# Mighty Block - Backend Developer Test Exercise:

You need to develop the backend API for a single-page application based on the provided mockup ( Mighty-Market.png and Mint-Popup.png). The app consists of a feature-limited *"Opensea"* (NFT Marketplace) clone. It must have the options to mint (create) NFTs and to buy/sell them. Don't worry about managing users, you can suppose there are users provided by another system (hardcode them somewhere and use their id).

For this challenge, we will consider an NFT as an image with a description. Each NFT has an owner and a set of creators. The user who mints (creates) the NFT indicates which users are co-creators of the NFT. All of the creators will receive a fee each time the NFT is sold. The first owner of the NFT is the user who minted it.

## How it Works

Inside the app you can see all the minted NFTs (with their image, description, upload time and co-creators) arranged chronologically (from newest to oldest) ( Mighty-Market.png ). It should support the pagination of content.

To mint a new NFT you need to drag the image (of the NFT) onto the drag area which will show the mint popup ( Mint-Popup.png ). When the popup shows up you have to add a description and, optionally, select the co-creators (You can send a comma-separated list of user ids). Once you click the *Mint* button, the new entry will appear on the main page.

From the main page you can buy an NFT. To do that, you should press the *Buy now* button. The buyer, seller and co-creators' balances should be updated with the new amounts. Also, the ownership of the NFT should be transferred to the new owner.

Every user will have an initial balance of $100. (It is not necessary to create an endpoint to deposit more money)

Each time an NFT is sold, 20% of the total amount goes to the creators (divided equally) and 80% goes to the owner. There is no limitation on the number of creators.

Example:

| | Balances | | | | | | |
|---|---|---|---|---|---|---|---|
| Action | creator | co-creator 1 | co-creator 2 | co-creator 3 | buyer 1 | buyer 2 | Ownership NFT |
| Initial balance | 100 | 100 | 100 | 100 | 100 | 100 | - |
| Creator mints NFT | 100 | 100 | 100 | 100 | 100 | 100 | creator |
| Buyer 1 buys NFT for $ 10 | 108.50 | 100.50 | 100.50 | 100.50 | 90 | 100 | buyer 1 |
| Buyer 2 buys NFT for $ 100 | 113.50 | 105.50 | 105.50 | 105.50 | 170 | 0 | buyer 2 |

## Tasks

Design and implement the backend API for supporting all the functionality mentioned.

Design and implement the DB schema.

Document the API and data model in Markdown format delivered in the Readme of the repo.

All the business logic functions should be tested with unit tests and the buy endpoint should also be tested with integration tests.

Dockerized for easy dev environment setup (using Docker-Compose) and deployment.

## Tools/Tech to use

**The DB selected should be an SQL one.**
**Any other tech you think it's needed**: You should explain why you have chosen such tech instead of another.

## Deliveries

You should send the whole project sources. Documentation (feel free to include diagrams, Swagger files, or anything you think is needed).
A *readme* file explaining anything you think is important about your solution.

All this should be inside a zip file named web_backend_test_exercise.zip

# What we will evaluate?

Code tidiness/source code organization.
Functional aspects of the exercise.
Design and architecture of the solution.
Use of the requested tech/tools.

# Contact info

Doubts? Contact us via email: Gianluca gianluca@mightyblock.co