



dockerfile example-frontend X dockerfile example-backend

example-frontend > dockerfile > ...

```
1  # Install node
2  FROM node
3
4  ENV REACT_APP_BACKEND_URL=http://localhost:8080/
5
6  WORKDIR /usr/src/app
7
8  EXPOSE 5000
9
10 # follow example node install instructions for LTS node 14.x on git
11 RUN curl -sL https://deb.nodesource.com/setup_14.x | bash
12 RUN apt install -y nodejs
13
14 # check install with node -v && npm -v
15 RUN node -v && npm -v
16
17 # Copy all of the content from the project to the image
18 COPY . .
19
20 # install all packages with npm install
21 RUN npm install
22
23 # run the project
24 # build the static files with npm run build to generate them into build folder
25 RUN npm run build
26
27 # Use npm package called serve to serve the project in port 5000
28 RUN npm install -g serve
29
30 #add an user "frontenduser"
31 RUN useradd -m frontenduser
32
33 #change user with the directive USER
34 USER frontenduser
35
36 # CMD echo $REACT_APP_BACKEND_URL
37 CMD ["serve", "-s", "-l", "5000", "build"]
38
```

 dockerfile example-frontend

 dockerfile example-backend 

example-backend >  dockerfile > ...


```
1  # Install golang 1.16
2  FROM golang:1.16
3
4  # REQUEST_ORIGIN to pass an url through the cors check
5  ENV REQUEST_ORIGIN=http://localhost:5000
6
7  WORKDIR /usr/src/app
8
9  # project runs in a docker container with port 8080 published
10 EXPOSE 8080
11
12 COPY . .
13
14 # Run go build. It will generate a binary named "server"
15 RUN go build .
16
17 #add an user "backenduser"
18 RUN useradd -m backenduser
19
20 #change user with the directive USER
21 USER backenduser
22
23 # Execute the file
24 # CMD echo $REQUEST_ORIGIN
25 CMD ["/server"]
26
```



REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
frontend	latest	97b41f80be71	29 minutes ago	1.34GB
backend	latest	338f36102152	34 minutes ago	1.07GB


dockerfile example-frontend X dockerfile example-backend

example-frontend > dockerfile > ...

```
1  # Install node
2  FROM node:alpine
3
4  ENV REACT_APP_BACKEND_URL=http://localhost:8080/
5
6  WORKDIR /usr/src/app
7
8  EXPOSE 5000
9
10 # Copy all of the content from the project to the image
11 COPY . .
12
13 # install all packages with npm install
14 RUN npm install && \
15     # run the project
16     # build the static files with npm run build to generate them into build folder
17     npm run build && \
18     # Use npm package called serve to serve the project in port 5000
19     npm install -g serve && \
20     rm -rf /var/lib/apt/lists/* && \
21     #add an user "frontenduser"
22     adduser -h /home/frontenduser --disabled-password frontenduser
23
24 #change user with the directive USER
25 USER frontenduser
26
27 # CMD echo $REACT_APP_BACKEND_URL
28 CMD ["serve", "-s", "-l", "5000", "build"]
29
30
```

 dockerfile example-frontend

 dockerfile example-backend 

example-backend >  dockerfile > ...

```
1  # Install golang 1.16
2  # FROM golang:1.16
3  FROM golang:alpine
4
5  # REQUEST_ORIGIN to pass an url through the cors check
6  ENV REQUEST_ORIGIN=http://localhost:5000
7
8  WORKDIR /usr/src/app
9
10 # project runs in a docker container with port 8080 published
11 EXPOSE 8080
12
13 COPY . .
14
15 # Run go build. It will generate a binary named "server"
16 RUN go build . && \
17     #add an user "backenduser"
18     adduser -h /home/backenduser --disabled-password backenduser
19
20 #change user with the directive USER
21 USER backenduser
22
23 # Execute the file
24 # CMD echo $REQUEST_ORIGIN
25 CMD ["/server"]
26
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
frontend	latest	99f1445ab185	2 minutes ago	408MB
backend	latest	21c75bbe0514	12 minutes ago	467MB