Simon Gomez - Microservices

Documentation

Initiate your documentation

- if not already done, initiate your git repository on github.com
- commit your motus App
- initiate your Readme.md

it should have a least two sections :

- an explanation of what the project do
- a technical explanation on how to RUN the project

TIPS :

- markdown cheatsheet https://www.markdownguide.org/cheat-sheet/

Play with mermaid

go to https://mermaid.live/

Sequence diagram

- Create a sequence diagram explaining all the network request that will happen when someone play with your motus App
- Commit the Readme.md with this version of the diagram inside

Score Management

Document the API you will implement for your scoring system. This API has to be in a dedicated server and should not be implemented in the same codebase as the motus APP. You should be able to document in your Readme the answer to the following questions :

- which server are you gone use ?
- which port are you gone use ?
- which API are you gone call ? which parameters ?
- Can we handle more than one user ?
- What data do we want to store ?

All these information should be in your documentation

- Update your documentation
- Update the previous sequence diagram
- add another digram (graph diagram) which will represent your architecture
- Commit this new documentation

Send me by mail your github repository with the diagrams and the documentation (one mail by team)

Doc Done

Now that the documentation is done,ensure that everything is working correctly.. The scoring app should be a dedicated node server distinct from the first one. the docker-compose file should allow you to deploy and run everything.

TIPS :

- storing data in a database is not mandatory (a flat file is ok) but would be better (depending on your progress on the project)
- having a reverse proxy in your docker-compose, to only expose port 80 is a great idea but it is not mandatory

Microservice pattern

Now be creative ! We have seen a few microservices pattern, pick one or more to implement in your application :

- loadbalancing
- API Gateway
- BFF/BFB
- CQRS
- Service Discovery
- Throttling

Start with the documentation, and then just code.

Published with GitHub Pages