

Catastima

Insert product

Available or not

Go take and pay

Ansary MARECAR, Blaise VILLIEN, Axel ABOYEJI

Catastima

Le nom ...

- « Catastima » s'écrivant en grec « κατάστημα », signifie « boutique » ou « magasin »
- « Tast » en anglais signifie le « goût »
- « Cata », pour faire référence à la « catastrophe » de cette période
- Adjectif « ta » et « ma », pour faire référence à l'expression « toi et moi »

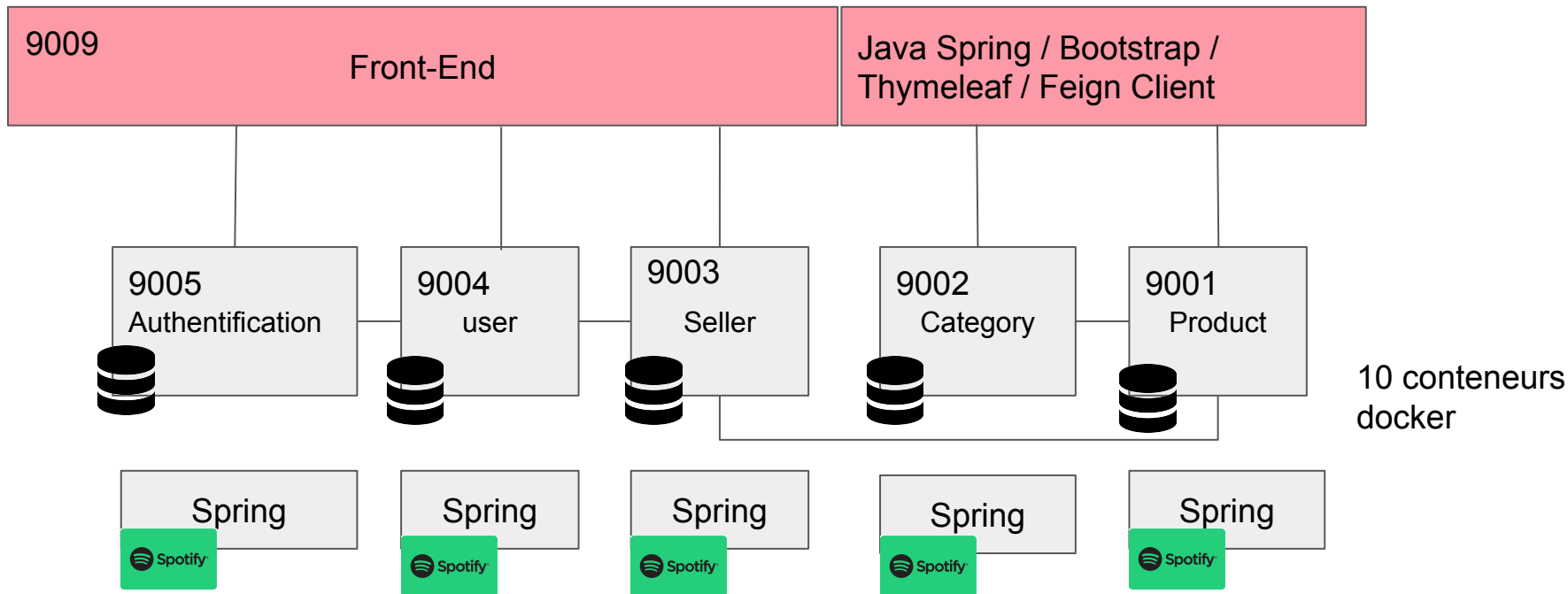
Catastima

Système permettant de visualiser les produits de différentes boutiques (site vitrine).

Seuls les personnes en charge des boutiques nécessite une connexion à l'application.

Les micro services

Définition : structure une application comme un ensemble de services, permet de réduire les charges



Les données transitent via API REST

Base données



product (id, name, picture, category_id, seller_id, price, quantity, stock)

category(id, name, parent)

seller(id, name, user_id)

user(id, name, surname, mail, phone, adresse, city)

authentication (id, user_id, password_id, token)

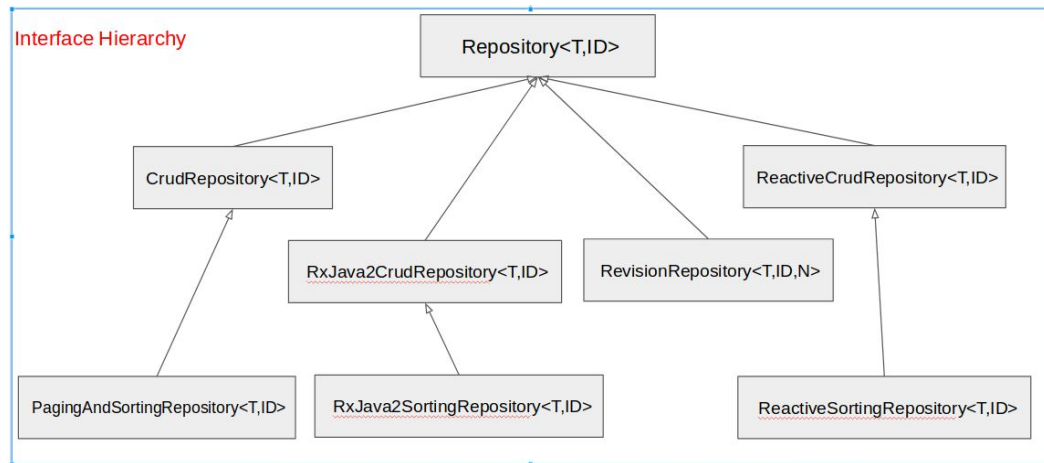
session ⇒ Créé automatiquement par Spring security

