Laravel 超入門

…

岡田州平

自己紹介

名前：岡田州平

経歴：2年間教育業界に携わる
　　エンジニアとして10か月

趣味：本を読むこと

直近の目標：一人で稼ぐ力を身につけること

所属：デジタル・ヒュージ・テクノロジー

事業：https:///hotservice.jp

DHT
Digital Huge Technology

今日の目標

- Webサイトの仕組みについて、ざっくりと理解する
- Laravelの構造について理解する
- フロントエンドエンジニアに求められるスキルを理解する

目次

Webサイトの仕組み

サーバーと通信

リクエストとレスポンス

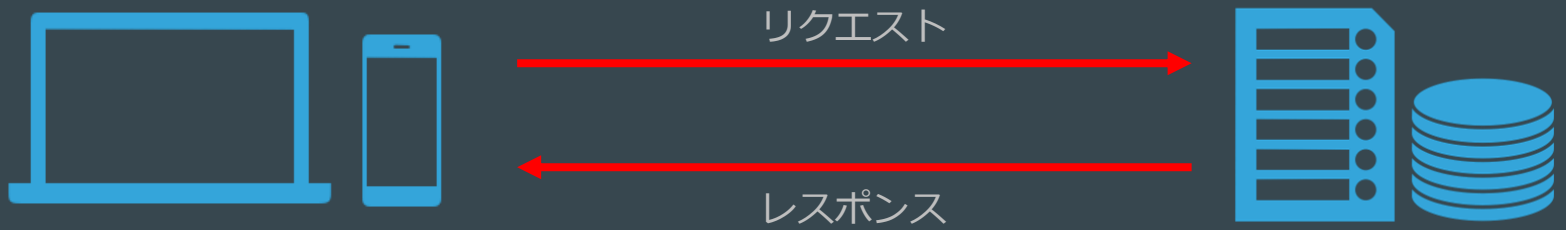
クライアントサイド

リクエスト

https://kusanagi.dht-jpn.co.jp
が見たい

リクエストとレスポンス

まとめ



https://kusanagi.dht-jpn.co.jp
が見たい

送るよ！

疑問

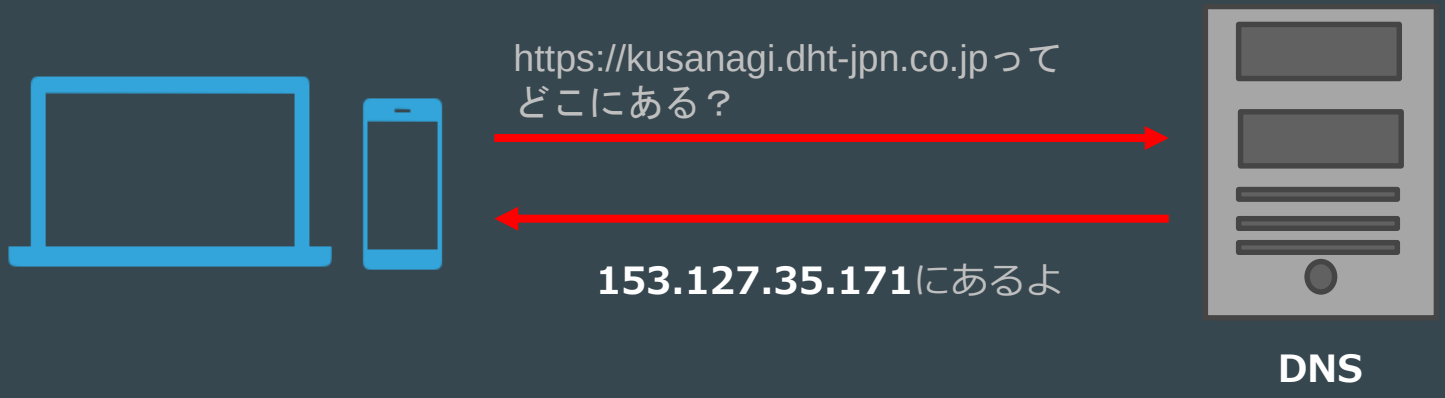
世の中にはたくさんのサイトがあるけど
どうやって見つけているの？

サーバーにもアドレスがある

