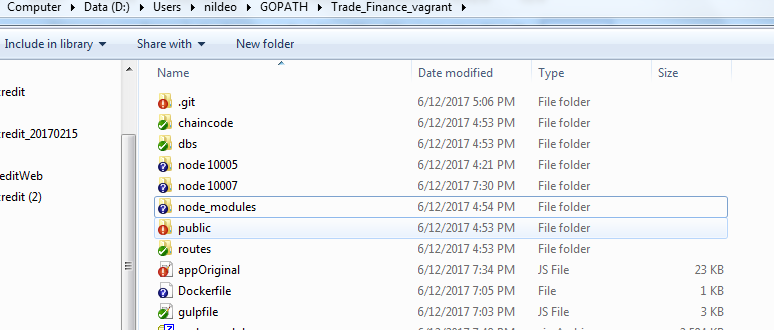
Dockerizing the application :

Take a checkout from the repository



Steps:open docker and Go to the folder where app is checked out



Create one extra file Dockerfile inside the same folder.



Open the file and write the following.

Steps:

* ***vi Dockerfile***
* ***press i***

and paste the following

***FROM node:boron***

***# Create app directory***

***RUN mkdir -p /usr/src/app***

***WORKDIR /usr/src/app***

***# Install app dependencies***

***COPY public /usr/src/app***

***COPY routes /usr/src/app***

***COPY dbs /usr/src/app***

***COPY appOriginal.js /usr/src/app***

***COPY node\_modules /usr/src/app/***

***# Bundle app source***

***COPY . /usr/src/app***

***EXPOSE 10005***

***CMD [ "node", "appOriginal.js" ]***

Exit after saving it

Steps:

Press Escape key

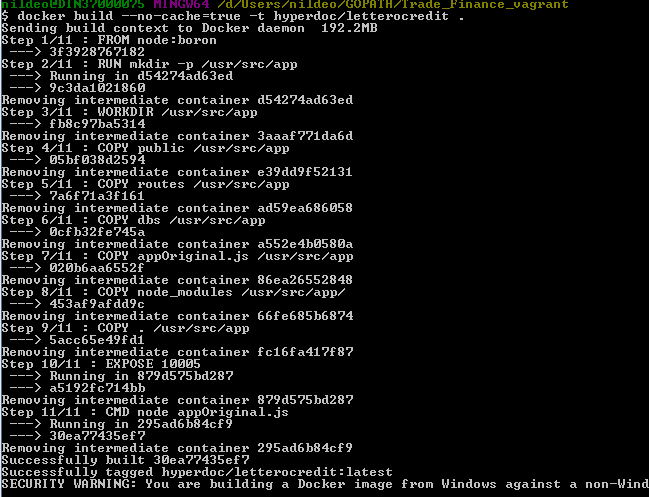
Type the following:

***:wq!***

Build the image and run it using the following steps:

***docker build --no-cache=true -t hyperdoc/letterocredit .***

(this will remove intermediate container and better if changes in the applications need to be replaced while running newly again)



check your images

docker images

run the image in some port of your machine(in docker container the app is running in **10005** and mapped to **10005** on local machine).

