

[ホーム](#)[タイムライン](#)[トレンド](#)[質問](#)[公式イベント](#)[公式コラム](#)[募集](#)[Organizati](#)

Qiitaユーザアンケートご協力のお願い/抽選で20名様にAmazonギフト券10,000円分プレゼント



@kashimuuuuu

投稿日 2021年09月19日 更新日 2021年10月02日

DockerでReact+TypeScriptの環境を構築する手順

環境構築, TypeScript, dockerfile, React, docker-compose

なぜこの記事を書いたのか

ReactとTypeScriptは相性が良いといわれており、この二つを組み合わせで開発する人も多いため、Dockerfile と docker-compose.ymlを一から作成して環境構築する方法を記事としてまとめておきたかった。

環境構築完了（ゴール）の確認

この記事のゴールは以下のようにReactのページが起動すること。



環境構築の手順

- ①Docker、nodeのインストール
- ②Dockerfile、docker-compose.ymlの作成
- ③ビルド
- ④TypeScriptで動くReactアプリの作成
- ⑤コンテナの実行

①Docker、nodeのインストール

- Dockerのインストール <https://www.docker.com>

✓ Dockerはコンテナ型の仮想環境。

Dockerをインストールして、Dockerの環境の中で開発をすることになる。

- nodeのインストール <https://nodejs.org/ja/download/>
- ✔ Dockerで環境構築したり、ライブラリをインストールする為にnodeが必要になる。

②Dockerfile、docker-compose.ymlの作成

docker-react-ts-app（プロジェクトフォルダ）

```
├── Dockerfile
└── docker-compose.yml
```

上記のフォルダ構成を参考に、ファイルを作成する。

- ✔ 「docker-react-ts-app」というフォルダを作成し、その直下に「Dockerfile」「docker-compose.yml」という名称で新規ファイルを作成すればOK。

Dockerfile

```
FROM node:14.17.5

ENV LANG=C.UTF-8
ENV TZ=Asia/Tokyo

WORKDIR /usr/src/app
```

- ターミナルで「node -v」を実行することでnodeのバージョンを確認できる。バージョンを「FROM node:〇〇.〇〇.〇」という形で指定する。
- 「ENV LANG=C.UTF-8」は環境変数LANGにC.UTF-8を設定することを意味する。Cは最低限の英語表記を表す。日本語を指定する場合はja_JPだが文字化けを起こすことがある。（参考 <https://eng-entrance.com/linux-localization-lang>）
- 「ENV TZ=Asia/Tokyo」はタイムゾーンをAsia/Tokyoに設定する。

docker-compose.yml

docker-compose.yml

```
version: '3'

services:
  frontend:
    build: .
    environment:
      - NODE_ENV=development
    volumes:
      - ./:/usr/src/app
    command: sh -c 'cd frontend && yarn start'
    ports:
      - '3000:3000'
    tty: true
```



ンによって、docker-compose.ymlの書き方が異なる。

- servicesはアプリケーションを動かすための各要素をネストさせて定義する。上記 docker-compose.yml の場合はfrontendというServiceを定義している。（参考 <https://qiita.com/yuta-ushijima/items/d3d98177e1b28f736f04> ）
- commandは「docker-compose up -d」したとき、frontendディレクトリに移動してdevサーバーを起動する。
- ttyはコンテナが起動してすぐ終了してしまうのを防ぐ。

③ビルド

「③ビルド」「④Reactアプリの作成」「⑤コンテナの実行」で実行する「docker-compose」コマンドは、Dockerfileが存在するディレクトリで実行する必要がある。ターミナルにて「cd docker-react-ts-app」を実行してディレクトリを移動しておく必要がある。

このコマンドを実行してビルドする。

```
docker-compose build
```

※buildコマンドはimageを作成する。コンテナは作成しない。

④ TypeScriptで動く Reactアプリの作成

```
docker-compose run --rm frontend sh -c 'npx create-react-app frontend --template typescr
```

上記「①Docker、nodeのインストール」でnpmをインストールしていれば「npx create-react-app frontend --template typescript」が実行され、TypeScriptで動くReactアプリが作成される。生成されるファイルもjsxファイルではなくtsxファイルとなる。

⑤ コンテナの実行

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

ここまで完了したらWebブラウザで「<http://localhost:3000>」へアクセスする。以下のようにReactのページが起動すれば環境構築完了となる。



docker-compose upに失敗する場合

compose upやcompose downを繰り返していると、いくつもコンテナが立ち上がってしまい、docker-compose upに失敗することがあります。その場合は起動中のコンテナを停止したりするなどしてみてください。（参考 <https://qiita.com/musatarosu/items/31d6293a93e75ca6073e>）

参考

- [Docker上にReact+TypeScriptの環境を構築](#)

Dockerで環境を構築する手順の記事

- [Reactの環境構築](#)
- React + TypeScriptの環境構築 📌 [今ここ](#)



kashi

@kashimuuuuuu

React+TypeScriptメインで開発するWebエンジニア React / Next.js / Golang / JavaScript / TypeScript / GraphQL / Nest.js / PHP / VB.NET / ASP.NET / C / SQL / PLSQL

フォロー






コメント

この記事にコメントはありません。

h 投稿する

編集 プレビュー ? 😊

テキストを入力

 画像を選択 0B / 100MB 投稿

記事投稿イベント開催中



Go強化月間～開発する上で知っておくべき知見を共有しよう～

2022/08/15 - 2022/09/15

<https://qiita.com/kashimuuuuu/items/b5f35057dfe1980d053a>

[イベント詳細を見る](#)

エンジニア夏休み企画！～自由研究や読書感想文を発表しよう～

2022/08/15~2022/09/16

[イベント詳細を見る](#)[👉 すべて見る](#)

How developers code is here.

© 2011-2022 Qiita Inc.

About	リリースノート	🐦 Qiita（キータ）公式
利用規約	公式イベント	🐦 Qiita マイルストーン
プライバシーポリシー	公式コラム	🐦 Qiita 人気の投稿
ガイドライン	募集	📘 Qiita（キータ）公式
デザインガイドライン	アドベントカレンダー	
ご意見	Qiita 表彰プログラム	
ヘルプ	API	
広告掲載		
Qiita 関連サービス	運営	
Qiita Team	運営会社	
Qiita Jobs	採用情報	
Qiita Zine	Qiita Blog	
Qiita 公式ショップ		