

Block 3 Informatique de gestion

Systèmes distribués  
Dossier final



The Big Bazar

Document écrit par :

DELAVAL Kevin

COLLETTE Loïc

Année académique 2020 – 2021

Table des matières

[Correspondance des services 2](#_Toc61462874)

[Diagramme global 2](#_Toc61462875)

[Relevé des problèmes rencontrés 2](#_Toc61462876)

[Exemple de test avec POSTMAN 3](#_Toc61462877)

[Service Order 3](#_Toc61462878)

[Service TVA 4](#_Toc61462879)

[Service Checkout 4](#_Toc61462880)

[Service Stock 4](#_Toc61462881)

[Service Cart 5](#_Toc61462882)

[Scénario de démonstration 5](#_Toc61462883)

[La solution apportée à un problème exceptionnelle 6](#_Toc61462884)

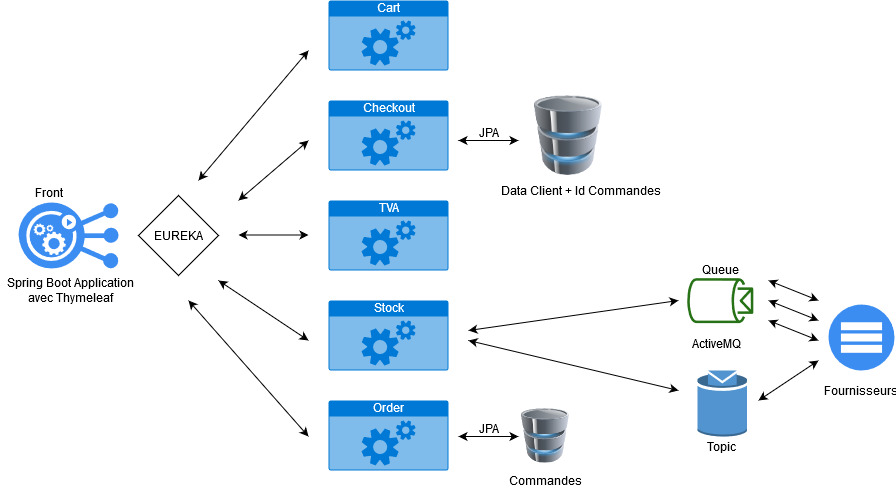
[Gestion de stock 6](#_Toc61462885)

[Deux instances du service TVA 6](#_Toc61462886)

# Correspondance des services

|  |  |
| --- | --- |
| Services | Technologies |
| Eureka | Spring-Boot, Eureka-Server, Maven |
| Cart | Spring-Boot, Rest-Controlleur, Maven |
| TVA | Spring-Boot , Rest-Controlleur, Maven |
| Stock | Spring-Boot , ActiveMQ, Rest-Controlleur, Maven |
| Order | Spring-Boot , JPA, H2, Hibernate, Rest-Controlleur, Maven |
| Checkout | Spring-Boot , JPA, H2, Hibernate, Rest-Controlleur, Maven |
| Api ( front ) | Spring-Boot , Thymeleaf, Spring-Security, Maven |