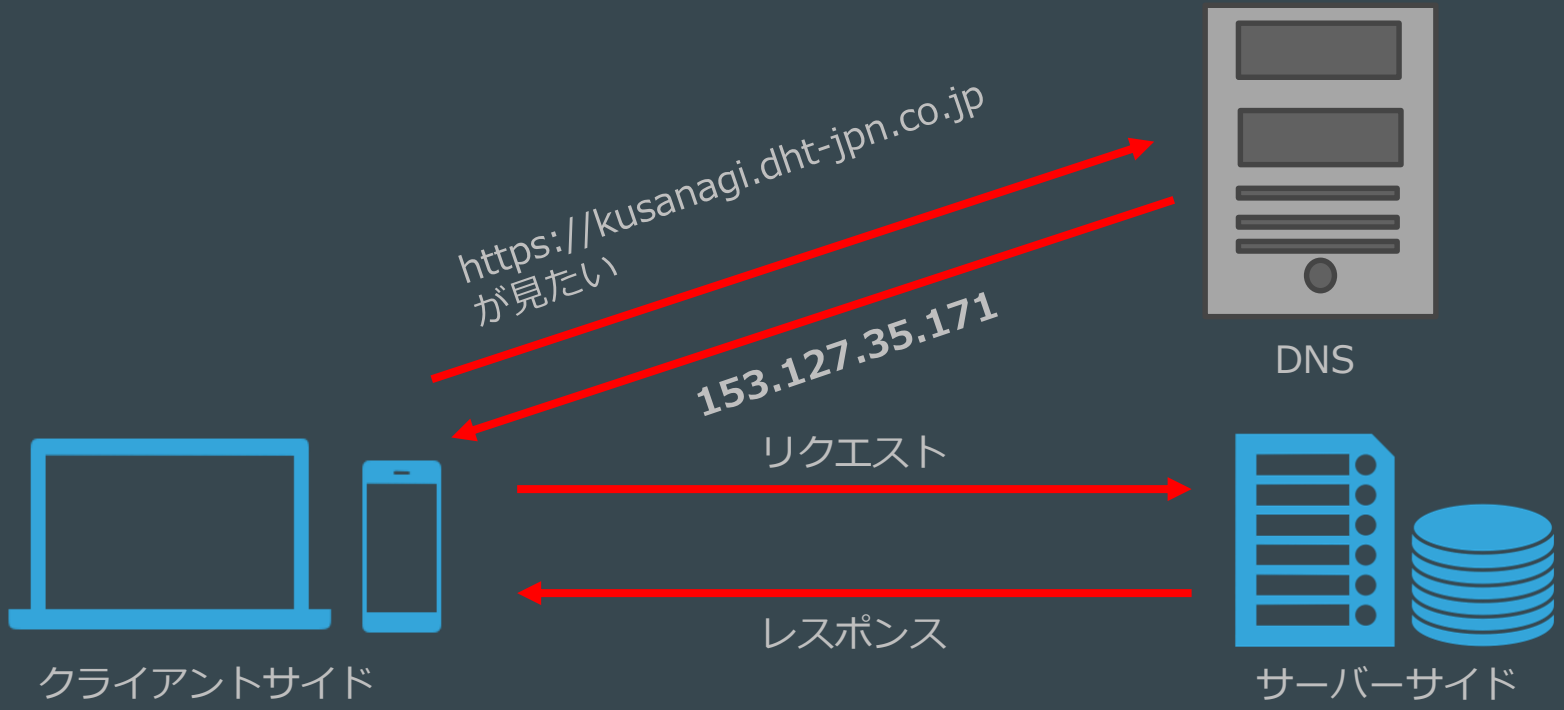
IPアドレス：
153.127.35.171

DNS（ドメインネームシステム）

DNSは、ドメイン名(dht-jpn.co.jp)とIPアドレス(153.127.35.171)を紐づけて管理するシステム

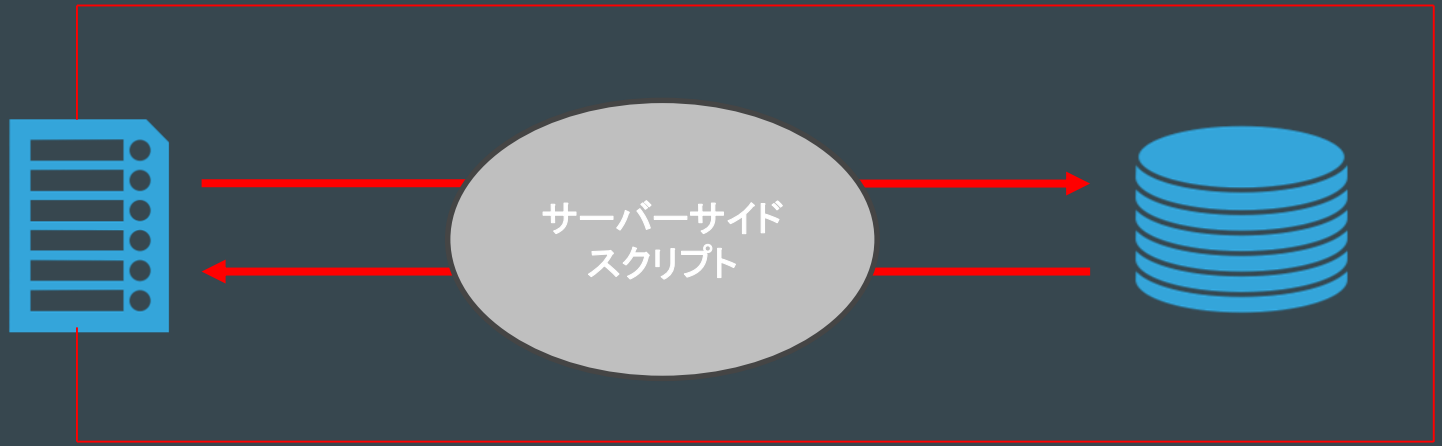


まとめ


https://kusanagi.dht-jpn.co.jp
が見たい
153.127.35.171
DNS
リクエスト
レスポンス
クライアントサイド
サーバーサイド

