

# mean-stack (0.0.1)

Maksim Kostromin

Version 0.0.1, 2018-06-16 23:10:40 UTC

# Table of Contents

1. Introduction .....	2
2. Implementation .....	3
2.1. frontend .....	3
3. Docker .....	4
3.1. frontend .....	4
4. Heroku deployment .....	5
4.1. frontend .....	5
4.1.1. prepare nginx in docker for heroku .....	5
4.1.2. prepare heroku config .....	6
4.1.3. create and deploy heroku app .....	6
5. Links .....	7

Travis CI status:

# Chapter 1. Introduction

[Read reference documentation](#)

- Initial setup: Angular 6 (frontend)
- TODO: Express + MongoDB (backend)

# Chapter 2. Implementation

## 2.1. frontend

*add material design to project:*

```
ng add @angular/material
```

*configure material design as separated module (app-angular-material.module.ts file):*

```
// import { NoopAnimationsModule } from '@angular/platform-browser/animations';
import { BrowserAnimationsModule } from '@angular/platform-browser/animations';

import { MatToolbarModule } from '@angular/material';

const modules = [
  // NoopAnimationsModule,
  BrowserAnimationsModule,
  MatToolbarModule,
];

@NgModule({
  imports: modules,
  exports: modules,
})
export class AppAngularMaterialModule { }
```

*configure application routes (app-routing.module.ts file):*

```
import { Routes, RouterModule } from '@angular/router';
import { CreateComponent } from './components/create/create.component';
import { EditComponent } from './components/edit/edit.component';
import { ListComponent } from './components/list/list.component';

const routes: Routes = [
  { path: 'create', component: CreateComponent },
  { path: 'edit/:id', component: EditComponent },
  { path: 'list', component: ListComponent },
  { path: '**', redirectTo: 'list', pathMatch: 'full' },
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule { }
```

# Chapter 3. Docker

## 3.1. frontend

*build frontend application*

```
npm i  
npm run dist
```

*build docker image based on nginx*

```
docker build -f Dockerfile.frontend -t daggerok/mean-stack:frontend ./frontend
```

*run application in docker*

```
docker run -d --rm -p 80:80 -e "PORT=80" --name run-mean-stack-frontend daggerok/mean-stack:frontend
```

*check application is working on [0.0.0.0/](http://0.0.0.0/)*

```
open http://127.0.0.0/
```

*finally stop docker container for cleaning up...*

```
docker stop run-mean-stack-frontend
```

# Chapter 4. Heroku deployment

## 4.1. frontend

### 4.1.1. prepare nginx in docker for heroku



Idea is simple: as far nginx doesn't support environment variables, but heroku is required ENV `$PORT` to be used, we can create nginx default.conf template and during build or bootstrap generate and replace it in nginx. see `Dockerfile.frontend` for details

prepare frontend Dockerfile (`Dockerfile.frontend` file):

```
FROM nginx:1.15.0-alpine
LABEL MAINTAINER="Maksim Kostromin https://github.com/daggerok"
ARG PORT_ARG="80"
ARG HOST_ARG="localhost"
ENV PORT=${PORT_ARG}
ENV HOST=${HOST_ARG}
RUN apk add --update --no-cache libintl \
    && apk add --no-cache --virtual build_deps gettext \
    && cp /usr/bin/envsubst /usr/local/bin/envsubst \
    && apk del build_deps \
    && echo '#!/bin/sh' >> /usr/local/bin/entrypoint.sh \
    && echo 'PORT=${PORT:=80}' >> /usr/local/bin/entrypoint.sh \
    && echo 'HOST=${HOST:=localhost}' >> /usr/local/bin/entrypoint.sh \
    && echo '#/bin/sh -c "envsubst < /root/default.conf.tpl >'
/etc/nginx/conf.d/default.conf && exec nginx -g \"daemon off;\"\"\" >>
/usr/local/bin/entrypoint.sh \
    && echo '/bin/sh -c "envsubst < /root/default.conf.tpl >'
/etc/nginx/conf.d/default.conf && (cat /etc/nginx/conf.d/default.conf | grep listen)
&& (cat /etc/nginx/conf.d/default.conf | grep server_name) && exec nginx -g \"daemon
off;\"\"\" >> /usr/local/bin/entrypoint.sh \
    && chmod +x /usr/local/bin/entrypoint.sh
ENTRYPOINT entrypoint.sh
CMD /bin/sh
COPY ./docker/default.conf.tpl /root/default.conf.tpl
COPY ./frontend/dist/frontend /usr/share/nginx/html
```

prepare nginx config template (docker/default.conf.tpl file):

```
server {
    listen      ${PORT};
    server_name ${HOST};
    charset     utf-8;

    location / {
        root    /usr/share/nginx/html;
        index   index.html index.htm;
        allow   all;
    }

    # redirect some server error pages to the fallback page /index.html
    error_page 400 404 500 502 503 504 /index.html;
    location = /index.html {
        root    /usr/share/nginx/html;
        allow   all;
    }
}
```

#### 4.1.2. prepare heroku config

prepare heroku.yml file:

```
build:
  docker:
    frontend-build: Dockerfile.frontend-build
```

#### 4.1.3. create and deploy heroku app

create heroku application

```
heroku create daggerok-mean-stack-frontend --stack=container

# output:
Creating    daggerok-mean-stack-frontend... done
https://daggerok-mean-stack-frontend.herokuapp.com/ | https://git.heroku.com/daggerok-mean-stack-frontend.git
```

push app

```
git add .
git commit -am "Deploy heroku app."
git push heroku master
```



# Chapter 5. Links

[GitHub](#)