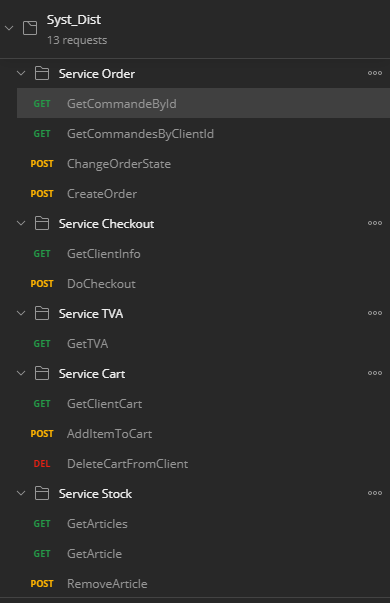
# Diagramme global



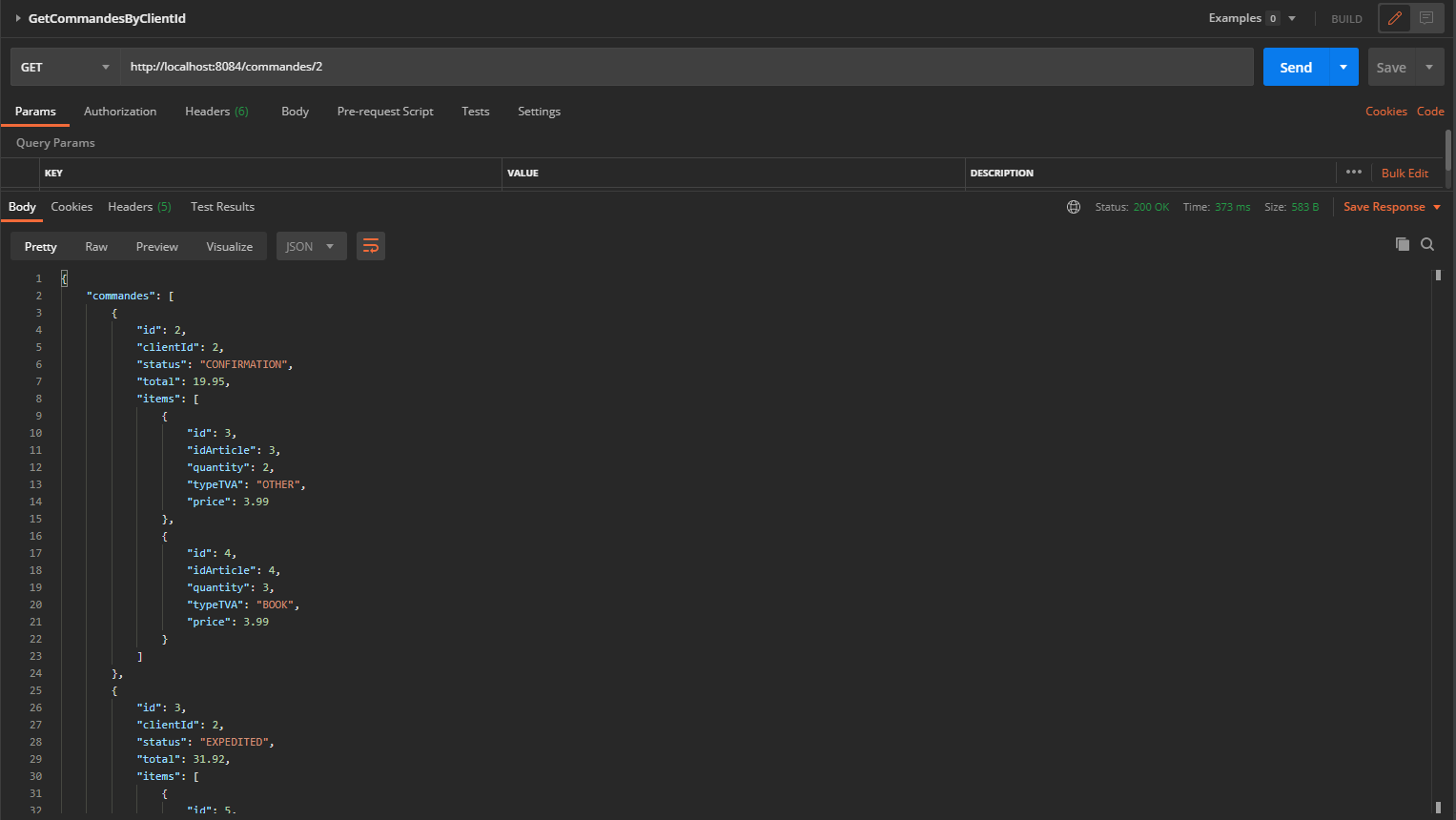
# Relevé des problèmes rencontrés

Durant le développement de la solution un problème a été de comprendre les fonctionnalités fournies par Eureka. Un tutoriel Youtube de ‘‘Java Brain’’ nous a permis d’y voir plus clair. Ensuite le plus gros problème a été de mettre en place JMS avec les Topic et les Queue de messages. Pour cela nous avons consulter beaucoup de ressources tel que spring.io, stackoverflow.com, l’exemple fourni dans le cours sur gitlab.com, etc. Nous avons aussi rencontré un problème du repo GitHub et suit à cela nous avons dû recommencer le projet.