サーバーサイドスクリプト

・リクエストに応じて、DBから情報を取ってくる。
・DBからの取得情報をもとにHTML/CSSを生成

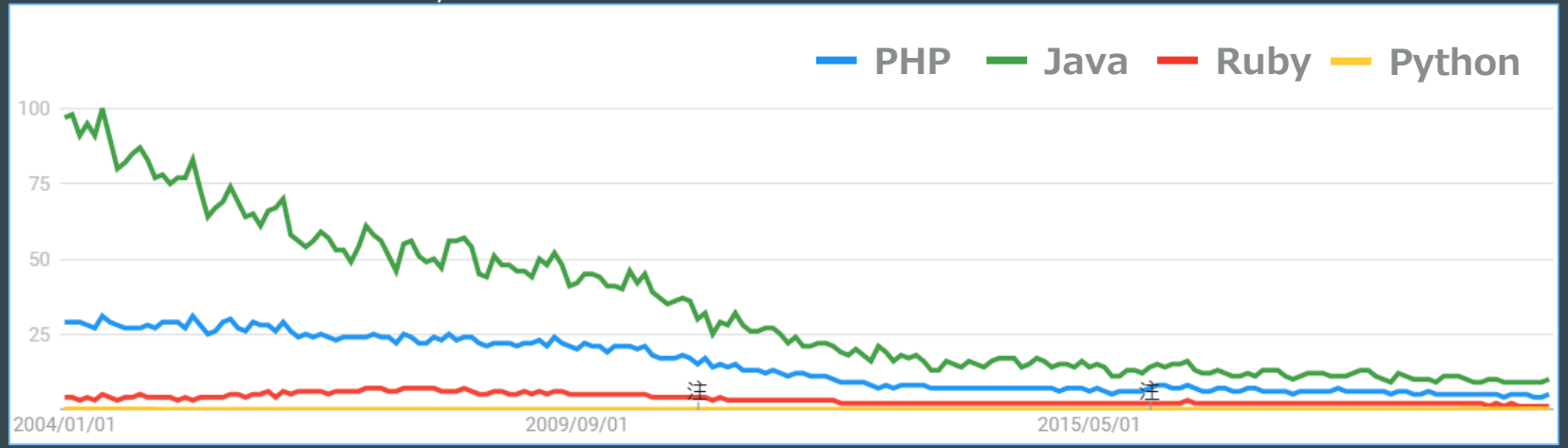


開発言語

開発言語の選定

・**言語の情報量**
情報が多いと、プログラムではまってしまったときに解決策をみつけやすい

・**将来性**
コミュニティなどが活発で，今後も開発が進んでいくか



Google Trendsより

PHPとは

- 動的なWebページ生成が可能
- Webアプリ・サービスを作ることに特化した言語
- FacebookやWikipediaもPHPで開発された
- WordPressの開発にも使われている

PHPコード

```
<html>
  <body>
    <p>
      <?php
      $message = ‘Hello World’;
      print($message);
      ?>
    </p>
  </body>
</html>
```

DBを使う

```
$host = ‘example.com’;
$dbname = ‘example’;
$user = ‘xxx’;
$pass = ‘xxx’;

$db = new PDO(“mysql:host={$host};dbname={$dbname}”, $user, $pass);

$q = $db->query(‘SELECT xx FROM yy’);

while ($row = $q->fetch()) {
    print(“$row[zz]”); // データを表示
}
```

開発でぶつかった問題

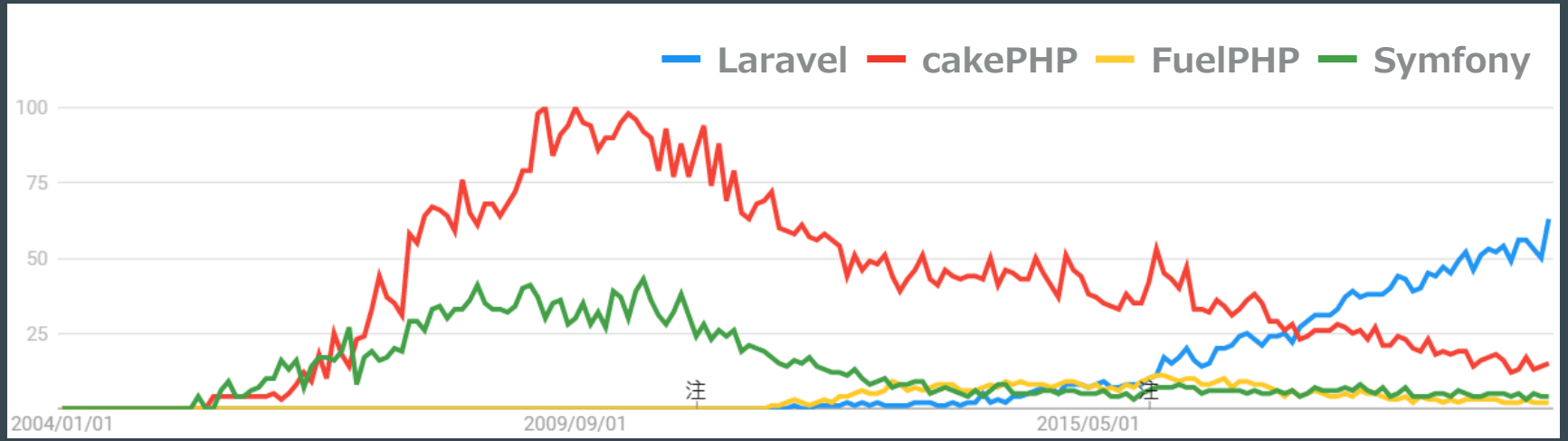
- **マークアップと混ざってしまう**
  最終的なコードが汚くなってしまう
- **phpタグをたどるのが難しい**
  データを用意する部分なのか表示する部分なのかわからなくなる
- **バグが発生したときに，該当箇所を探すのが大変**
  バグの原因が明確でも，どの部分に書いたか探すのが大変

