サーバーサイドスクリプトⅡ

１．Laravel・ルーティングとコントローラ

１－１． Laravelとは

Laravelとは、PHPのフレームワークです。フレームワークとは、「枠組み」や「骨組み」といった意味合いを持つ言葉です。フレームワークでは最初から便利な枠組み、関数、クラスなど必要な処理が用意されています。それらを使用することにより、簡単にプログラムを作成することができます。

１－１－１． Laravelを使用するメリット/デメリット

|  |  |
| --- | --- |
| メリット | セキュリティやメンテナンス性、拡張性を考えて設計された機能を利用できる。 |
| デメリット | （素のPHPと比べて） 自由度が低い。  関数、クラス、メソッドの名前など、フレームワークの規則に従う必要がある。 |

１－１－２． セキュリティ

　メジャなフレームワークでは、基本的な脆弱性には対応されています。フレームワークを利用すれば、フレームワークの利用方法に則った実装を行うだけで、対処法を知らない脆弱性であっても既に守られているコードを実装できます。

|  |  |
| --- | --- |
| 脆弱性 | 内容 |
| XSS | Webサイトに埋め込まれたJavaScriptコードによってCookieなどから情報を抜き出す等の脆弱性 |
| CSRF | 他社になりすましてリクエストしたり、偽物のWebサイトからリクエストさせたりすることを許容してしまう脆弱性 |
| SQLインジェクション | アプリケーションが想定していないSQLを実行させられてしまう脆弱性 |

１－２． Laravelの特徴

Laravelには以下のような特徴があります。

|  |  |
| --- | --- |
| MVCフレームワーク | LaravelはアプリケーションをModel-View-Controllerの各機能に分けて整理し、これらのパーツを作ることで開発を行います。MVCは、アプリケーションフレームワークで広く使われているアーキテクチャです。  ※MVCそれぞれについて、詳しくは後述します。 |
| 低い学習コスト | Laravelは非常にコードが分かり易く、習得するために多くの時間と労力を必要としません。多くの機能を持っていますが、実装が容易で、簡単なコードで各種の機能を利用できるようになります。 |
| Symfonyベース | Laravelはプログラムの土台にSymfonyを使っています。SymfonyはPHPで古くから使われているフレームワークで、大規模な開発に多くの実績があり、非常に堅牢なシステムが土台となっています。 |
| ORM | Laravelではデータベースアクセスに、ORM（Object-Relational Mapping）と呼ばれる技術を導入しています。これにより、PHPのオブジェクトを扱うようにデータベースを利用できます。 |
| Bladeテンプレート | Laravelでは画面の表示には一般的なPHPの他に「Blade」と呼ばれるテンプレートエンジンを搭載しています。これにより、複雑なデザインも分かり易く、すっきりと記述できるようになります。 |

１－３． Laravelのディレクトリ構造

■ディレクトリ構造

|  |  |  |  |
| --- | --- | --- | --- |
| グラフィカル ユーザー インターフェイス  中程度の精度で自動的に生成された説明 | |  | | --- | | app | | Console、Exceptions，Http、Providersの各ディレクトリがあります。  コントローラーやミドルフェア、例外クラス、コンソール、サービスプロバイダーなど、アプリケーションの主要な処理クラスは「 app 」ディレクトリ内にあります。 | |
| |  | | --- | | bootstrap | | アプリケーションで最初に実行される処理やオートローディング設定が入っています。 | |
| |  | | --- | | config | | アプリケーションの設置を記載したファイルを入れます。 | |
| |  | | --- | | node\_modules | | フロントエンドのアプリケーションの依存するライブラリファイルが配置されます。 | |
| |  | | --- | | database | | データベース関連のファイルが入っています。マイグレーションファイルや初期投入データなどを置きます。 | |
| |  | | --- | | public | | Webアプリケーションとして公開する場合は、このフォルダをドキュメントルートに設定します。エントリポイントとなるindex.phpが入っているほか、JavaScriptやCSSなど公開するファイルを配置するディレクトリです。 | |
| |  | | --- | | resources | | Viewのテンプレートファイルや、LESSやSASSなどのメタ言語ファイル、言語ファイルを配置します。 | |
| |  | | --- | | routes | | アプリケーションのルート定義のファイルを配置します。 | |
| |  | | --- | | storage | | プログラム実行時にLaravelが生成するファイルの出力先となります。ログファイルやファイルキャッシュのほか、コンパイルされたテンプレートファイルなども保存されます。 | |
| |  | | --- | | tests | | テストコードを記載したファイルを配置します。 | |
| |  | | --- | | vendor | | Composerによってダウンロードされる各種パッケージのディレクトリとなります。LaravelやSymfonyのコードもここに入っています。vendorディレクトリ配下のlaravelディレクトリが、Laravel本体のコード配置場所です。 | |

１－４． MVCモデルとは

LaravelはMVCと呼ばれるアプリケーションのアーキテクチャ（構造）を採用しています。MVCはモデル（Model）、ビュー（View）、コントローラ（Controller）の略です。

|  |  |
| --- | --- |
| 名称 | 内容 |
| Model | データ処理全般を担当します。データベースやデータリソースにアクセスし、データの作成・更新・削除といったビジネスロジックを実装する領域です。 |
| View | 画面表示を担当します。WebサイトであればHTML、HTTP APIであればXMLやJSONといった形式でレスポンスを出力するための領域です。 |
| Controller | 全体の制御を担当します。必要に応じてModelを使ってデータを取得したり、Viewを利用して画面表示を作成したりします。  Controllerはルーターと紐づいてURLのパスに対応します。  ルーターとModel、Viewの橋渡し役となる存在です。 |

