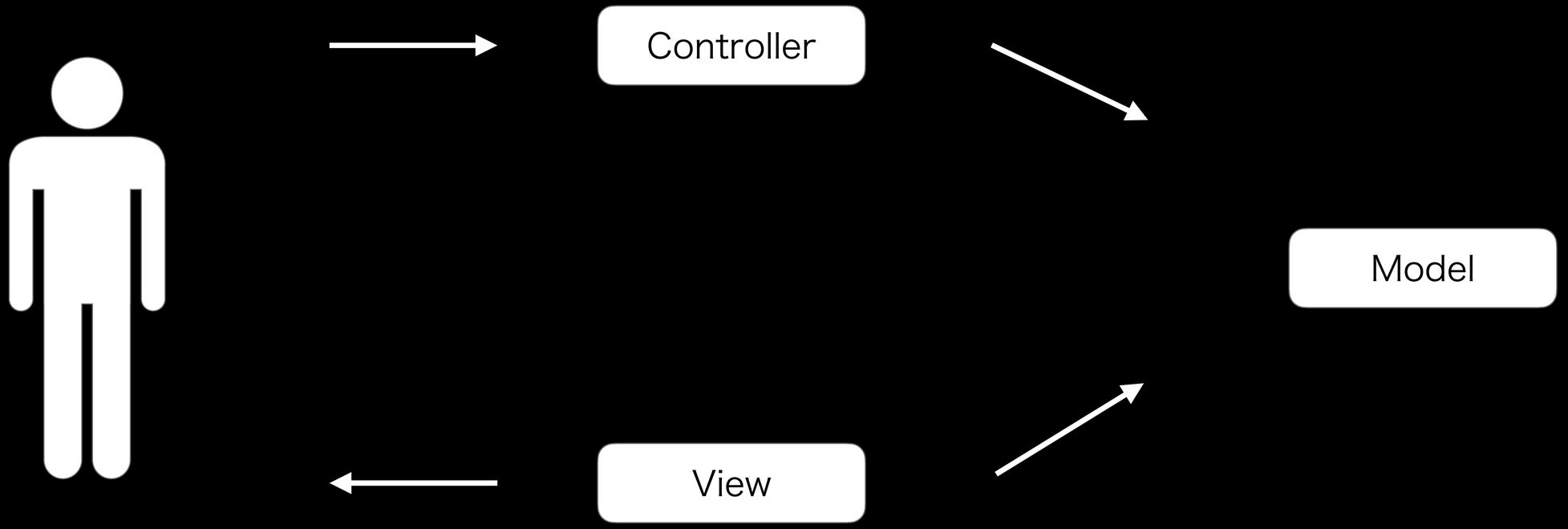
Composite，Observer，Strategyの３つのデザインパターンを軸に構成