フレームワークの導入

人気のFW

・cakePHPの人気度が下がっている

・laravelの人気度が上昇中



Google Trendsより

Laravelを選んだ理由

・希少価値がある
　　人気になって、時間が経っていない

・短時間で習得できる
　　公式ドキュメントが詳細に書かれていてわかりやすい

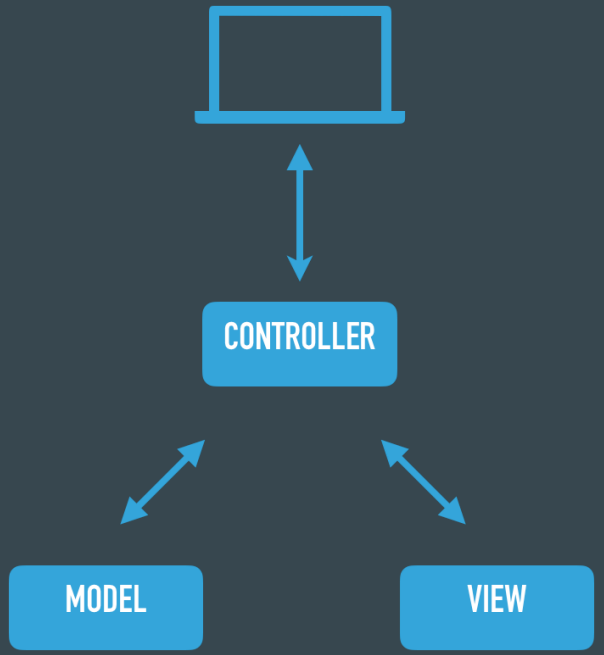
・MVCアーキテクチャ
　　機能別に分けて書けるため，見やすくなり処理が追いやすい

MVC

Model: アプリケーションのデータの管理など

View: ユーザに表示する画面の生成など

Controller: ユーザの入力値をもとに，使用する
ModelとViewを決定する



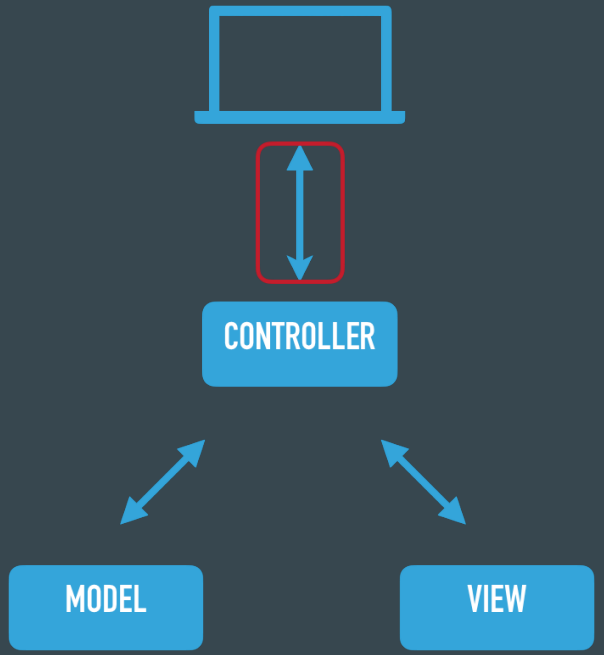
Laravel

Routing

・リクエストされたURLとControllerをつなぐ

・URLに対してどの処理を行うのか明確にする

```
// http://アドレス/foo で呼び出される
Route::get('foo', function ()
{
    return 'Hello World';
});


// コントローラの呼び出し
Route::get('foo', 'FooController@index');
```

