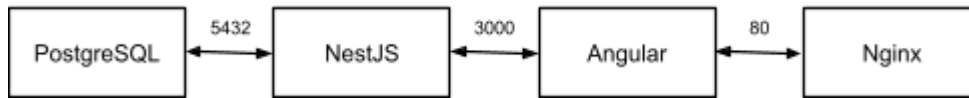


Software Architecture

- Schema of the architecture :



- Our architecture uses Docker with 3 containers : one for the database, one for the backend and one for the frontend and server. Our Nginx container copies the files necessary for the deployment of frontend likewise we save the use of an additional dependency. Our database uses PostgreSQL in a separate container from the backend one as it is more appropriate and make us able to persist the data.

- Services :

- PostgreSQL database
- NestJS backend
- Nginx server
- RabbitMQ administration (for dev mode)

- All of these services are fully implemented.

- Using the app :

- Production mode : `docker compose up --build`
- Development mode : `docker-compose -f docker-compose.dev.yml up --build`
- Supprimer le dossier `node_modules` du backend avec chaque lancement de l'app.

Once it has started :

You can do this command line to make sure there is at least one use in the database :

```
curl -X POST -H "Content-Type: application/json" -d '{"lastname": "admin",  
"firstname": "admin", "age": 0, "password": "admin"}' http://localhost:3000/users
```

Then go to localhost, it will redirect you to localhost/login.

username : 1 and password : amin is an example of credential for connection.

Then you can for example try to add users to see that our system is working.

In development mode it's similar but additionally you have the hot-reloading and you can go to <http://localhost:15672/> with username: user and password: password to administrate RabbitMQ.

- In production mode ports 15672 and 5672 are not open as they are only useful for RabbitMQ in dev mode.