1 base de données par micro-service

jpa repository

Notre créateur de requêtes

Créateur de base de données



Nous permettons de réaliser à la fois une création de table en l'absence de fichier SQL dans les dockerfile, mais également via les fichiers SQL permettant de récupérer un important nombre de données.

Connexion

Inscription

Ma boutique

Liste Catégories
(type_product/type_product_parent)

Carrefour



Bordeau, 20 Cl

14€

2

mettre à jour

Carrefour



Bordeau, 20 Cl

14€

2

mettre à jour

Ajouter un produit

Carrefour



Bordeau, 20 Cl

14€

2

mettre à jour

Carrefour



Bordeau, 20 Cl

14€

2

Mettre à jour

Boutique Carrefour

Vin

Carrefour



Bordeau, 1L

20€

2

Carrefour



Sauvignon, 70 Cl

17€

2

Carrefour



Chardoney, 1 L

22€

2

Carrefour



Saint-Emilion, 1 L

54€

2

Carrefour



Bourgogne, 70 Cl

18€

2

Carrefour



Mouton Cadet Bordeaux, 1L

14€

2

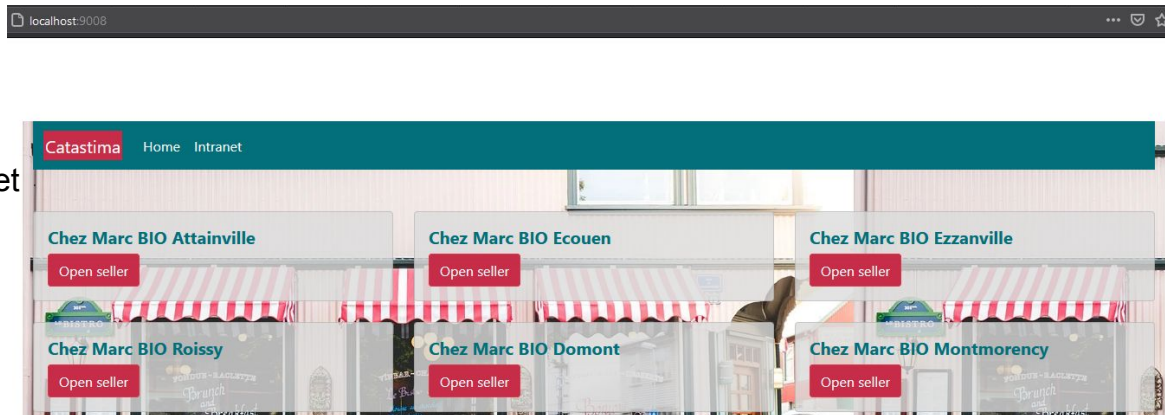
Interface graphique obtenues

```
localhost:9003/map/sellers

JSON  Données brutes  En-têtes
Enregistrer Copier Tout réduire Tout développer Filtre le JSON

0:
  seller_id: 1
  seller_name: "Chez Marc BIO Attainville"
  user_id: 1
1:
  seller_id: 2
  seller_name: "Chez Marc BIO Ecouen"
  user_id: 1
2:
  seller_id: 3
  seller_name: "Chez Marc BIO Ezzanville"
  user_id: 1
3:
  seller_id: 4
  seller_name: "Chez Marc BIO Roissy"
  user_id: 1
4:
  seller_id: 5
  seller_name: "Chez Marc BIO Domont"
  user_id: 1
5:
  seller_id: 6
  seller_name: "Chez Marc BIO Montmorency"
  user_id: 1
```