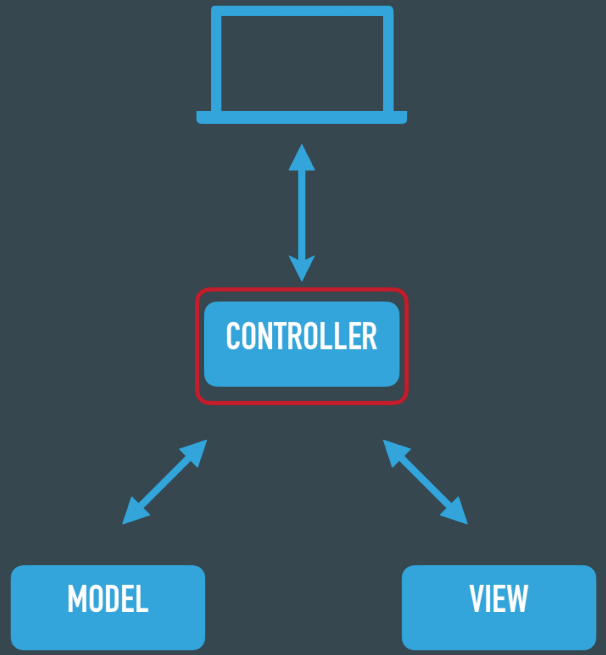


Controller

・Modelからデータを取得

・Viewを利用して画面表示を生成

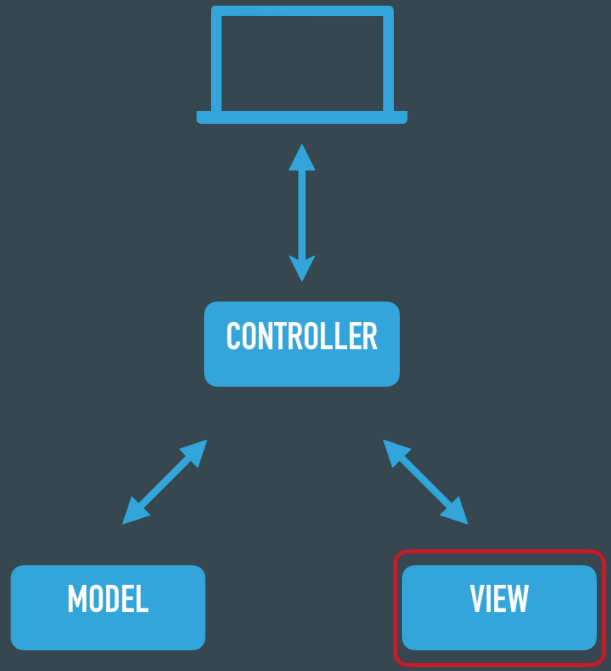
```
class ExampleController extensds Controller
{
    public function index(Request $request)
    {
        // モデル（データ）の用意
        $model = ExampleModel::all()

        // モデルをもとにビューを作成
        $view = view('example', ['model'=>$model]);

        // 作成したビューをレスポンスとして返す
        return $view;
    }
}
```



View

- Bladeというテンプレートエンジンを使用
- @ディレクティブ内で、PHPから渡されたデータを扱える

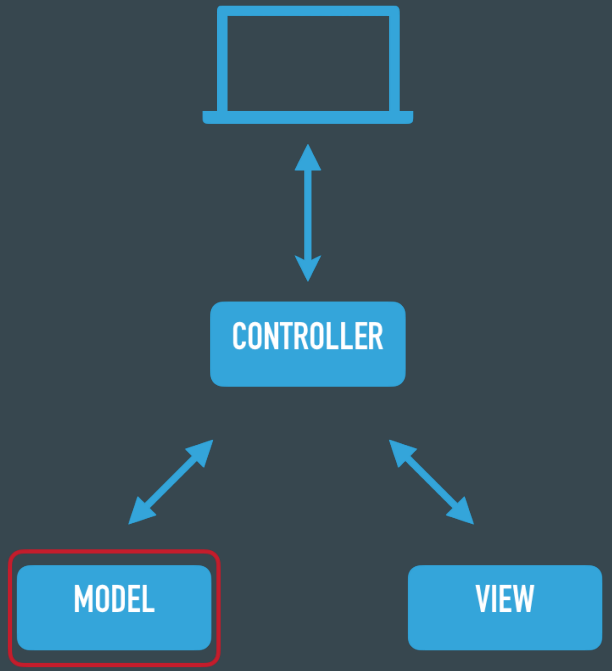
```
<html>
  <body>
    <table>
      {{-- 渡されたデータの表示 --}}
      @foreach ($items as item)
        <tr>
          <td>{{ $item->name }} </td>
          <td>{{ $item->email }} </td>
        </tr>
      @endforeach
    </table>
  </body>
</html>
```



