

laravelの環境構築

Composer



Composerとは

<https://getcomposer.org/>


Composerは、PHPのソフトウェアおよび必要なライブラリの依存関係を管理するパッケージ管理システムになります。コマンドライン上で動作し、アプリケーションが依存するライブラリなどのインストールや更新をします。また、Laravelを含めた有名なオープンソースのPHPプロジェクトの重要な機能の一部として利用されています。

Composerのインストール

<https://getcomposer.org/doc/00-intro.md#installation-windows>

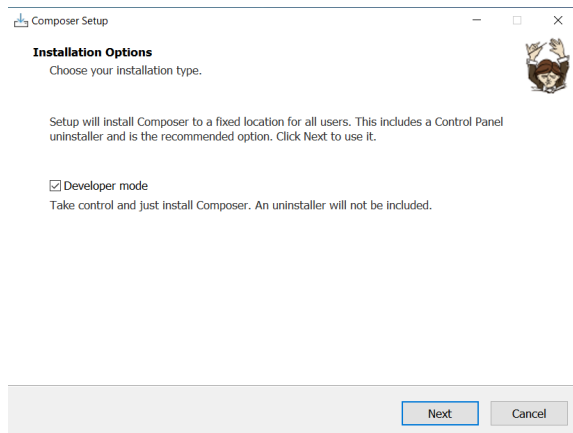
Introduction - Composer

Composer is a tool for dependency management in PHP. It allows you to declare the libraries your project depends on and it will manage (install/update) them for you. Composer is not a package manager in the same sense as Yum or Apt are.

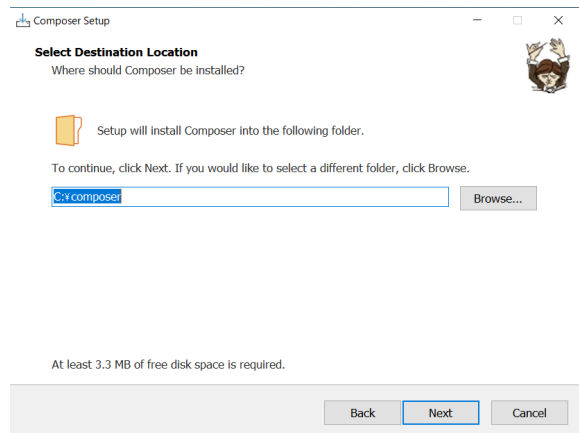
 <https://getcomposer.org/doc/00-intro.md#installation-windows>

インストーラーexeのダウンロードページ

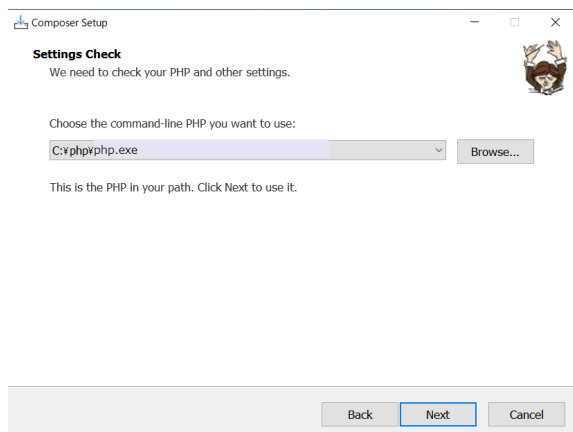
「Composer-Setup.exe」のリンクからexeファイルをダウンロードしてください。
「Composer-Setup.exe」のダウンロードができれば、exeファイルをダブルクリックしてイントール画面を進めていきます。



