

2018年8月20日  
株式会社インプレスR&D  
<https://nextpublishing.jp/>

Vue.js と Nuxt.js でユニバーサルアプリを構築！

## 『Hello!! Nuxt.js』発行

技術書典シリーズ、8月の新刊

インプレスグループで電子出版事業を手がける株式会社インプレス R&D は、『Hello!! Nuxt.js』(著者:那須理也)を発行いたします。

### 『Hello!! Nuxt.js』

<https://nextpublishing.jp/isbn/9784844398400>



著者:那須理也

小売希望価格:電子書籍版 1600円(税別)／印刷書籍版 1800円(税別)

電子書籍版フォーマット:EPUB3／Kindle Format8

印刷書籍版仕様:B5判／カラー／本文 102 ページ

ISBN:978-4-8443-9840-0

発行:インプレス R&D

### << 発行主旨・内容紹介 >>

#### 【Vue.js と Nuxt.js でユニバーサルアプリを構築！】

本書は Vue.js でユニバーサルなアプリケーションを構築するためのフレームワークである「Nuxt.js」の入門書です。ユニバーサルアプリを Nuxt.js を使って実現するための手法をわかりやすく解説しています。

#### 〈本書の対象読者〉

Vue.js は使ったことがあるが Nuxt.js は使ったことがない人

Nuxt.js の使い方が今ひとつわからない人

(本書は、次世代出版メソッド「NextPublishing」を使用し、出版されています。)

# Nuxt.js の導入方法を丁寧に解説

## 第2章 Nuxt.jsのはじめ方

本章ではNuxt.jsのプロジェクトのはじめ方を解説していきます。大きく分けると、何も無い状態からnpmを使う方法と、vue-cliというVue.jsのプロジェクトを作るためのコマンドラインツールを使う方法があります。ここではそれぞれについて紹介していきます。

### 2.1 npmを使う

npmを使って、ゼロからNuxt.jsで開発する環境を作ります。まずはnpm initをしてプロジェクトを開始します。

```
$ npm init
```

次にNuxt.jsをインストールします。

```
$ npm install --save nuxt
```

-saveをつけてインストールします。

その後、Nuxt.jsの作法にしたがったディレクトリ構成でディレクトリを作成することで、開発をはじめることができます。ディレクトリ構成に関しては第4章「ディレクトリ構成と役割」で詳しく説明します。

今回は最低限必要なディレクトリであるpagesディレクトリを作成します。

```
$ mkdir pages
```

pagesディレクトリを作成したら、その中に最初のページを作成します。pagesディレクトリの中に配置するのはvueファイルです。vueファイルはtemplate・script・styleをひとつにまとめたファイルで、「単一ファイルコンポーネント」と呼ばれます。index.vueを作成して次のように記述します。

リスト2-1: index.vue

```
<template>
  <div>
    <h1>Hello Nuxt !!</h1>
  </div>
</template>
```

これで最低限の文字列「Hello Nuxt !!」が表示されるコンポーネントが作成されました。次に、Nuxt.jsを実行できるようにpackage.jsonのscriptにNuxt.jsを実行するコマンドを追加します。

リスト2-2: package.json

```
{
  // 略
  "scripts": {
    "dev": "nuxt",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  // 略
}
```

nuxtコマンドはNuxt.jsを開発モードで起動します。コマンドについては第11章「コマンド」で詳しく説明します。

これでnpm scriptでNuxt.jsを実行できるようになります。早速実行してみます。

```
$ npm run dev
```

コマンドを実行したあとhttp://localhost:3000をブラウザで開くと、次のように表示されるはずです。

# Nuxt.js のミドルウェアやプラグイン、モジュールやコマンドについて各1章ずつ詳しく紹介

## 第7章 プラグイン

Nuxt.jsではプラグインを定義することができます。プラグインとはルートのVue.jsアプリケーションがインスタンス化される前に実行されるプラグラムです。主にVue.jsのプラグインを使用するときや、グローバルで使いたいコードや、OSSのVue.jsのプラグインなどを使用する際に使用します。

### 7.1 プラグインとは

プラグインとは、アプリケーションをインスタンス化する前に実行したいプログラムのことです。

グローバルで使いたいコードや、OSSのVue.jsのプラグインなどを使用する際に使用します。

プラグインを作成するときは次のように記述します。

リスト7-1: plugins/my-plugin.js

```
import Vue from 'vue'
// Vue.jsアプリケーションがインスタンス化される前に実行したい処理
```

リスト7-2: nuxt.config.js

```
module.exports = {
  plugins: ['~/plugins/my-plugin.js'] // pluginの場所を指定する。
}
```

### 7.2 OSSのVueプラグインを使用する場合

Vue.jsを使用する際、OSSのプラグインを使用する場合はVue.jsアプリケーションのインスタンスが作成される前にプラグインを登録する必要があります。その場合もNuxt.jsのプラグインの仕組みを使用します。

element-uiを使用する場合の例を次に示します。

```
{
  src: '~/plugins/my-plugin.js',
  ssr: false
}
]
```

以上のように読み込みの設定を行うと、クライアントサイドでのみプラグインの読み込みが可能です。

### 7.5 サーバサイドでのみプラグインを利用したい場合

逆にサーバサイドでのみプラグインの使用をしたい場合もあると思います。その場合はprocess.server変数を使用することでサーバサイドかどうかを判定し分岐することで実現できます。

process.server変数はサーバサイドの場合にのみtrueになります。if文等で分岐をさせることで実現していきます。

リスト7-3: plugins/my-plugin.js

```
export default (obj, inject) => {
  if (process.server) {
    console.log('in server')
  }
}
```