Model

主にデータベースとの連携を行う
LaravelにはEloquentというORMが実装されている

```
class Example extensds Model
{
    //
}
```



DBへの接続

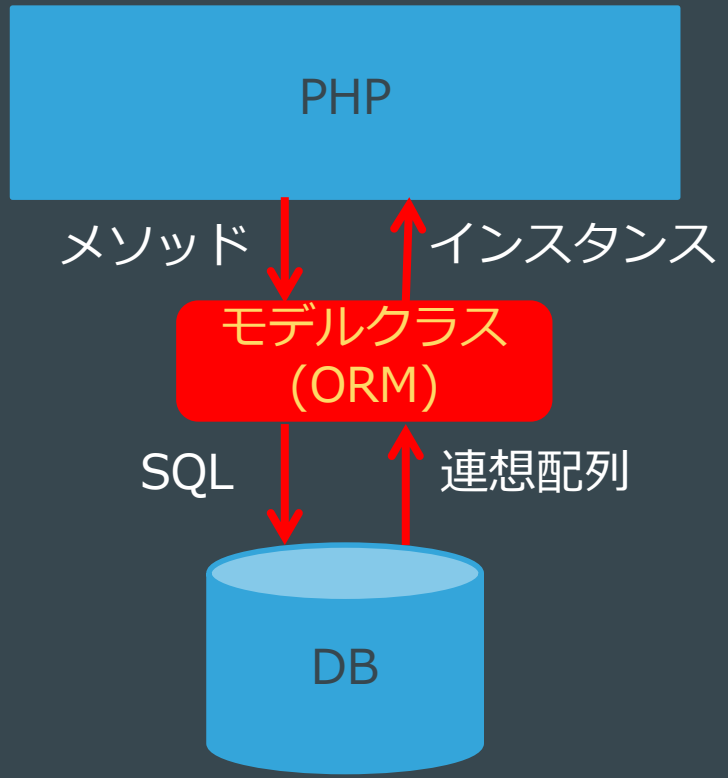
・.envの環境変数とconfigフォルダ内のdatabase.phpに以下のコードを記述

```
DB_CONNECTION=pgsql
DB_HOST=example.com
DB_PORT=5432
DB_DATABASE=xxx
DB_USERNAME=xxx
DB_PASSWORD=xxx
```

Eloquent

PHPのオブジェクトを自動でDBのレコードに変換しDBのレコードをPHPのオブジェクトに変換する。

DBのSQL文を意識せず、
PHPのオブジェクトで処理ができる。



DBへの接続

```
# 空のデータを用意
$data = new Example();

# データを設定
$data->name = ‘xxx’;

# DBにインサート
$data->save();
```

Laravelのドキュメント

ドキュメント章別ページ

前章	リリースノート	アップグレードガイド	貢献ガイド	APIドキュメント
準備	インストール Valet	設定 デプロイ	ディレクトリ構造	Homestead
構成の概念	ライフサイクル 契約	サービスコンテナ	サービスプロバイダ	ファサード
基礎	ルーティング リクエスト セッション	ミドルウェア レスポンス バリデーション	CSRF保護 ビュー エラー処理	コントローラ URL生成 ログ
フロントエンド	Bladeテンプレート	多言語化	スカフォールド	アセットコンパイル
セキュリティ	認証 暗号化	API認証 ハッシュ	認可 パスワードリセット	メール確認
より深く知る	Artisanコンソール イベント 通知	ブロードキャスト ファイルストレージ パッケージ開発	キャッシュ ヘルパ キュー	コレクション メール タスクスケジュール
データベース	データベースの準備 シーディング	クエリビルダ Redis	ページネーション	マイグレーション
Eloquent ORM	Eloquentの準備 APIリソース	リレーション シリアライズ	コレクション	ミューテタ
テスト	テストの準備 データベース	HTTPテスト モック	コンソールテスト	ブラウザテスト
公式パッケージ	Cashier Passport	Dusk Scout	Envoy Socialite	Horizon Telescope
言語 ファイル	auth.php	pagination.php	passwords.php	validation.php