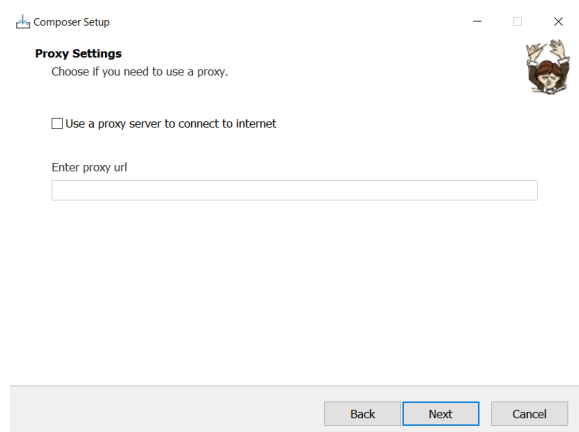
Developer modeにチェックを入れましょう。



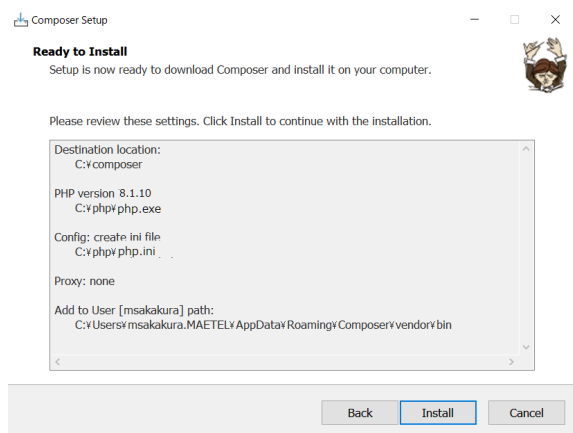
Cドライブの直下へ「composer」というフォルダでインストール場所を指定してください。



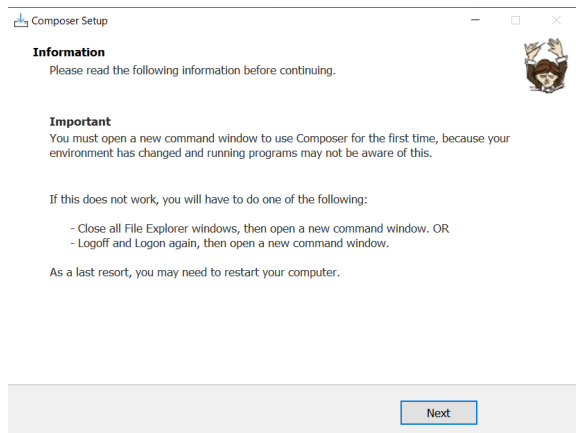
環境変数へのPHPパスが設定されている場合、自動的に「php.exe」へのパスが入力されます。



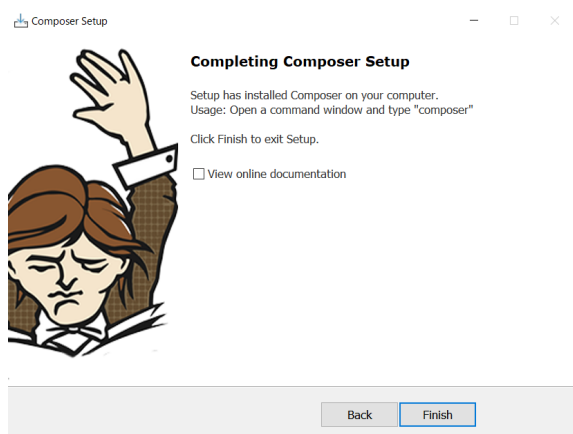
proxyのURLは入力しなくて大丈夫です。



[Install] ボタンをクリックしてインストールを進めてください。



[Next] で進みましょう。



[Finish] でインストール作業の終了となります。

Composerの確認

コマンドプロンプトを起動して、composerのバージョン確認コマンドを実行します。

```
composer -v
```

```
コマンド プロンプト
Microsoft Windows [Version 10.0.19H42.1000]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Masakura.MAETEL\composer > composer -v

Composer version 2.2.0 2022-02-04 17:00:38

Usage:
  composer [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
  --ansi                   Force ANSI output
  --no-ansi                Disable ANSI output
  -n, --no-interaction      Do not ask any interactive question
  --profile                Display timing and memory usage information
  --no-plugins             Whether to disable plugins
  --no-scripts             Skips the execution of all scripts defined in composer.json file.
  --working-dir=WORKING-DIR If specified, use the given directory as working directory.
  --no-cache               Prevent use of the cache
  -v, --verbose             Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug
```