docker run -p 10005:10005 -d hyperdoc/letterocredit



get the container id using

***docker ps***



now you can see the logs using ***docker logs <container id>***

ex***: docker logs 119c42db0621***

To debug in docker, run the container in ***attached*** mode using

***docker attach <container\_id>***

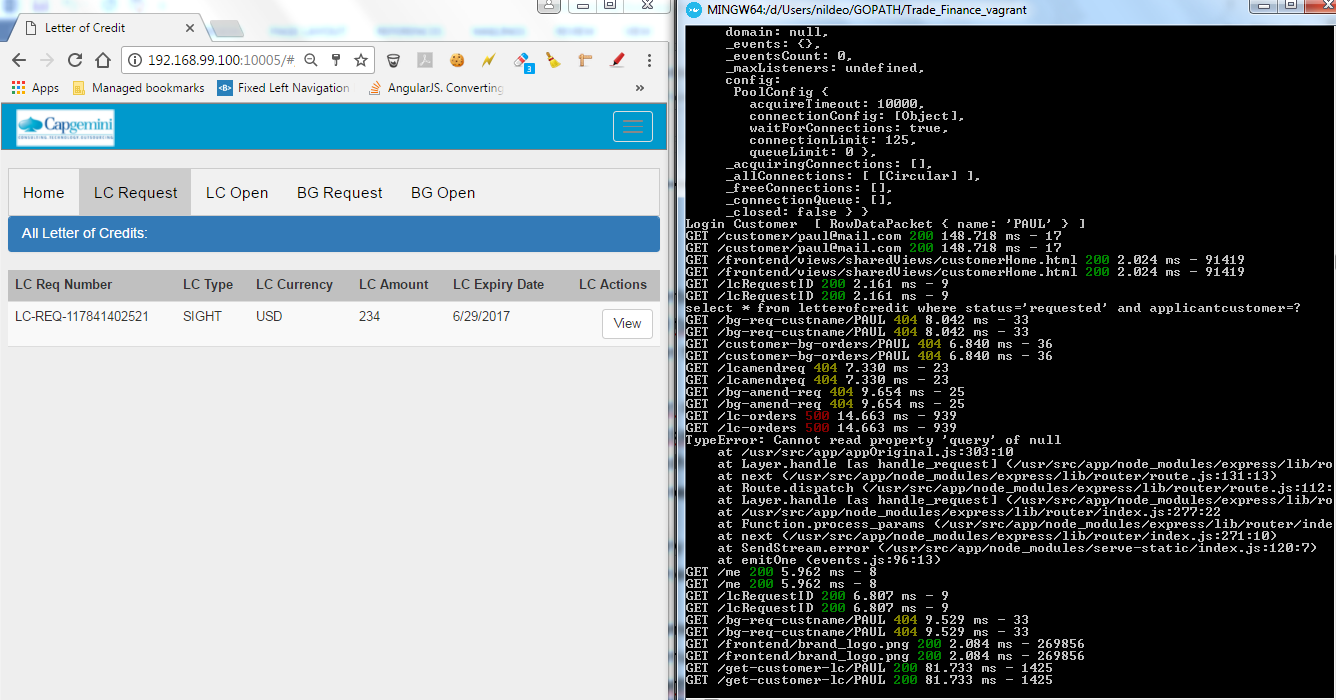


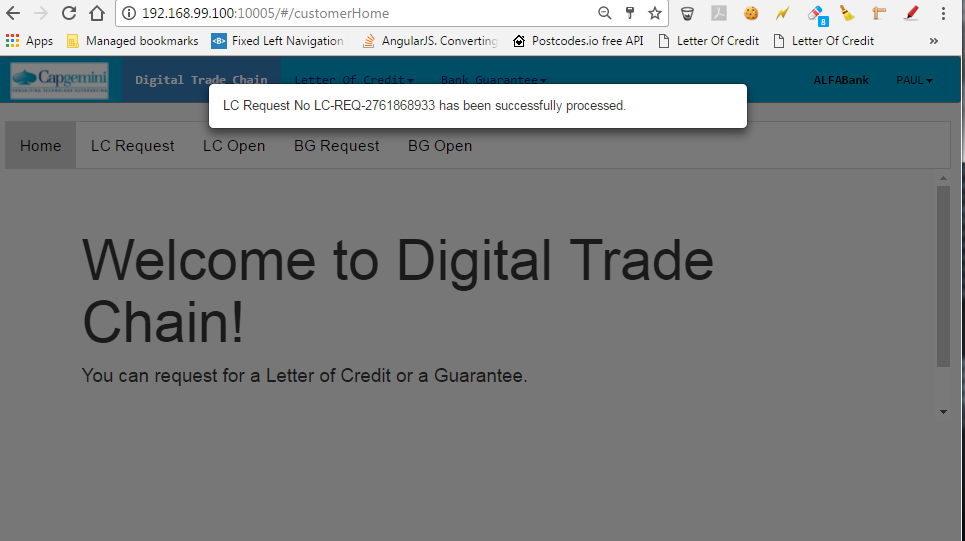
Now give permission to the docker to access mysql db( to get the ip of docker run: ***docker-machine ip default***)

***GRANT ALL PRIVILEGES ON \*.\* TO 'root'@'<ip of docker>' IDENTIFIED BY 'admin';***



Now run the app





For containerizing the other node follow the same steps .

And open the app in 192.168.99.100:10007

Precautions:

Check in the appOriginal file ,the host should be local host/IP of the machine (not Docker)