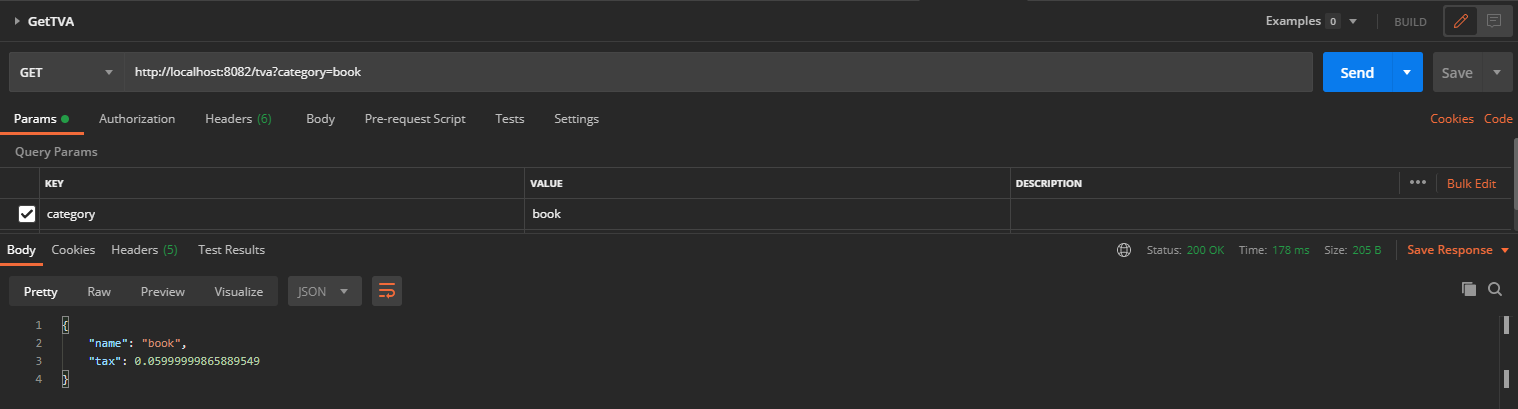
# Exemple de test avec POSTMAN



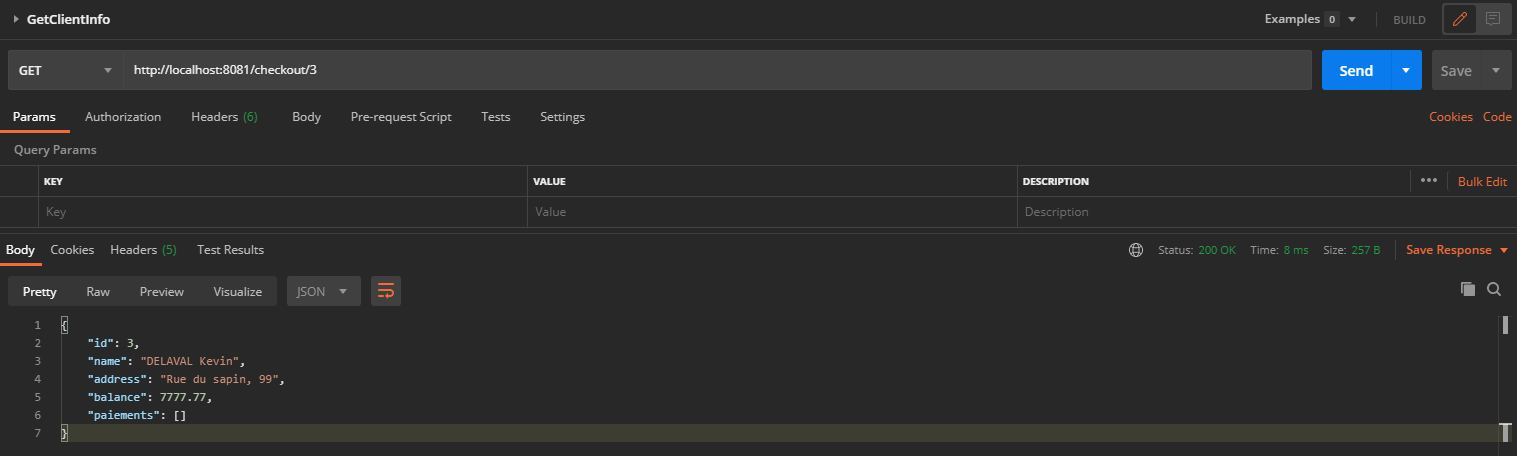
## Service Order



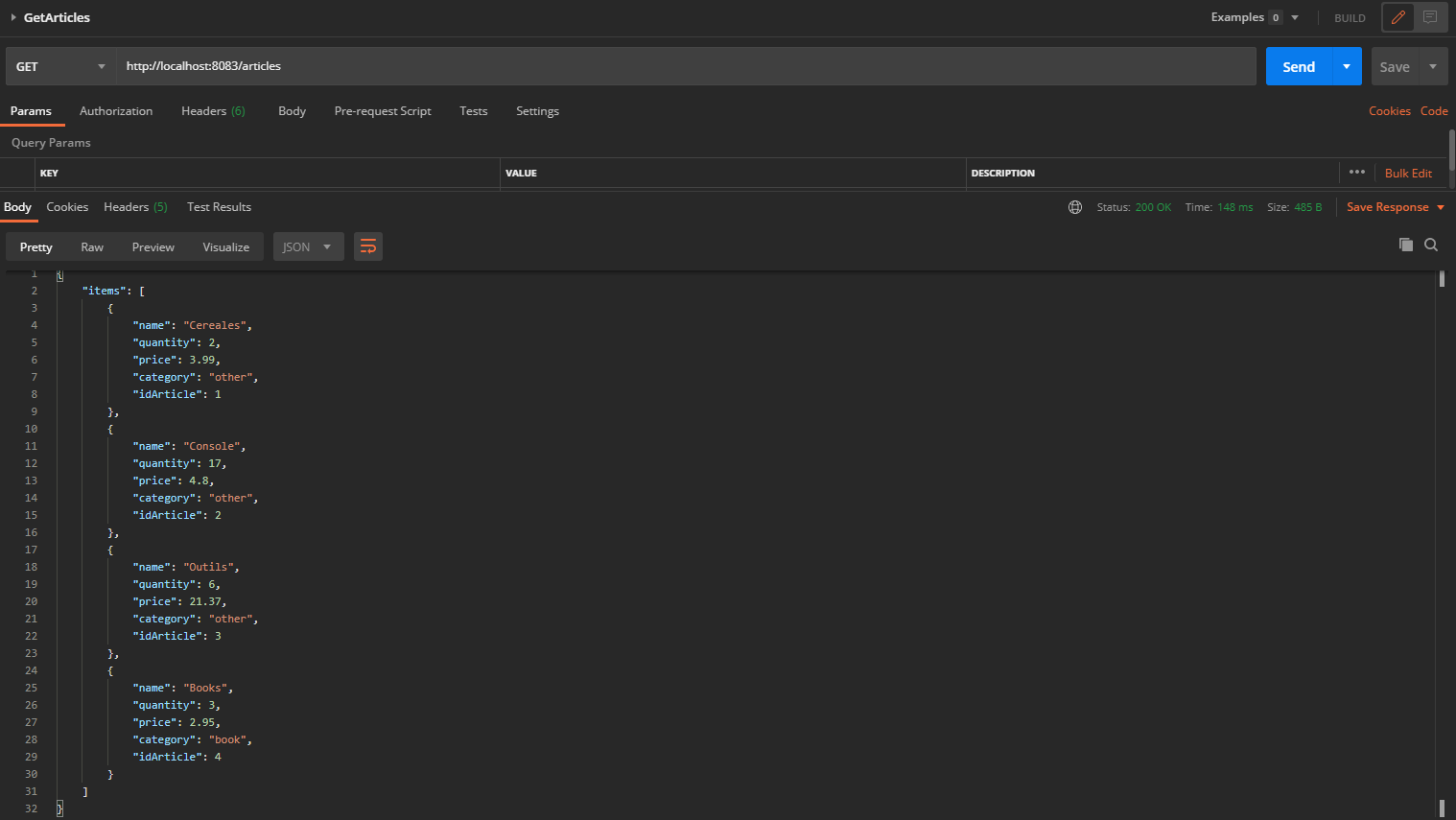
## Service TVA



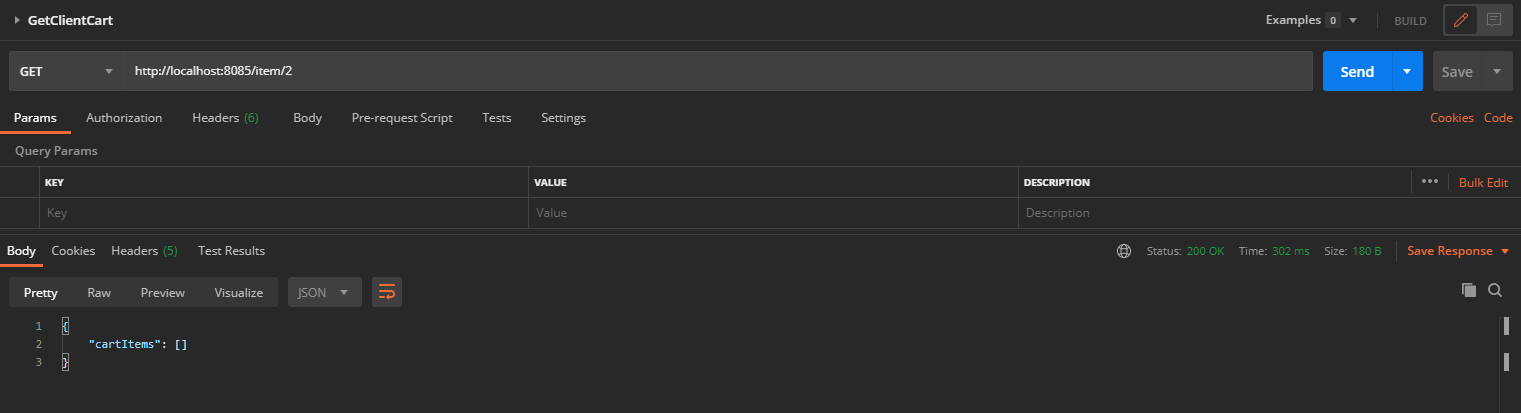
## Service Checkout



## Service Stock

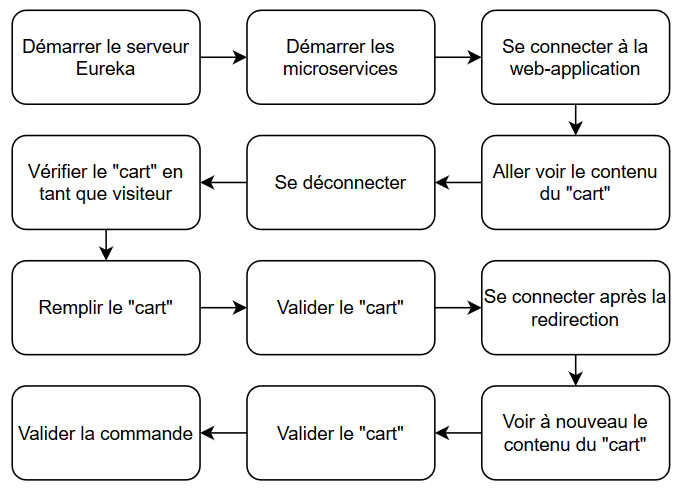


## Service Cart



# Scénario de démonstration

Pour la démonstration il y a une vidéo jointe. Mais nous pouvons quand même expliquer comment va se dérouler la démonstration :



# La solution apportée à un problème exceptionnelle

## Gestion de stock

Plusieurs vérifications ont été mises en place à divers endroits pour vérifier le stock. En effet celui si peut changer à tout moment.

* Si le stock est inférieur à la quantité désirée sur la page du ‘‘magasin’’, le bouton ‘‘add’’ sera grisé et donc désactivé.
* Si sur la page ‘‘cart’’ le stock est inférieur à la quantité du panier, alors l’article concerné sera en rouge et l’on ne pourra pas passer la commande.

## Deux instances du service TVA

L’on peut démarrer deux instances du service TVA (qui est ‘‘stateless’’) et les deux seront disponible. Si jamais l’un des deux viendrait à disparaitre, le serveur EUREKA s’en rendrai compte et permettrai de faire du ‘‘down detecting’’ et de donc tout redirigé sur le service encore fonctionnel et ce sans que le problème ne soit visible par l’utilisateur. De plus cela permet également de faire du ‘‘load balancing’’, si jamais un des 2 services est trop occupé, le serveur EUREKA orientera les requêtes vers l’autre service.