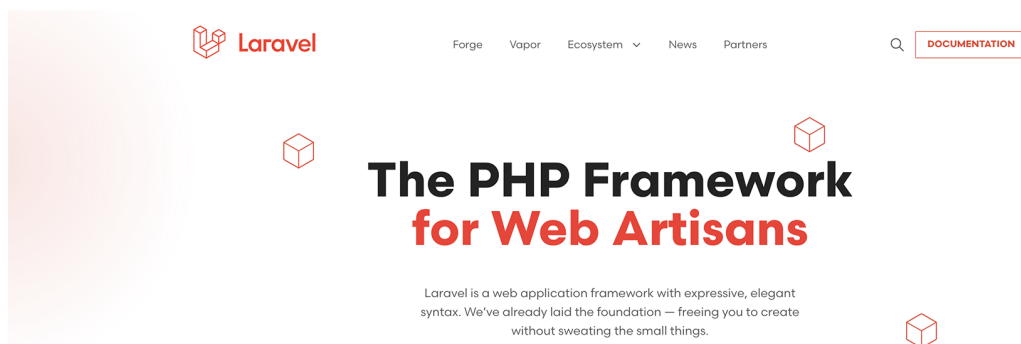
PHPの確認

コマンドプロンプトを起動して、PHPのバージョン確認コマンドを実行しましょう。

```
php -v
```

バージョンが表示されればOK。バージョン情報が表示されない場合は、PHPのパス設定が正しくできているか再確認しましょう。

Laravel



Laravelとは

Laravelは2011年にリリースされたPHPのフレームワークです。PHPフレームワークの中では後発ながら、その手軽さや扱いやすさから人気を集め、今やPHPの中でも代表的なフレームワークとなっています。マイクロソフトの.NETの開発に関わっていたTaylor Otwell氏によって開発され、活発なコミュニティ活動により今もどんどんアップグレードされているフレームワークになります。PHPフレームワークの一つであるSymfonyを土台に開発されており、今までPHPの別のフレームワークを触ってきた人にとっても比較的扱いやすいと言われています。

laravelプロジェクトの作成

「Sites」フォルダの直下に授業用Laravelプロジェクトを作成（Laravelファイル群のインストール）します。

コマンドプロンプトを起動して、「Sites」フォルダ直下まで移動しましょう。

```
cd c:\Sites
```

「Sites」フォルダの直下まで移動できたら、composerを使ってLaravelのインストールを行います。

授業で使用するLaravelのバージョンは8.x系を利用します。



composer create-project --prefer-dist "laravel/laravel:バージョン" プロジェクトフォルダ名

```
composer create-project --prefer-dist "laravel/laravel:8.*" laravel8
```