https://readouble.comより

フロントエンド

フロントエンド開発の複雑化

・1991年にWebが登場
　　CSSの装飾，静的なページが多い

・1995年にJavaScript（JS）が開発される
　　動的なページが作成できるようになる
　　JSの実装に絡んだブラウザでセキュリティーホールが見つかる

フロントエンドに求められるもの
　　HTML，CSS，JSを使った見た目の補助

フロントエンド開発の複雑化

・2005年にAjax（Asynchronous Javascript + XML）の登場
　　JSを使って，非同期にサーバからXMLデータを取得

・2009年にNode.jsの登場
　　サーバサイドJSの実行環境.

・2010年にJQueryの登場
　　JSのライブラリ．プラグイン機能が備えてあり，プラグインの開発も可能

フロントエンドに求められるもの
　　Ajaxをベースとした開発

フロントエンドの複雑化

・フレームワークの登場
　　設計の構造化が考慮されるようになった

・双方向バインディングの流行
　　データとViewを自動で更新してくれる

・仮想DOMの流行
　　DOMの書き換え時に，変更部分のみ書き換えることにより高速化する

フロントエンドに求められるもの
　　アプリケーションのプレゼンテーション全般

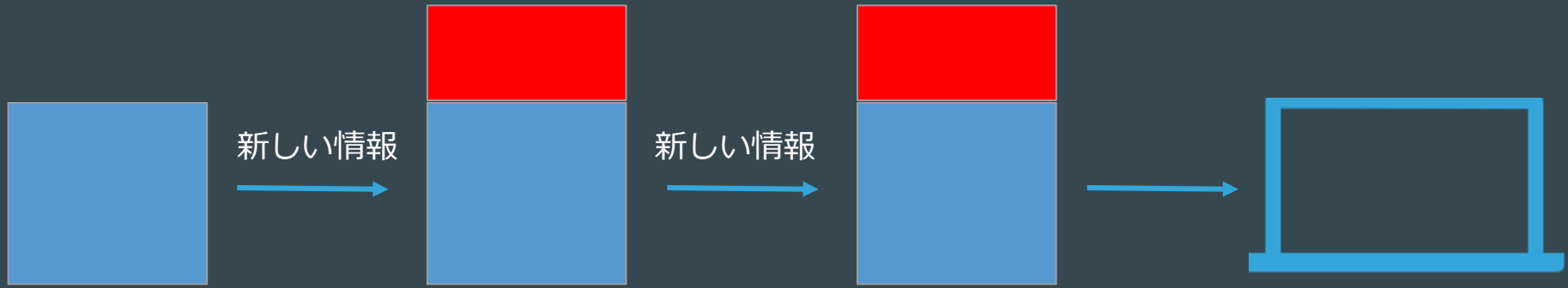
仮想DOM


```
//idの取得
var h1 = document.getElementById(“title”)
//取得したidに変更をくわえている
h1.textContent = “タイトル変更”
```

```
//タイトルにidを追加
<h1 id = “title”>タイトル</h1>
//Javascriptを反映後
<h1 id = “title”>タイトル変更</h1>
```

仮想DOM

必要な部分のみDOMに反映させるから高速化する。



Vue.jsの特徴

- Javascriptのフレームワーク

  →Webアプリケーションを簡単に作ることができる

- 現在のWebアプリケーション開発の流行にのっている

  →双方向バインディング

  →仮想DOM

Vue.jsのセットアップ

```
# laravel/uiをインストール
$ composer require laravel/ui --dev

# Vueのセットアップ
$ php artisan ui vue

# npm経由で開発環境の構築
$ npm install && npm run dev
```

※ # : 説明　$: 入力コマンド

hotservice

- https://hotservice.jp
-

hotservice

- 会議システムはオープンソースソフトウェアのjitsi meetを使用
- 用途

・オンライン展示会
・面接
・英会話などの教育
・飲食店