

## Code Challenge

Technology stack:

- Java 17.
- Postgresql.
- Maven 3.8.3.
- Springboot 3.1.1.

Follow the next steps:

- Clone the following repository: [https://github.com/nontoa/Senior\\_test.git](https://github.com/nontoa/Senior_test.git).
- Run a postgresql instance on your local environment.
- Execute the script called ***senior\_test\_database.sql*** for creating the database and tables.
- Run services antifraud-mock, bank-mock and startup-app.
- For testing, use the postman collection attached called ***SeniorTest.postman\_collection.json***.

Services will be running in the following ports:

- Antifraud-mock: 10983.
- Bank-mock: 10982.
- Startup-service: 8899.

### IMPORTANT NOTES FOR API USE:

- Payment methods available: **VISA, MASTERCARD, AMEX**.
- If you want to simulate a fraudulent transaction, use the following card number **4542759458958822**, if you want to pass the fraud validation use any other valid card number.
- If you want to simulate an approved transaction by the bank, use the csc **777**, otherwise use any other csc for declined transactions.
- If you want to simulate a declined refund, use the reason "**Invalid**", otherwise use any other reason for approving refunds.
- The startup-app is resilient, if the antifraud-mock or bank-mock are down, the service will be working well and will try to connect to them with defined retries. The orders and transactions will always have a consistent state.