プロジェクトフォルダが作成され、フォルダ内にLaravelのファイル群がダウンロードされます。

「~~~ successfully」と表示されればダウンロード完了。

Laravelの確認

コマンドプロンプトで、作成されたLaravelプロジェクトフォルダまで移動。

```
cd c:¥Sites¥laravel8
```

※ 既にSitesフォルダの直下にいる場合は、「cd laravel8」でOK。

laravel8フォルダへ移動したら、artisanコマンドでバージョンを確認。

```
php artisan -v
```

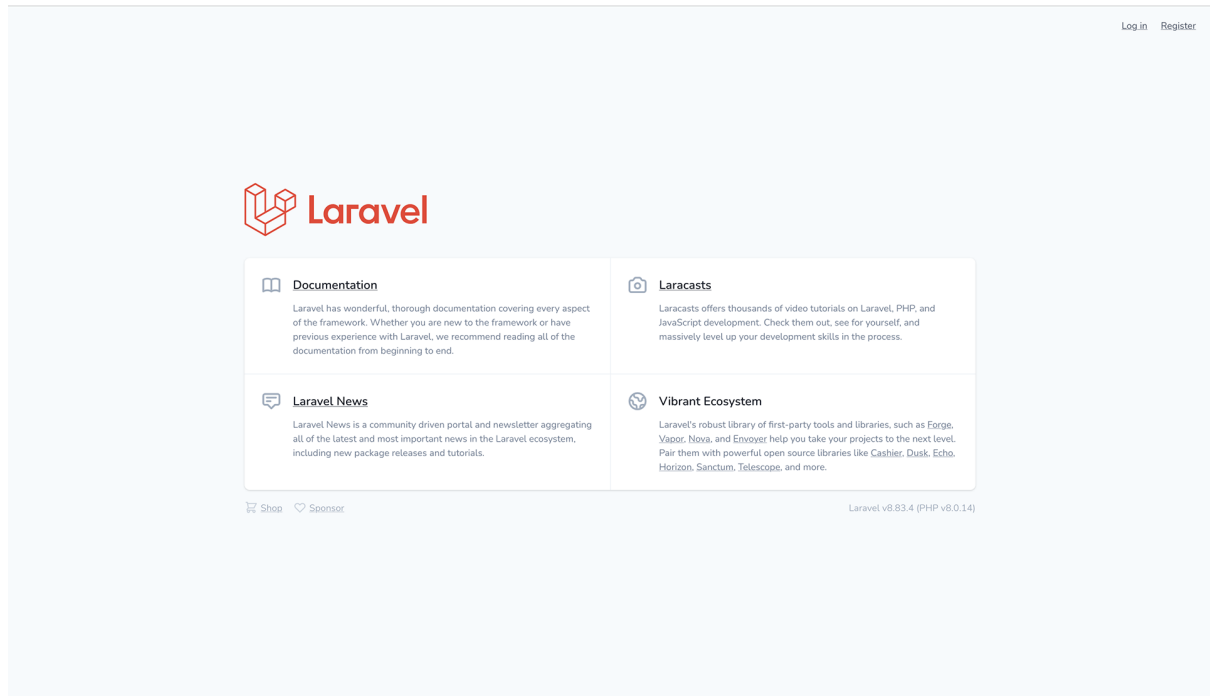
Laravelサーバーの起動と確認

Laravelプロジェクトの作成が完了したら、Laravelで構築されたWebページを確認してみます。

授業では、Laravelの仮想Webサーバーで課題などを確認しながら開発を進めていきます。

```
php artisan serve
```

ブラウザのURLに「<http://127.0.0.1:8000>」と入力し、LaravelのWelcomeページが表示されればOK.



Node.js



Node.jsとは

<https://nodejs.org/ja/>

Node.jsとは、Google Chromeのために開発されました。PHPやRubyやPythonやJavaなどのプログラミング言語と同じく、サーバーの方で動作を行うためのJavaScriptで、「サーバーサイドJavaScript」と呼ばれることもあります。

Node.jsの確認

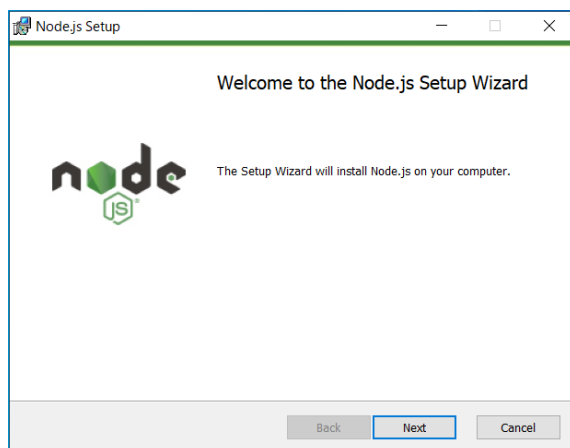
コマンドプロンプトを起動して、PHPのバージョン確認コマンドを実行しましょう。

```
node -v
```

