D1 - Salezi

D-SAL-100

Salezi

Design APIs fast, manage content easily.

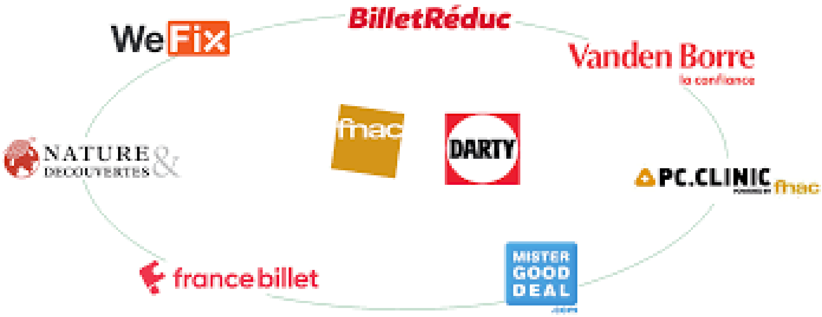
0.2

Salezi

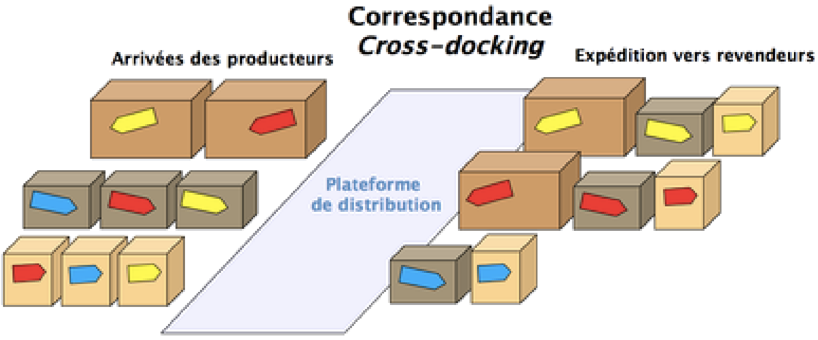
delivery method: GitHub
language: Javascript, SQL

Subcontracting is a recurrent notion in the professional sector. It is a contract between a client company and another provider company asking for the realization of all or part of what the client company had to realize and provide to its own customers.

The company you work for is called Cloudly. And you have just won a big contract with the FNAC which asks you to realize an MVP (Minimum Viable Product) for a new API that it wishes to put at the disposal of their sales partners.



This interface must allow the management of stocks and sales of products for each of the partner sellers. This means that a seller A will not be able to modify the items sold by a seller B but will be able to see them. Moreover, the sellers will be able to received or sell at least books, CDs and electronics. Be careful, some types of products can be added later, this must be taken into account when creating the database.



As a member of Cloudly, your job is to prototype a secure, stable and efficient **backend**, compliant with the specifications you are given. You need to have at least three different roles: **Super Admin**, **Fnac Manager** and **Seller**.

Super Admin can do everything on the application.
Fnac Manager can see and edit all data and he can validate transaction of seller.
Seller can see and edit only her own product and ask for new transaction.
Be careful, **Super Admin** and **Fnac Manager** can access to Admin Panel, but **Seller** can not access to it.
You don't need to develop front office interface, we focus her to back office interface of Strapi.
But you need to anticipate permissions of seller and create it in Strapi Admin Panel.

> Feel free to organize and create the different collections you need in your application.

You are expected to deliver:

- a **backend application** in Strapi with a connection to a SQLite database (you have to push it in Github repository);
- a clear and concise **README** that explains how to launch your API and its features;
- a **Postman** collection exported in JSON format, that tests and covers all relevant's routes of your API for **Seller**

To go further

If you have some time left, FNAC would appreciate to have on the Dashboard of the solution some elements related to the stocks and sales informed on the API. You are free to display the relevant data.

Steps

- Create collections / single type / components
- Create relative postman collection
- Find a way to keep the inventory of a product positive
- Create roles and permissions
- Custom Strapi Dashboard