MVCモデル
Laravel MVC

- 原初MVCのデータ同期は宣言的同期．Laravelは手続き的同期
- ViewやModelの再利用性を高めた構造

MVCモデル
MVCの特徴と歴史

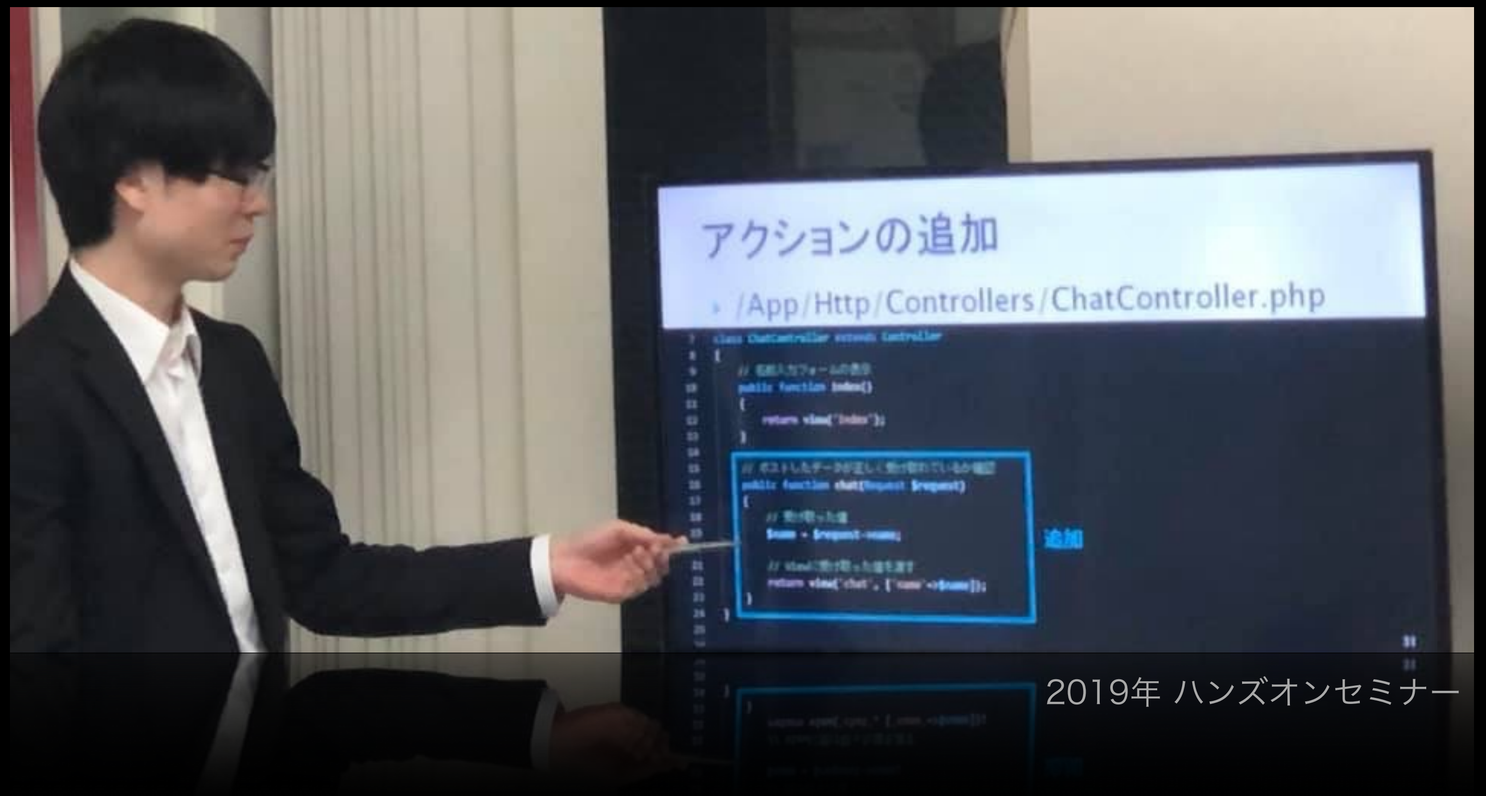


Laravel
特徴

- Webアプリケーションフレームワーク
- 人気
- セットアップが簡単
- MVCモデル
- 比較的学習コストが低い
- サーバサイドのFWながらVue.jsやBootstrapとの連携も可能

Laravel
低い学習コスト

最低限の操作で，それなりの機能ができる



Laravel
ルーティング

特定のアドレスと特定の処理を結びつける

Laravel
ルーティング

```
Route::get('foo', function () {
    return "Hello World!";
});
```

http://xxx/foo に対して呼び出される

Laravel
ルーティング

```php
Route::get('foo', function () {
    // 受け取ったデータをDBに反映
    // DBからデータを取得
    // レスポンス用のHTMLなどを生成
    return $response;
});
```

Laravel
Model-View-Controller

Model　　：DBと連携を行い，データ管理を行う

View　　：モデルのデータを埋め込んだ画面を生成する

Controller：モデルを用いてビューを生成する

Laravel
Modelの定義

```
class User extends Model {
    /** 生年月日から年齢を計算する */
    public function calcAge() {
        ・・・
    }
}
```

Laravel
Model-View-Controller

Model　　：DBと連携を行い，データ管理を行う

View　　：モデルのデータを埋め込んだ画面を生成する

Controller：モデルを用いてビューを生成する

Laravel
Viewの定義

```
<html>
    <body>
        @foreeach ($users as $user)
            <p>{{ $user->name }}</p>
        @endforeach
    </body>
</html>
```

Laravel
Model-View-Controller

Model　　：DBと連携を行い，データ管理を行う

View　　：モデルのデータを埋め込んだ画面を生成する

Controller：モデルを用いてビューを生成する

Laravel
Controllerの定義

```
class CustomController extends Controller {
    public function index(Request $request) {
        // モデルを用いてDBからデータを取得
        $users = User.all();
        // モデルデータをビューに渡してページを生成
        return view('customview', ['users'=>$users]);
    }
}
```

Laravel
ルーティングの変更

```
Route::get(‘foo’, ‘CustomController@index’);
```

Laravel
特徴

- Webアプリケーションフレームワーク
- 人気
- セットアップが簡単
- MVCモデル
- 比較的学習コストが低い
- サーバサイドのFWながらVue.jsやBootstrapとの連携も可能

Laravel
ルーティングの変更

```
$ composer require laravel/ui —dev

$ php artisan ui bootstrap

$ php artisan ui vue

$npm install && num run dev
```

Laravel
特徴

- Webアプリケーションフレームワーク
- 人気
- セットアップが簡単
- MVCモデル
- 比較的学習コストが低い
- サーバサイドのFWながらVue.jsやBootstrapとの連携も可能