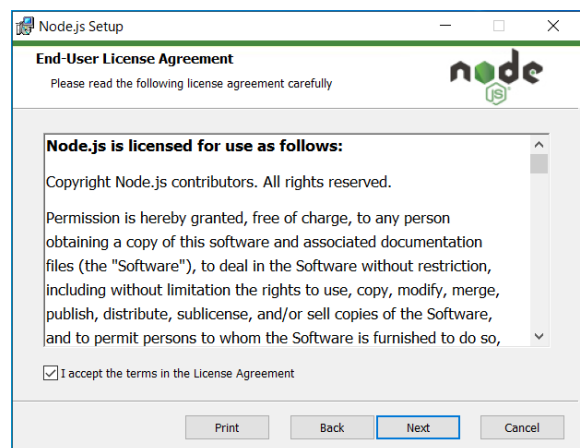
Node.jsのインストール

Node.jsが既にインストールされている人は、インストール作業はする必要がありません。

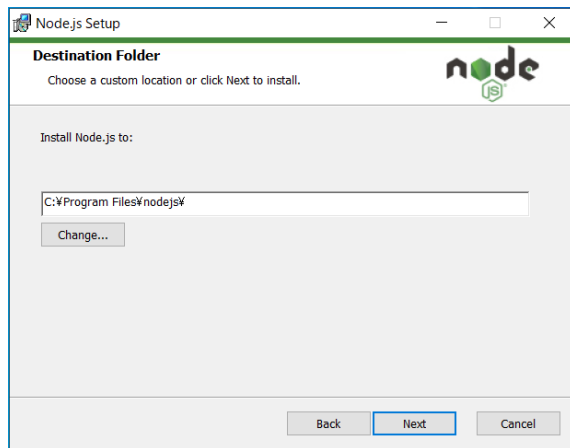
基本的にインストール画面の指示に従って、次に進んでいくだけになります。



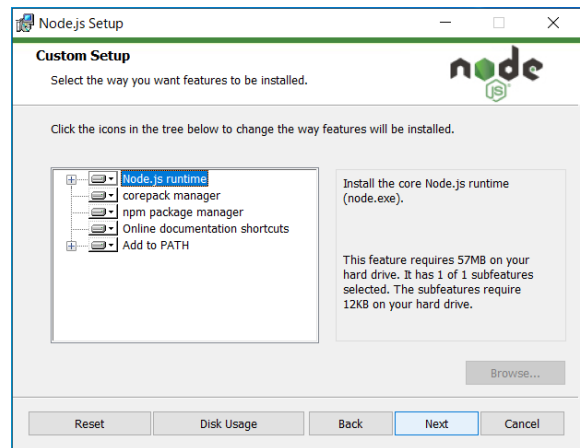
[Next]ボタンを押して進めましょう。



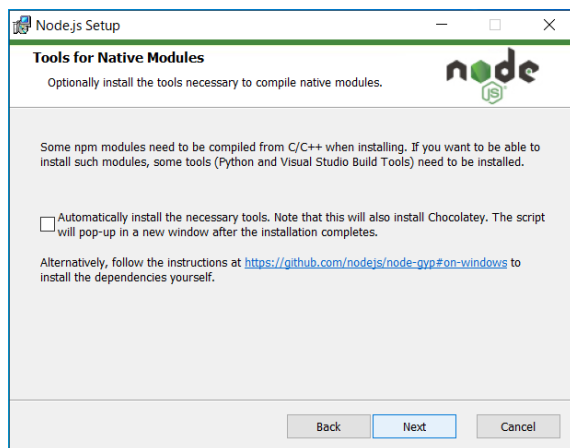
ライセンスの同意にチェックを入れて、[Next]ボタンで進めましょう。



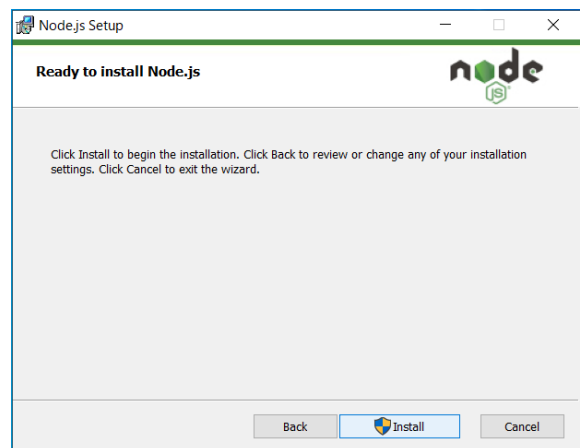
インストールフォルダのパスも特に変更する必要はありません。



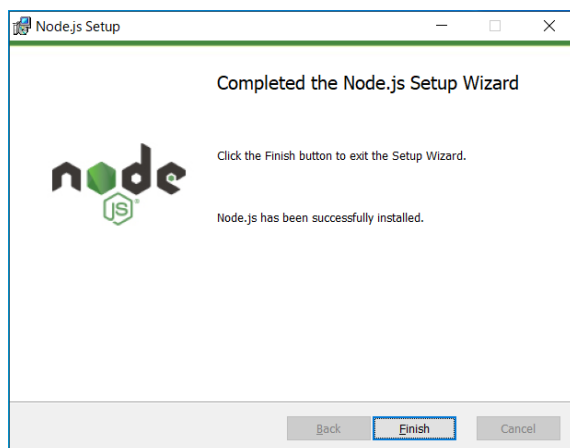
[Next]ボタンを押して進めましょう。



特にチェックを入れる必要はありません。[Next]ボタンを押して進めましょう。



[Install]ボタンを押してインストールを開始しましょう。インストールデータをダウンロードしながらになるため、インターネット環境によって時間がかかる場合があります。



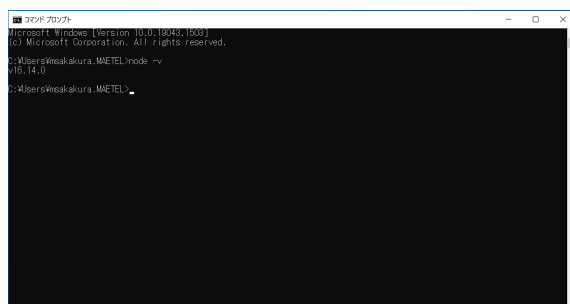
Node.jsのインストールが終了となります。[Finish]ボタンを押してウィンドウを閉じましょう。

Node.jsの確認

コマンドプロンプトを起動して、Node.jsのバージョン確認コマンドを実行しましょう。

```
node -v
```

インストールで自動的にパスが設定されるため、正常にインストールができていればバージョン情報が表示されます。



Laravel UI

認証スcaffoldなどに必要となるので、コマンドプロンプトを使ってLaravel UIをインストールします。

Breeze (Tailwind CSS)

今回はBootstrapではなく、Breezeを利用してみます。Bootstrap版も記載していますが、実行する必要はありません。

