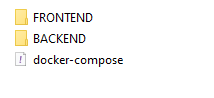
**Docker Hosting Steps**

****

Create docker-compose.yml file (commented example in folder) and in the same directory where you put your front and back end

FrontEnd Part(angular) :

1. Create nginx.conf file inside of your frontend folder ”FRONTEND” in my case (nginx.conf example in the same folder)

Watch this for more info about NGINX <https://www.youtube.com/watch?v=JKxlsvZXG7c&ab_channel=Fireshp>

1. Set up your services links for example you use <http://localhost:backendport/admin/> => then it becomes this

<http://SERVERIPADRESS:backendport/admin>

BackEnd Part(springboot) :

1. Set up your port to which one you want use if you dont its 8080 by default
2. You might have something like this in your application.properties (springboot ressource file )

spring.datasource.url=jdbc:mysql://localhost:3306/etablissement

To

spring.datasource.url=jdbc:mysql://database:3306/etablissement

database => DATABASE container name(the one you used in your docker-compose.yml)

3306 => DATABASE PORT it should be the same in the compose file

etablissement => your database name which is etablissement in my case

Running Steps :

1. First you need to create angular dist file which is created by simply running this command “ ng build “
2. Run “ mvn clean install –DskipTests “ if it says that mvn not recognized try running it from cmd instead of ur IDE terminal, if still giving the same error make sure that you have maven installed and path variables set if not follow this video to set up maven => <https://www.youtube.com/watch?v=RfCWg5ay5B0&ab_channel=CodingMagic>
3. When the command in the second step succeds push everything into your github repo then pull it into a directory in the server (to access server from cmd use the command ssh user@adressip and add the port using –p port if there is any then enter your password)
4. Now that you have the project on the server try browsing to the directory where ur docker-compose file is by using “ cd ” command and run “sudo docker-compose up --build” (refer to the last part for some usefull commands)

Possible Errors :

1. Communication link failure

The cause of this error might be either a mistake in your datasource url or that your spring boot container ran before your database container ran if not make sure that all your ports are in sync or not being already used

1. Username and password not accepted

This error is caused if you are using smtp gmail on an unauthorized port or is occupied , or just simply you are not using the right password or username from the email you are trying to send the message.

Depending on the port for your smtp gmail you use of the configs below.

PORT 465 :

spring.mail.properties.mail.smtp.socketFactory.port = 465

spring.mail.port= 465

spring.mail.properties.mail.smtp.socketFactory.class = javax.net.ssl.SSLSocketFactory

PORT 587 :

spring.mail.port=587

spring.mail.properties.mail.smtp.starttls.enable=true

1. MYSQL associated error like < “column-name” has no default value >

If you get an error of something like this or any other error which is associated to MYSQL and you are sure that everything is right and works on your localhost try shutting down all the containers using the command “sudo docker-compose down” then delete all existing docker images using “sudo docker system prune -a --volumes” this will delete all networks , containers and images then just restart everything using “sudo docker-compose up --build”.

Beware: you might get a communication link failure you just have to run it again cause sometimes your backend container may run before ur database container as I told you before

Usefull commands  and tips:

* sudo docker-compose up --build => running and building the docker containers from the docker compose file
* sudo docker-compose down => shutting down the containers
* sudo docker cotainer ls => see which containers are running and their general infos like name , livingtime, port , …
* If you have more than one spring / angular app just add them the same way for the first ones and it should work simply with their networks and separate containers
* Sometimes you might want to modify your database directly just from the server to access your mysql container :
* sudo docker exec -it container\_name bash
* mysql –u root

Then you will need to enter your database password which you added on ur docker-compose file and app propreties if you don’t have one just click enter without entering everything