

Nuxt.jsはローカルにNode.jsを入れてあげれば動かすことができますが、ローカル環境をできるだけ汚したくないので、***docker-compose***で動かします。特にNode.jsはバージョン差異が結構きついで。。

構成

- Vue.js v2.6.14
- Nuxt.js v2.15.7
- コンポーネント設計
 - compositionApi
 - Nuxt.js用のプラグイン(@nuxtjs/composition-api)を使用
- TypeScript
- Vuetify(UIフレームワーク)

環境構築手順

コンテナでNode.jsのバージョン確認まで

- nodeのイメージを使用する
- 作成したnodeのイメージに、yarnを用いて使用するモジュールをインストールしていく

```
FROM node:lts

EXPOSE 3000
ENV HOST 0.0.0.0

RUN apt-get update && \
    apt-get upgrade -y && \
    yarn global add @vue/cli && \
    yarn global add @vue/cli-init
```

```
version: '3'
services:
  nuxt-app:
    build: . # Dockerfileの場所
    working_dir: /src # ホストOSとコンテナ内でソースコードを共有する
    volumes:
      - ./src
    ports:
      - 5000:3000 # コンテナ内部の3000を外部から5000でアクセスする
    tty: true # コンテナ起動後、StatusがExitにならないようttyの設定
```

ビルド実行

```
docker-compose up -d
```

コンテナを起動。バックグラウンドで起動します。

```
docker-compose build
```

起動したコンテナにアタッチ。Node.jsのバージョンを確認

```
docker-compose exec nuxt-app bash
$ node -v
$ yarn -v
```

Nuxt.jsの導入

Nuxt.jsの導入については、を参考に実行しました。

yarn / npm / npxのそれぞれでプロジェクトを開始する方法が記載されています。

今回は、yarnを使用します。

```
yarn create nuxt-app <project-name>
```

nuxtのプロジェクト作成が始まり、何を導入するのか（packageやモジュールなど）質問されるので、必要なものを選択します。

この段階でモジュールのインストールを選択しなかったものについても、プロジェクト作成後に追加できます。

質問されるモジュールの内容と今回私が導入したものを紹介します。

- Project name
任意
- package manager
 - npm
 - yarnを選択
- Programing language
 - TypeScriptを選択
 - JavaScript
- UI Framework
 - Vuetify.jsを選択
 - Tailwind CSS
 - その他UIフレームワーク
- Nuxt.js modules
 - Axios 選択
 - Progressive Web App(PWA) 選択
 - content(Git based headless CMS)

- Linting tools
 - ESLint 選択
 - Prettier 選択
 - StyleLint 選択
- Testing Framework
 - Jestを選択
 - AVA
 - WebdriverIO
 - Nightwatch
- Rendering mode
 - Universal(SSR/SSG)を選択
 - Single Page App(SPA)
- Deploy target
 - Server(Node.js hosting)
 - Static(Static/Jamstack hosting)を選択
- Development tool
 - jsconfig.jsonを選択
- Continuous integration
 - GitHub Actionsを選択
 - TravisCI
 - CircleCI
- Github Name
- Vergion controll system
 - Gitを選択

ローカルサーバを立ち上げる

Nuxtプロジェクトの作成が終了すると（結構時間がかかります。15分ほどかかった気がする）、ローカル上でプロジェクト作成の動作確認を行います。

```
($コンテナ内) cd <project>  
($コンテナ内) yarn dev
```

yarn devでエラーが発生

下記のエラーが発生。

今回、UIフレームワークにVuetifyを導入したのですが、Vuetifyで使用するモジュールが一部不足していたようです。

コンテナ内で

```
vue add vuetify
```

を実行し、再度

```
yarn dev
```

localhost:5000にアクセスし、以下のような画面が表示されていればプロジェクトの作成は完了です。

@nuxtjs/composition-apiの導入

vue2系でcompositionAPIを使用するため、@nuxtjs/composition-apiモジュールの追加を行います。

```
yarn add @nuxtjs/composition-api
```

最後に

Docker環境が準備できれば、Nuxtを導入するときにいろんなモジュールを試せたりできそうですね。
(この記事を書くにあたって、数回環境作りなおしましたw)