



## 2. Laravelのインストール

### composer でインストール

Composer で Laravel プロジェクトを作成します。ターミナルや PowerShell でコマンドを入力します。

```
% composer create-project オプション laravel/laravel プロジェクト名
```

### 最新バージョンの場合

バージョン省略すると 最新バージョンがインストール されます。

```
% composer create-project laravel/laravel プロジェクト名
```

### バージョン指定する場合 (7.x)

laravel/laravel のあとに バージョンの指定 できます。

```
% composer create-project laravel/laravel=7.x プロジェクト名
```

### prefer-dist

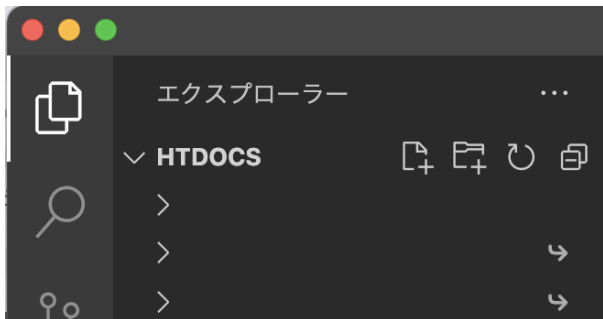
prefer-dist は Github からダウンロードするオプション です。

```
% composer create-project --prefer-dist laravel/laravel プロジェクト名
```

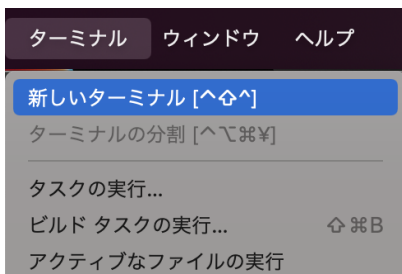
### Laravel プロジェクト作成

## ターミナルでWebルートを開く

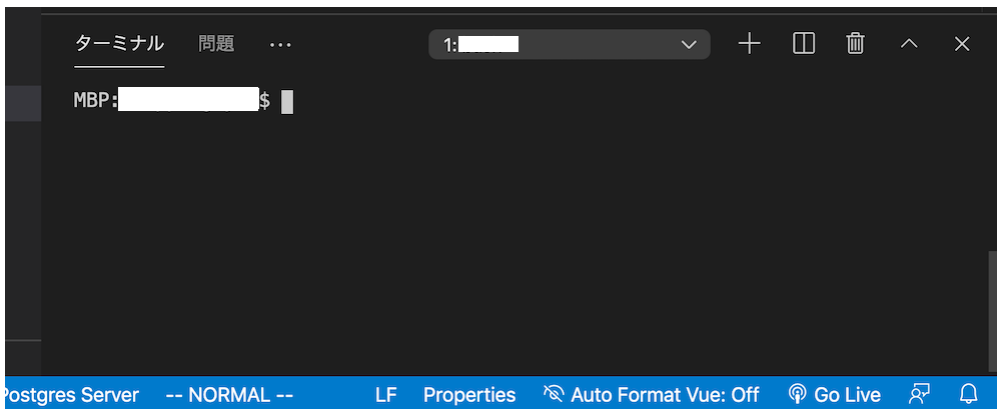
VSCode で `htdocs` などの Webルートを開きます。



「ターミナル > 新しいターミナル」を開きます。



ターミナルが開きました。



## Composer でインストール

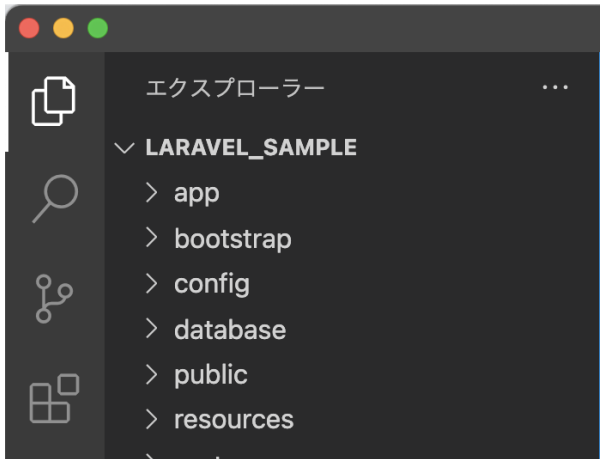
ターミナルで、`Laravel` バージョン8 のプロジェクトを作成します。プロジェクト名は `laravel_sample` にします。

```
% composer create-project laravel/laravel=8.x laravel_sample
```

## プロジェクトの確認

## プロジェクトフォルダを開く

作成したプロジェクト `laravel_sample` を `VSCode` で開きなおします。



## ファイル構成

作成されたプロジェクト内は、たくさんのファイルやフォルダが確認できます。

```
.
├── .editorconfig
├── .env
├── .env.example
├── .gitattributes
├── .gitignore
├── .styleci.yml
├── README.md
├── app
├── artisan
├── bootstrap
├── composer.json
├── composer.lock
├── config
├── database
├── package.json
├── phpunit.xml
├── public
├── resources
├── routes
├── server.php
├── storage
├── tests
├── vendor
└── webpack.mix.js
```

