A brief introduction on the architecture

The backend solution is developed in .NET 6. It's constituted from 2 small API with a Domain Driven Development (DDD) architecture to showcase the utility of having a functional problem split in microservices to be scaled depending on the load on each.

- RideAPI is a microservice API with its own database SQLite, wit
- PriceAPI is a stateless microservice that acts a bit like a static singleton to calculate the price. There is a light layer of parameters validation in its controller to filter unwanted values. I designed it in a way it can have its own database in the future where the static values of the price equation can be stored (like the initial fare of 1€, the frequency of tariffication 1/5 and the periods coefficient: normal, busy, night time)

In both APIs you'll find examples of Dependency Injection, Repository pattern and a pseudo-3-tier layer architecture (Controller/Presentation Layer-Busines Service Layer-Data Layer). Again, the price business logic could've been a simple static method in the RideAPI, but with a DDD architecture, the Price has its own functional team that works - with but - autonomously from the Product Team.

If it wasn't for the requirement of having a 2-tier only architecture, I would've split each of the APIs into 3 layers: API, Domain and Infrastructure, plus the Presentation layer in React.

I don't have a lot to say on the React part apart that it contains one component that it renders as many rides as they are.

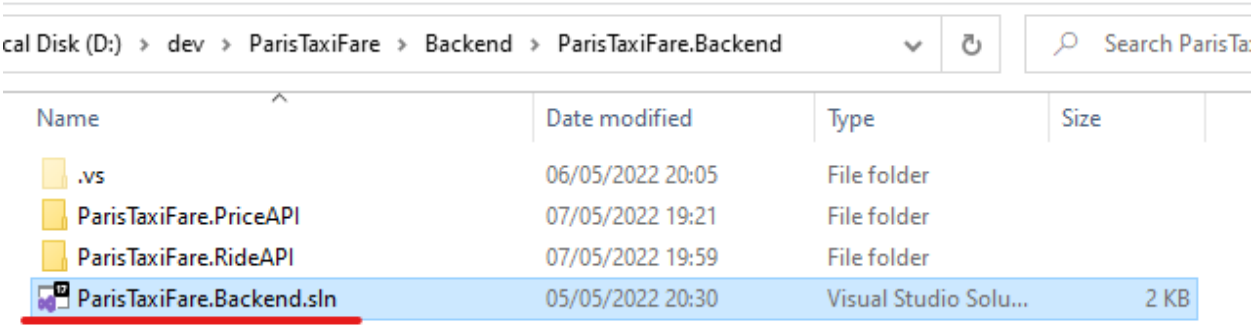
What you'll need installed

- Visual Studio 2022
- .NET 6
- Node.js

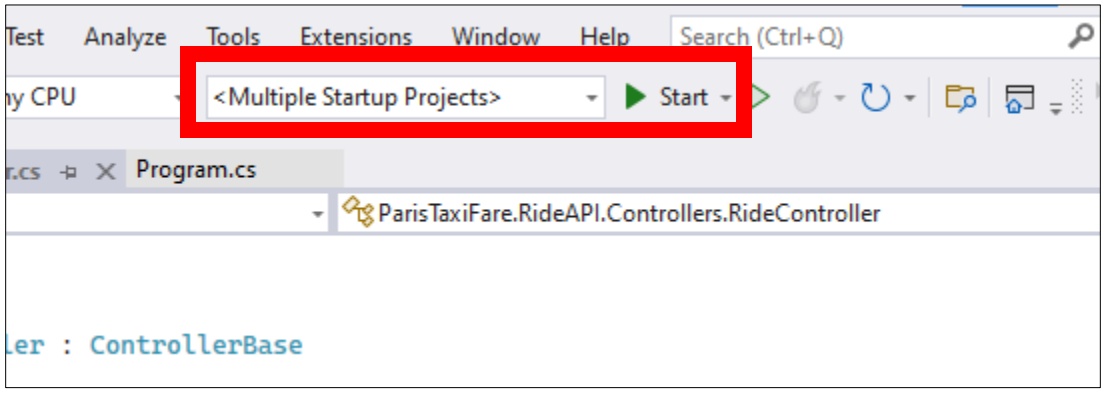
How to run the solution:

The Backend First:

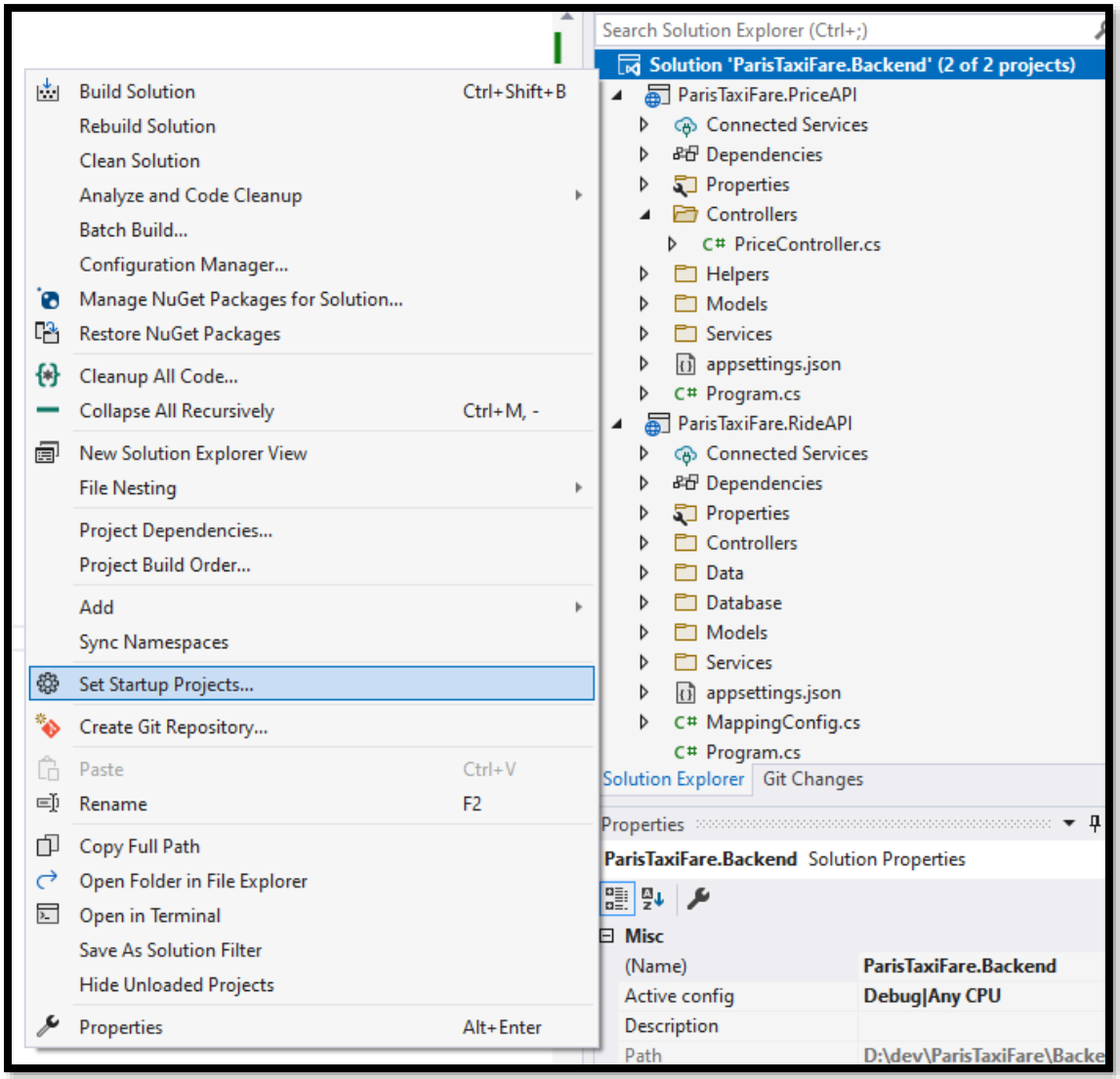
Open the solution with Visual Studio:



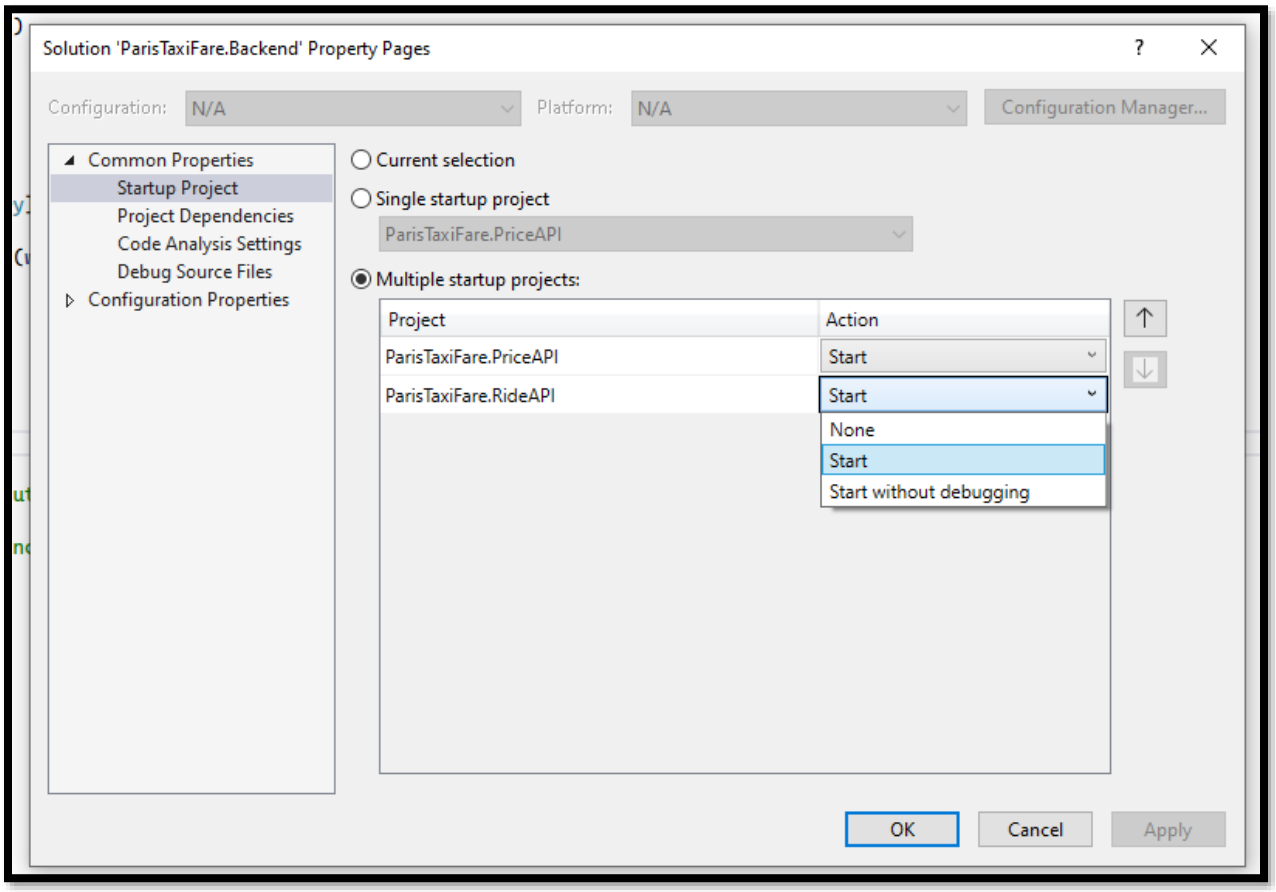
The solution is configured to launch all the APIs at the same time. If VS doesn't launch all the API at the same time, please follow these steps:



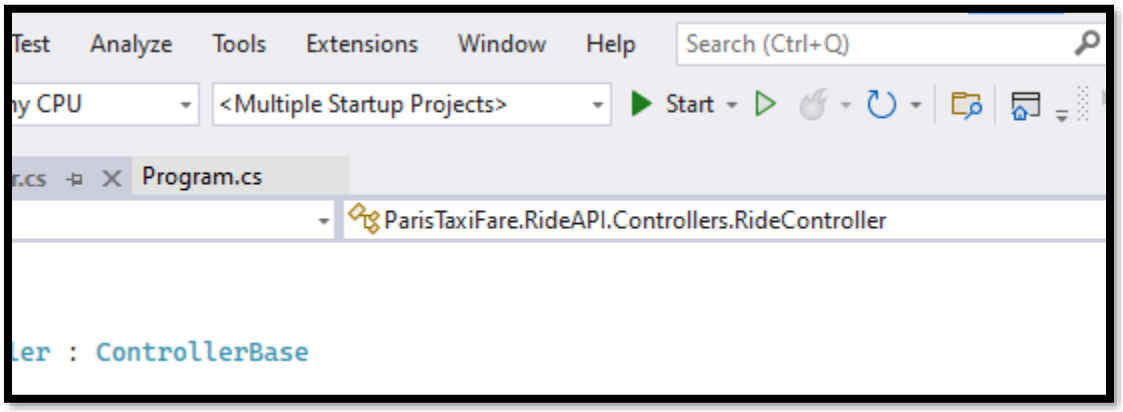
1- Right click on the solution and select "Set Startup Projects…"



2- In the new window, select “Multiple startup projects:” and det both RideAPI and PriceAPI to “Start”



