

# *Narration&Reference* Docker用VS Code拡張機能の使い方

Base File Name: NarrationReference\_Docker\_VSCodeExtension\_Angular7\_ja

2018.11.14

## *Docker*用VS Code拡張機能の使い方

by *Shuichi Ohtsu*

---

### 動作環境

今回は、Visual Studio CodeのDocker用拡張機能をご紹介します。

また、この拡張機能を使用しながら、Angular7のDockerイメージを作成していきます。

サーバとしては、Angularに内蔵されているサーバではなく、Alpine Linuxで稼働するNginxを使用します。

このような環境を配慮し、今回はDocker for Windowsではなく、WSL(Windows Subsystem for Linux)上のUbuntuで稼働するDockerを利用することにします。

なお、WSLにまだDockerのインストールをされていない方は、別のビデオを参照してください。

---

### Visual Studio CodeのDocker用拡張機能のインストール

まず、WSL上のUbuntuを開きます。

次に、コマンド・ラインで、`code .`と入力してVisual Studio Codeを開きます。

---

VS Codeが開きましたら、左側の拡張機能アイコンをクリックし、検索ボックスにDockerと入力します。

---

すると検索結果リストにMicrosoft社製のDockerという拡張機能が表示されますので、インストールをクリックします。

---

Dockerの解説ページが表示されましたら、再読み込みしてアクティブにするをクリックします。

これでDocker拡張機能が有効になりました。

ここで一旦VS Codeを閉じます。

---

## テスト用Angular7の生成

ここで、テスト用のAngular7のプロジェクトを作成します。

最初に稼働環境をチェックすることになります。

まずコマンド・ラインで、`ng --version`と入力して、AngularのバージョンおよびNode.jsのバージョンを確認します。

Angularのバージョンは、7.0であり、Node.jsのバージョンは、10.12です。このNode.jsのバージョンは後で重要になりますので、銘記しておいてください。

---

次に、Angular7のテスト用プロジェクトを作成します。

`ng new ng7-initial`と入力します。

`ng7-initial`はプロジェクト名であり、別の名称でも構いません。

---

まず、ルーティング機能を追加するか否かの問いがありますので、ここではYとしておきます。

次にスタイルシート・フォーマットの問いがありますので、ここでは、`scss`を選んでみます。

するとインストールが開始されます。

このプロセスには若干時間を要します。

---

インストールが終了しましたら、そのプロジェクト・ディレクトリに移動し、`code .`と入力し、Visual Studio Codeを起動します。

---

VS Codeが起動しましたら、`control + @`キーでコマンド・ライン・ウィンドウを開きます。

そして、`ng s -o`と入力し、ローカル・サーバを起動します。

---

Angularのデフォルトのページが表示されました。

OKです。

---

VS Codeに戻り、`control + c`でローカル・サーバを停止します。

---

## Dockerファイルの作成

次に、このプロジェクトをAlpine Linuxで稼働するNginx上にコピーする*Dockerfile*を作成します。

このDockerfileの構成は、2段階になっています。

まず、Node.js上でAngular7をインストールし、そこにこのプロジェクトをプロダクションモードでコンパイルし、配布用圧縮JavaScriptを生成します。

次に、Alpine Linux上にNginxを生成し、そのホームページ・ディレクトリに、必要Node.jsとともに、この圧縮JavaScriptをコピーします。

---

ここでは、Nginxで稼働したページであることを明確にするために、デフォルト・ページの内容も若干変更します。

---

## Dockerイメージの作成

次にDockerイメージの作成になりますが、*Docker* 拡張機能のおかげで、この操作は非常に簡単になりました。

VS CodeでDockerfileを表示し、その中で右クリックし、ポップアップメニューを表示します。

そして、Build Imageを選択するだけです。

すると生成するイメージの名称の問い合わせがありますので、ここでは、ng7-initial:1.0としておきます。

: 以下はバージョンを指定します。

すると生成が開始されます。

このプロセスには若干時間を要します。

---

途中でエラーが発生しました。

エラー・メッセージによると、最新のNode.jsでは、スタイル・フォーマットで、SCSSを選択した場合、問題があるようです。

そこで、最初に起動をチェックした、Node.jsのバージョンに変更します。

そのバージョンは、10.12です。

Dockerファイルを修正し、再度Dockerイメージの生成をします。

今回は、OKです。

---

## Dockerイメージの起動

ここで、Dockerイメージが生成されたかをチェックします。

VS Codeの左側のDockerアイコンをクリックします。

そして、Imagesを選択します。

すると確かに、ng7-initialが生成されているのがわかります。

---

次に、このイメージをローカル上で起動してみることにします。

---

VS Code上のコマンド・ラインから、`docker run --rm -d -p 80:80 ng7-initial:1.2`と入力します。

指定している各オプションは

--rm はコンテナ終了時にコンテナ自動的に削除

-d コンテナをバックグラウンドで実行

-p 外部からアクセスされるポート番号:コンテナ側のポート番号を指定

を意味しています。

Nginxは、内部で80ポートを使用していますので、それをそのまま外部に対して、80ポートでアクセス可能にしているという意味になります。

---

ここで、ブラウザで稼働を確認します。

外部に対して、80ポートでアクセス可能にしていますので、

`http://localhost` そのままでアクセス可能なはずです。

---

Docker上のAngular7およびNginxの稼働を確認することができました。

OKです。

---

ご清聴ありがとうございました。

---

## Reference

- "Docker Community Edition for Windows",  
<https://store.docker.com/editions/community/docker-ce-desktop-windows>
- "Docker/Kubernetes 実践コンテナ開発入門",  
<http://amazon.co.jp/o/ASIN/4297100339/>
- "プログラマのためのDocker教科書 第2版 インフラの基礎知識&コードによる環境構築の自動化",  
<http://amazon.co.jp/o/ASIN/4798153222/>

- "Containerizing Angular with Docker - Dan Wahlin",  
<https://www.youtube.com/watch?v=cLT7eUWKZpg&t=1140s>
- "Deploy Angular 5 app in Docker Container in under 10 mins - For local development",  
<https://www.youtube.com/watch?v=L2UkQ2CND68&t=178s>
- "Angular5, Angular6, Angular7 Custom Library: Step-by-step guide",  
<https://www.udemy.com/angular5-custom-library-the-definitive-step-by-step-guide/>
- "Angular5, Angular6, Angular7用 カスタムライブラリの作成: 完全ステップ・バイ・ステップ・ガイド",  
<https://www.udemy.com/angular5-1/>