**Laravel** でよく利用するファイルやフォルダは以下のとおりです。

## フォルダ 説明

app/	<b>MVC</b> のモデル、コントローラーをはじめとする、 <u>主要なプログラムを格納する</u> フォルダです。
resources/	<b>View</b> 、 <b>JS</b> 、 <b>SCSS</b> といったリソースファイルを格納します。
public/	<u>Web 公開用のフォルダ</u> で、 <b>JS</b> 、 <b>CSS</b> 、 画像ファイルなどを格納します。 <b>public</b> フォルダ以外は、原則 <b>Web</b> に公開されません。
routes/	フレームワークに搭載されている <u>ルーティング (Routing) の設定ファイル</u> です。ルーティングは、 <b>URI</b> と起動コントローラーを基本に設定します。
database/	<u>データベースの設定ファイル (マイグレーション)</u> を中心に、テストデータなどを格納します。
.env	アプリ名、DB、メールといった、 <u>個人的な設定に関するファイル</u> です。個人設定のため、 <b>Git</b> 管理しません。
vendor/	<b>vendor</b> には <b>Laravel</b> の核となる <u>大量のライブラリファイル</u> が詰まっています。勉強するにはとてもよい題材ですが、修正してしまうと動作しなくなりますので注意してください。

## artisan コマンド

### artisan コマンドとは

**artisan** コマンドは、**Laravel** に搭載されたPHPコマンドです。 **artisan** コマンドを利用するとコントローラーやモデルの作成、データベースの更新（マイグレーション）をはじめ、効率よく開発できます。

```
% php artisan コマンド
```

## プロジェクト直下で実行

---

`artisan` コマンドは作成したプロジェクトフォルダ直下に存在するので、`VSCode` でプロジェクトを開き、ターミナルでコマンド実行します。 ※プロジェクトフォルダ直下でしか実行できません

## バージョンを確認する

---

プロジェクトディレクトリ内で `artisan` を利用して Laravel バージョンを確認してみます。

```
% php artisan -V
Laravel Framework 8.5.0
```

以下のようにたくさんのコマンドが用意されています。

```
Available commands:
clear-compiled  Remove the compiled class file
db             Start a new database CLI session
down          Put the application into maintenance / demo mode
env           Display the current framework environment
help         Display help for a command
inspire       Display an inspiring quote
list          List commands
migrate       Run the database migrations
optimize      Cache the framework bootstrap files
serve        Serve the application on the PHP development server
test         Run the application tests
tinker        Interact with your application
up           Bring the application out of maintenance mode
auth
```

## Laravel のバージョン確認

---

ターミナルで `Laravel` のバージョンを確認してみましょう。 ※バージョンはインストール時に指定したもの

```
% php artisan -V
Laravel Framework 8.70.2
```

## アプリケーションキーの作成

---

`Laravel` を動作させるには アプリケーションキーの生成が必要で、プロジェクト作成時に自動で `.env` ファイルに書き込まれます。環境再構築などでアプリケーションキーが必要な場合

は、 `key:generate` で生成します。

```
% php artisan key:generate
Application key set successfully.
```

## Laravel サーバの起動

Laravel は PHP のサーバ機能で独自起動でき、開発で動作確認するときに非常に便利です。  
サーバは `artisan serve` コマンドで起動します。

```
% php artisan serve
Starting Laravel development server: http://127.0.0.1:8000
```

## ポート番号

デフォルトでは `http://localhost:8000` で起動します。もしポートを指定したい場合は、以下のコマンドになります。

```
% php artisan serve --port ポート番号
```

## 本稼働では `artisan serve` は利用しない

本稼働では Webサーバ（Apache、Nginxなど）経由で Laravel を実行するため、`artisan serve` コマンドは利用しません。

## Laravel の起動確認

ブラウザで `http://localhost:8000` にアクセスすると `Laravel` のトップ画面が表示されます。



### Documentation

Laravel has wonderful, thorough documentation covering every aspect of the framework. Whether you are new to the framework or have previous experience with Laravel, we recommend reading all of the documentation from beginning to end.



### Laracasts

Laracasts offers thousands of video tutorials on Laravel, PHP, and JavaScript development. Check them out, see for yourself, and massively level up your development skills in the process.



### Laravel News

Laravel News is a community driven portal and newsletter aggregating all of the latest and most important news in the Laravel ecosystem, including new package releases and tutorials.



### Vibrant Ecosystem

Laravel's robust library of first-party tools and libraries, such as [Forge](#), [Vapor](#), [Nova](#), and [Envoyer](#) help you take your projects to the next level. Pair them with powerful open source libraries like [Cashier](#), [Dusk](#), [Echo](#), [Horizon](#), [Sanctum](#), [Telescope](#), and more.

何もプログラミングしていませんが、[composer](#) だけで [PHP のアプリケーション](#)が作成できました。開発では、[Laravel](#) のサーバで確認するのが手軽なので、サーバの起動方法は覚えましょう。

## Laravel サーバを終了する

起動中のターミナル上で [Ctrl + C](#) キーを押して [Laravel サーバを停止](#)します。停止後はターミナル入力モードに戻ります。

```
% php artisan -V
Laravel Framework 8.70.2
```

当サイトの教材をはじめとするコンテンツ（テキスト、画像等）の無断転載・無断使用を固く禁じます。これらのコンテンツについて権利者の許可なく複製、転用等する事は法律で禁止されています。尚、当ウェブサイトの内容をWeb、雑誌、書籍等へ転載、掲載する場合は「ロジコヤ」までご連絡ください。