3- Press OK and launch with the Start button. You should see <Multiple Startup Projects> on its left



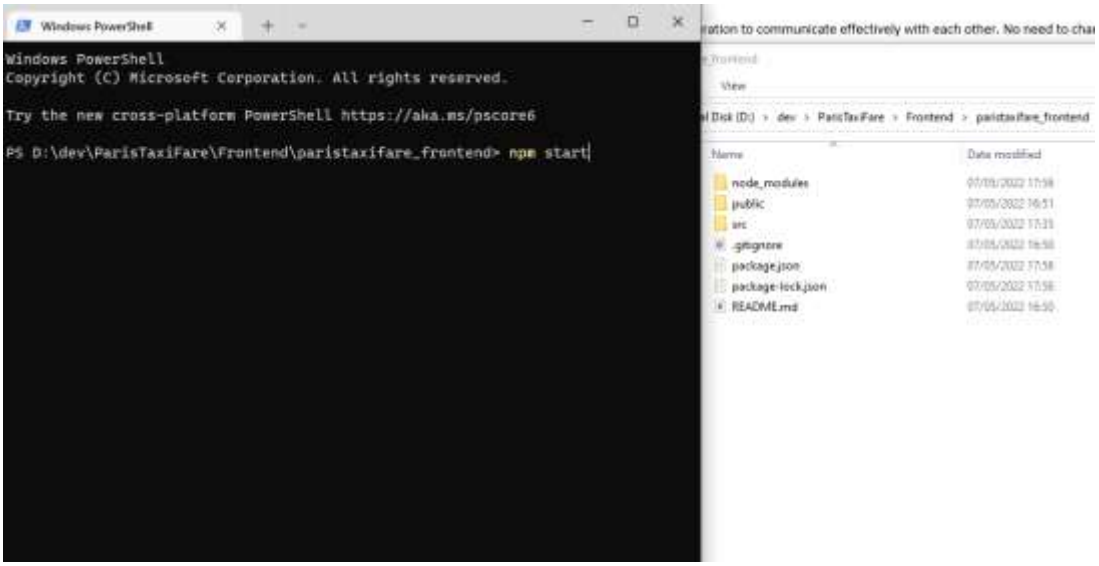
APIs Ports:

The ports are fixed in the configuration to communicate effectively with each other. No need to change or extra configuration:

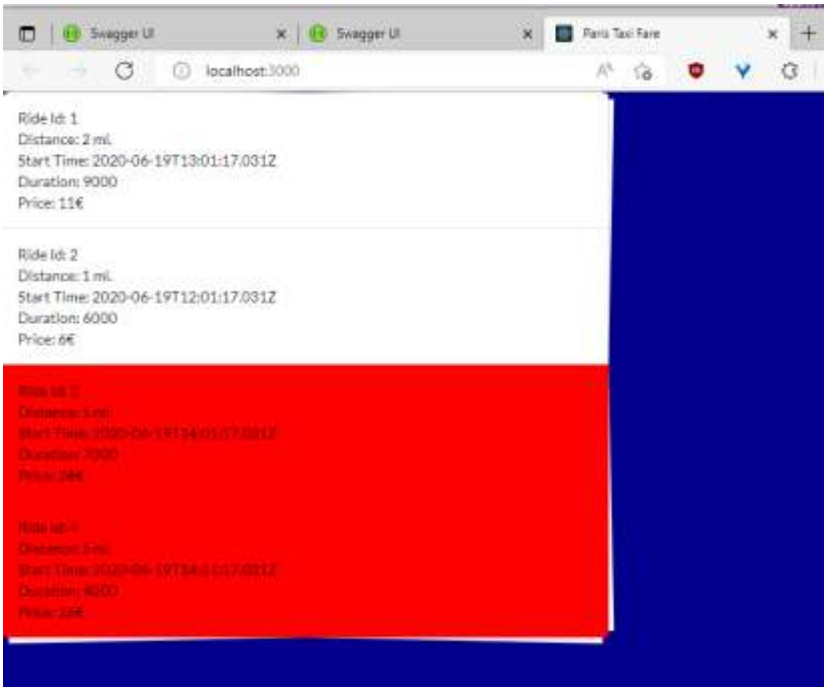
- RideApi is set to `https://localhost:7014`
- PriceApi is set to `https://localhost:7180`

The Frontend

After launching the 2 APIs, open a terminal in the frontend folder and launch the solution with the command: **npm start**



If all the steps are followed, your browser should look like this:



In case of need contact me on alami.khalil.pro@gmail.com or +212669190380

Happy testing!