■モデル・ビュー・コントローラの関係

ダイアグラム

自動的に生成された説明

これらの仕組みに則ってアプリケーションを作成することで、全体の見通しが良くなり、開発効率も高まります。

サーバーサイドスクリプトⅡでは、このMVC構造を理解し、Laravelの使い方を学んでいきます。まずは基本的な仕組みを把握してください。

１－５． MVC開発の手順

1. マイグレーション
2. モデルの作成
3. **コントローラの作成**
4. **ルーティングの設定**
5. ビューの作成

Laravelでは、データベースのテーブル定義も管理もプロジェクト上で行います。テーブルを記述するファイルをマイグレーションファイル、定義をデータベースに反映する作業をマイグレートと呼びます。マイグレーションとモデル、ビューについては、後の回の講義にて行います。

今回は、データベースやビューはまだ利用せずに、コントローラを用います。手順では③④の部分に当たります。

■図：　今回の範囲（赤枠部分）　MVCのC

ダイアグラム

自動的に生成された説明

１－５－１． コントローラの作成

まずはコントローラを作成します。

■コントローラについて

|  |  |
| --- | --- |
| 役割 | **コントローラはリクエストを受け取り、他のクラスへ作業を依頼**します。 |
| 作成方法 | コントローラは、Laravel標準のControllerクラスを継承して作成します。  作成するコントローラのクラス名は、必ず「 ＊＊＊＊Controller 」とクラス名の末尾をControllerとします。  ※手動で作成せず、コマンドを用いて作成します。（後述） |

■コントローラ作成コマンド

テキスト

中程度の精度で自動的に生成された説明

今回は「Kadai01Controller」を作成します。

VSCodeのターミナルを開いてください。

（VS Code [ターミナル] – [新しいターミナル]）

グラフィカル ユーザー インターフェイス, アプリケーション, Word

自動的に生成された説明

succesｓfullyと表示されればOK。また、コントローラが作成された場所も表示されています。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, Word

自動的に生成された説明

１－５－２． コントローラの処理

作成したKadai01Controller.phpファイルを開いてください。

※ファイルは app\Http\Controllers\　の配下に作成されています。

下記は、makeコマンドで作成した初期状態のファイルです。Controllerクラスが継承されていることを確認しましょう。

現在は、クラスしかないので、メソッドを作成していきます。

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, メール

自動的に生成された説明

クラスが持つ関数であるメソッドは、以下の構文で定義します。

　■構文

**テキスト

低い精度で自動的に生成された説明**

　アクセス修飾子は、public、protected、privateのいずれかを指定します。

■プロパティとメソッドのアクセス権

|  |  |
| --- | --- |
| アクセス修飾子 | 意味 |
| public | どこからでもアクセスできる |
| protected | 自クラスと、そのクラスを継承するサブクラスからのみアクセスできる |
| private | 自クラスからのみアクセスできる |

今回は、下記のメソッドを追加します。

　■仕様

|  |  |
| --- | --- |
| アクセス修飾子 | public |
| メソッド名 | index |
| 引数 | なし |
| 処理 | 文字列　「Hello Laravel Application」を　返す。 |

　■Kadai01Controller.php

class Kadai01Controller extends Controller

{

    public function index()

    {

        return "Hello Laravel Application";

    }

}

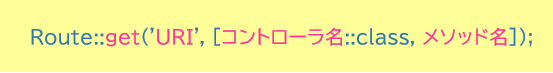
１－５－３． ルーティングの設定

続いて、ルーティングの設定を行います。

■ルーティングについて

|  |  |
| --- | --- |
| 役割 | ルーティングとは、クライアントから送信されたリクエストをどのコントローラが担当するのかの割り当てを行うことです。 |
| 設定方法 | ルーティングの設定は、下記のファイルで行います。  **routes/web.php**  ※詳細は後述。 |

　■構文　web.phpのRoute::getメソッド



URL「https://localhost:8000/」の最後の「/」をルートといいます。「/」と「**/文字列**」がルート定義で指定できる箇所です。Route::getメソッドの第1引数に、この「/文字列」を指定します。

今回は、下記の設定とします。

　■仕様

|  |  |
| --- | --- |
| リスエスト形式 | Get |
| URI | https://localhost:8000**/kadai01** |
| 呼び出しクラス・メソッド | Kadai01Controller の　index　メソッド |

　■web.php **※末尾に追加**

// Kadai01

Route::get('/kadai01', [Kadai01Controller::class, 'index']);

記述が終わったら、web.php の上部にKadai０１Controllerクラスを使用するuse文が追加されていることを確認してください。自動的に追加されない場合は、自分で追加しましょう。

■web.php

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション

自動的に生成された説明

１－５－４． 実行確認

ルーティングの設定ができたら、実行を確認します。

開発サーバーを起動します。

ターミナルで、下記のコマンドを打ち込んでください。

■開発サーバー起動コマンド

図形 が含まれている画像

自動的に生成された説明

サーバーが起動したら、ブラウザで動作を確認してください。

仕様で指定したURLを打ち込みます。

|  |  |
| --- | --- |
| URL | https://localhost:8000/**kadai01** |

ブラウザで、Kadai01Controllerのindexメソッドで戻り値に指定した文字列が表示されていればOKです。

■実行結果

グラフィカル ユーザー インターフェイス, テキスト, アプリケーション, チャットまたはテキスト メッセージ

自動的に生成された説明

ブラウザからリクエストを送ると、web.phpが指定されたクラスのメソッドに処理を振り分けます。今回はViewを使用していませんので、ブラウザにはコントローラのindexメソッドでreturnした文字列のみが表示されます。