プラグインファイルの読み込み自体は行われてしましますが、これで処理は実現されます。

### 7.6 まとめ

Nuxt.jsのプラグイン機能について紹介しました。Vue.jsのアプリケーションがインスタンス化される前に実行したい処理がある場合や、コンポーネント内で共通して使いたい関数を作成したい場合は活用していくとよいでしょう。

## Nuxt.js でのアプリケーション開発について、ケース別に解説



### <<目次>>

## 第1章 Nuxt.js とは

- 1.1 ユニバーサルアプリ
- 1.2 サーバーサイドレンダリング
- 1.3 静的ファイルジェネレータ
- 1.4 まとめ

## 第2章 Nuxt.js のはじめ方

- 2.1 npm を使う
- 2.2 vue-cli を使う
- 2.3 まとめ

## 第3章 Nuxt.js の設定について

- 3.1 build
- 3.2 css
- 3.3 dev
- 3.4 env
- 3.5 generate
- 3.6 head
- 3.8 modules
- 3.9 plugins
- 3.10 rootDir
- 3.11 router
- 3.12 srcDir
- 3.13 transition

### 3.14 まとめ

## 第4章 ディレクトリ構成と役割

### 4.1 pages

### 4.2 components

### 4.3 layouts

### 4.4 plugins

### 4.5 middleware

### 4.6 store

### 4.7 assets

### 4.8 static

### 4.9 まとめ

## 第5章 ページコンポーネント

### 5.1 コンテキスト

### 5.2 .vue ファイルに追加されたオプション

### 5.3 まとめ

## 第6章 レイアウト

### 6.1 デフォルトレイアウト

### 6.2 エラーページ

### 6.3 カスタムレイアウト

### 6.4 まとめ

## 第7章 プラグイン

### 7.1 プラグインとは

### 7.2 OSS の Vue プラグインを使用する場合

### 7.3 アプリケーションのルートや context に挿入する

### 7.4 クライアントサイドでのみプラグインを利用したい場合

### 7.5 サーバサイドでのみプラグインを利用したい場合

### 7.6 まとめ

## 第8章 ミドルウェア

### 8.1 ミドルウェアとは

### 8.2 ミドルウェアを実装する

### 8.3 まとめ

## 第9章 ストア

### 9.1 ストアの使い方

### 9.2 プラグインの作成

### 9.3 まとめ

## 第10章 モジュール

### 10.1 OSS のモジュールを使用する

### 10.2 モジュールの作成方法

### 10.3 まとめ

## 第11章 コマンド

### 11.1 nuxt

### 11.2 nuxt build

### 11.3 nuxt start

### 11.4 nuxt generate

### 11.5 まとめ

## 第12章 Nuxt.jsでのWebアプリケーション開発

### 12.1 Nuxt.jsをフロントエンドサーバーとして使う方法

### 12.2 静的ファイルジェネレータで出力したものをホスティングサービスで利用する

### 12.3 expressのミドルウェアとして使用する場合

### 12.4 まとめ

## 付録A Nuxtバージョン2について

### A.1 webpackがバージョン4になる

### A.2 nuxt.config.jsの設定でvendorオプションがなくなる

### A.3 buildオプションにsplitChunksが追加される

### A.4 nuxt.config.js内でES Moduleが使用できるようになる

### A.5 その他ブレーキングチェンジ

### A.6 まとめ

## <<著者紹介>>

那須 理也

Webアプリケーションエンジニア。大学卒業後、老舗ソフトハウスに3年間勤務し、クラウドソーシングサービスを提供する企業に転職。そこで主に Ruby on Rails を活用しサービス開発を行っている。JavaScript で作る動きのあるサービス開発が好みだが、最近の仕事はインフラ業務多め。多趣味。著書に「Hello! Vue.js」(インプレス R&D)がある。

## <<販売ストア>>

電子書籍:

Amazon Kindle ストア、楽天 kobo イーブックストア、Apple iBookstore、紀伊國屋書店 Kinoppy、Google Play Store、honto 電子書籍ストア、Sony Reader Store、BookLive!、BOOK☆WALKER

印刷書籍:

Amazon.co.jp、三省堂書店オンデマンド、honto ネットストア、楽天ブックス

※ 各ストアでの販売は準備が整いしだい開始されます。

※ 全国の一般書店からもご注文いただけます。

## 【株式会社インプレス R&D】 <https://nextpublishing.jp/>

株式会社インプレス R&D (本社: 東京都千代田区、代表取締役社長: 井芹昌信) は、デジタルファーストの次世代型電子出版プラットフォーム「NextPublishing」を運営する企業です。また自らも、NextPublishing を使った「インターネット白書」の出版など IT 関連メディア事業を展開しています。

※NextPublishing は、インプレス R&D が開発した電子出版プラットフォーム(またはメソッド)の名称です。電子書籍と印刷書籍の同時制作、プリント・オンデマンド(POD)による品切れ解消などの伝統的出版の課題を解決しています。これにより、伝統的出版では経済的に困難な多品種少部数の出版を可能にし、優秀な個人や組織が持つ多様な知の流通を目指しています。

## 【インプレスグループ】 <https://www.impressholdings.com/>

株式会社インプレスホールディングス(本社: 東京都千代田区、代表取締役: 唐島夏生、証券コード: 東証1部 9479) を持株会社とするメディアグループ。「IT」「音楽」「デザイン」「山岳・自然」「旅・鉄道」「学術・理工学」を主要テーマに専門性の高いメディア&サービスおよびソリューション事業を展開しています。さらに、コンテンツビジネスのプラットフォーム開発・運営も手がけています。

## 【お問い合わせ先】

株式会社インプレス R&D NextPublishing センター

TEL 03-6837-4820

電子メール: [np-info@impress.co.jp](mailto:np-info@impress.co.jp)