コマンドプロンプトを起動して、cdコマンドでLaravelプロジェクトのフォルダまで移動しておきましょう。

composerからbreezeをダウンロードして追加します。

```
composer require laravel/breeze --dev
```

追加したbreezeをartisanコマンドでインストールします。

```
php artisan breeze:install
```

npmコマンドで、Laravelに必要なNodeの依存ファイルをダウンロードします。

```
npm install
```

Node.jsを実行ます。
CSSファイルとJavaScriptファイルが出力されます。

```
npm run dev
```

Bootstrap

今回は実行する必要はありません。あくまで紹介となります。

```
composer require laravel/ui
```

```
php artisan ui bootstrap --auth
```

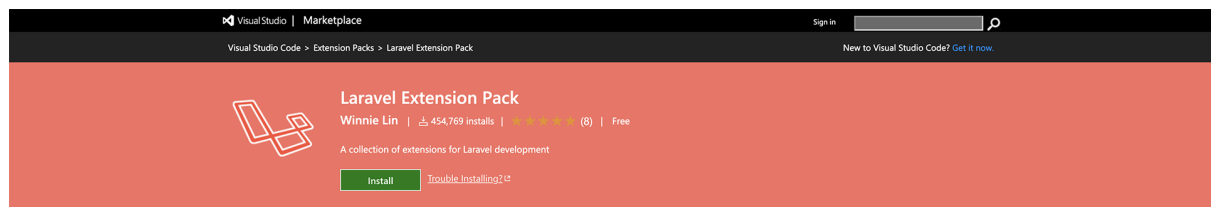
```
npm install
```

```
npm run dev
```

VSCoodeの拡張機能

PHP Intelephense

Laravel Extension Pack for Visual Studio Code



Laravelでの開発をサポートしてくれるいくつかの拡張機能がまとめられています。VSCoodeの拡張機能から「**Laravel Extension Pack**」を検索して、インストールします。

Laravel IDE helper

コマンドプロンプトで、LaravelプロジェクトフォルダでIDEヘルパー（_ide_helper.php）と（_ide_helper_models.php）をインストールするコマンドを実行しましょう。

```
composer require --dev barryvdh/laravel-ide-helper
```

```
php artisan ide-helper:generate
```

```
php artisan ide-helper:models -N
```