Récupération des sellers par get



Boisson

Fromager

Boucher

Fruit & Lgumes

Possibilité de
filtrer les
produits par
catégories



Jus de raisin blanc BIO

3.59 € Quantity: 1.5 L 16 available



Jus de pomme BIO

1.27 € Quantity: 6 * 1 L 14 available



Jus avoine BIO

3.5 € Quantity: 1.5 L 12 available



Boisson Épeautre original

3.2 € Quantity: 1 L 23 available



Jus de soja BIO

1.12 € Quantity: 1 L 34 available



Cocktail Orange Mangue

1.99 € Quantity: 25 cl 26 available



Nectar Framboise Fraise

2.12 € Quantity: 21 cl 23 available



Jus de raisin blanc BIO

3.52 € Quantity: 1.5 L 12 available



Nectar Framboise Fraise

2.28 € Quantity: 21 cl 65 available



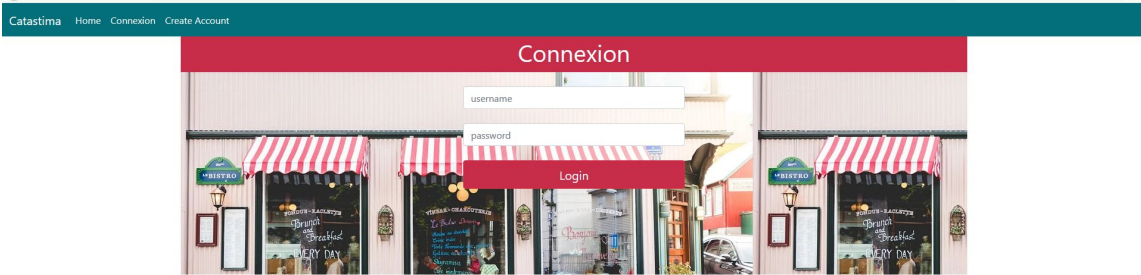
Jus de pomme BIO

1.2 € Quantity: 6 * 1 L 54 available



En ayant cliqué sur le bouton intranet, vous arrivez sur la page de connexion. Il est possible de s'inscrire également. Un compte vous as déjà été créé avec les identifiants suivants :

username : marecar
password : password



A présent, vous êtes connectés, chaque interface correspond à crud permettant de modifier les produits, les catégories et les sellers (vendeurs). Vous avez également une partie modification de vos données.

Exemple : CRUD Catégorie

Category				
Category ID	Name	Category Parent	Create New Category	
1	Fruits & Légumes	1	Edit	Delete
4	Fruits	4	Edit	Delete
5	Fromagers	5	Edit	Delete
7	Poissonnier	7	Edit	Delete
8	Boisson	8	Edit	Delete
9	Boucherie	9	Edit	Delete

Conclusion

difficultés :

- Interconnecter les conteneurs
- Appréhender les technologies (spring, etc.)
- Séparer le back du front

Ce qu'on aurait aimé utiliser :

- Swager
- Front langage différent
- Base de données NOSQL

Ce qu'on a appris:

- Microservices
- Docker
- Java Spring (dont spring security)
- Bootstrap 5
- JPA Repository
- Postman
- Session
- Communication API rest

La suite correspond à l'annexe

Micro-Service : Authentication



Database tables & DAO classes :

- authentication(authentication_id, authentication_login, authentication_password, authentication_timestamp, authentication_token, user_id)

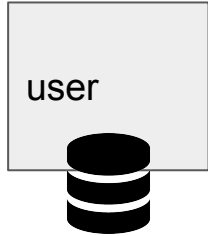
Fonctionnalités:

- Add authentication (at the creation of an account)
- Update password
- **Create a new session**

- **Verify session validity (=read)**

JWT (JSON Web Token) ⇒ Session ⇒ côté utilisateur ⇒ voir si le token est valide

Micro-Service : User



Database tables & DAO classes :

- user(user_id, user_name, user_surname, user_mail, user_phone)
- address(address_id, address_street, address_city, address_cp, address_country)
- user_address(user_id, address_id)

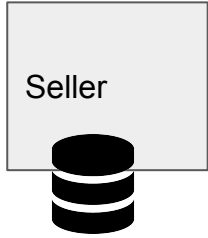
Fonctionnalités:

- Read user
- read address

- Update user
- Create user
- Delete user

- Add address
- Update address
- Delete address

Micro-Service : seller



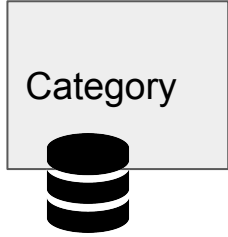
Database tables & DAO classes :

- seller(seller_id, seller_name)
- Seller_user(seller_id, user_id) | Blaise \Rightarrow Auchan, Ansary \Rightarrow Auchan, Axel \Rightarrow Carrefour

Fonctionnalités:

- read seller
- create seller
- add user to a seller
- delete user to a seller

Micro-Service : Category



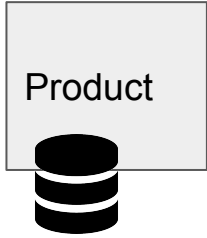
Database tables & DAO classes :

- `category(category_id, category_name, category_parent)`

Fonctionnalités:

- read categories (list of categories)
- create category

Micro-service : Product



Database tables & DAO classes :

product(product_id, product_name, product_price, product_quantity, category_id, seller_id, product_image, product_stock)

Fonctionnalités :

- list all product
- list product from a seller
- **list product from category (don't forget to read product from child categories)**
⇒ **fournir liste des catégories**
- create product
- update product
- delete product
- Read product