

Pizza Shop

Compte-rendu Partie 1

SIMONIN Enzo

HOLDER Jules

GROMANGIN Clément

S5-ADWM-1

Lien GIT : https://github.com/ClemGrom/pizza_shop_Gromangin_Simonin_Holder_Ouzeau

Table des matières

Installation du projet en local :	3
Partie Authentification :	6
Connexion :	6
Valider token :	7
Refresh token :	9
Inscription :	10
Partie Commande :	11
Accéder à une commande :	11
Valider une commande :	13
Connexion depuis l'API commande vers l'API user :	14
Créer une commande :	15
Partie Catalogue :	17
Accéder à tous les produits	17
Accéder à un produit :	18
Accéder à tous les produits d'une catégorie :	19
Partie RabbitMQ :	20
Lister les commandes :	20
Afficher une commande :	20
Changer l'état d'une commande :	21
Web sockets.....	22
S'abonner à une commande :	22
Ports du projet :	23

Installation du projet en local :

Configuration initiale :

- Clonez le projet :

```
git clone git@github.com:ClemGrom/pizza_shop_Gromangin_Simonin_Holder_Ouzeau.git
```

- Installez les dépendances dans les dossiers auth.pizza-shop, catalogue.pizza-shop, gateway.pizza-shop et shop.pizza-shop avec la commande :

```
composer install
```

- Lancez les containers docker depuis le dossier pizza.shop.components avec la commande :

```
docker-compose up -d
```

Configuration de rabbitmq :

- Lien : <http://localhost:15672>
User : admin, mot de passe : admin
- Créer un échange nommé "pizzashop" de type "DIRECT" :
Bouton exchanges, Name: pizzashop, Bouton add exchange
- Créer une queue nommée "nouvelles_commandes" de type "classic" avec la propriété Durable :
Bouton queues and streams, Type : classic, Name : nouvelles_commandes , Bouton add queue
- Créer un binding entre l'échange "pizzashop" et la queue "nouvelles_commandes" avec la routing key "nouvelle" :
Bouton exchanges, pizzashop, To queue : nouvelles_commandes, Routing key : nouvelle, Bouton bind
- Créer un utilisateur avec un mot de passe puis lui donner les permissions * sur toutes les actions :
Bouton admin, Username : user, Password : user, Bouton add user, Cliquez sur user, Bouton set permission
- Créer une queue dédiée dans le serveur RabbitMQ, nommée "suivi_commandes" avec un binding sur l'échange "pizzashop" avec la routing key "suivi" :
Bouton queues and streams, Type : classic, Name : suivi_commandes , Bouton add queue
Bouton exchanges, Cliquez sur Pizzashop, To queue : suivi_commandes, Routing key : suivi, Bouton bind

Il est possible qu'un redémarrage docker soit nécessaire afin que certain service comme express js et le web sockets fonctionne :
docker compose up -d

Insérez les données dans les bases de données auth, catalogue et commande :

- Auth :

Lien : http://localhost:41222/?server=pizza-shop.auth.db&username=pizza_auth&db=pizza_auth

Mot de passe : pizza_auth

Les données sont dans le dossier auth.pizza-shop/sql, ajouté le fichier pizza_shop.auth.schema.sql et ensuite le fichier pizza_shop.auth.data.sql

- Catalogue :

Lien : http://localhost:41222/?pgsql=pizza-shop.catalogue.db&username=pizza_catalog&db=pizza_catalog&ns=public

Mot de passe : pizza_catalog

Les données sont dans le dossier catalogue.pizza-shop/sql, ajouté le fichier pizza_shop.catalogue.schema.sql et ensuite le fichier pizza_shop.catalogue.data.sql

- Commande :

Lien : http://localhost:41222/?server=pizza-shop.commande.db&username=pizza_shop&db=pizza_shop&select=commande

Mot de passe : pizza_shop

Les données sont dans le dossier shop.pizza-shop/sql, ajouté le fichier pizza_shop.commande.schema.sql et ensuite le fichier pizza_shop.commande.data.sql

- Express :

Lien : http://localhost:41222/?server=api.service.db&username=pizza_express&db=pizza_express&ns=public

Mot de passe : pizza_express

Les données sont dans le dossier express.pizza-shop/sql, ajouté le fichier pizza_shop.schema.sql et ensuite le fichier pizza_shop.data.sql

Partie Authentication :

Connexion :

Local : <http://localhost:41218/api/users/signin>

Docketu : <http://docketu.iutnc.univ-lorraine.fr:41218/api/users/signin>

Utilisation de la méthode http POST, Auth Basic

E-mail : FrançoiseFischer@sfr.fr

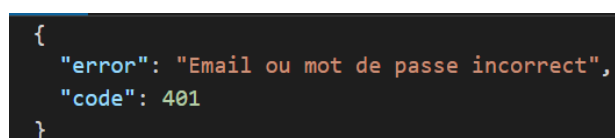
Mot de passe : FrançoiseFischer

Réponse :



```
{
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpc3MiOiJhdXRoLnBpenphLXNob3AiLCJpYXQiOiJlbnQ3NTMzMzksImV4cCI6MTcwNDc2ODMzOSwidXByIjp7ImVtYWlsIjoiriRnJhblx1MDBlN29pc2VGaXNjaGVyQHNMci5mciIsInVzZXJyYW1lIjoiriRnJhblx1MDBlN29pc2VGaXNjaGVyIiwicmVmcmVzaF90b2t1biI6ImI0MTk3ZDY4MmJmMGE4ZTEwODAsMTFhMDYyMGFmMzVkODIyMjYxNTkzM2M4MTJmNWYwZWl1ZmUwOGIyMDQ1NTMifX0.xmy4ECAR00Z2uMh-aEBoxQuhZl4ZVR0J67C4MQptj3-L10yQsROWkSEuHn5NQXtAE1FUeF1mI1kVffriqufaXA",
  "refresh_token": "b4197d682bf0a8e1180911a0620af35d8222615933c812f5f0eb5fe08b204553"
}
```

Message d'erreur en cas de mot de passe ou email incorrect :



```
{
  "error": "Email ou mot de passe incorrect",
  "code": 401
}
```

Valider token :

Local : <http://localhost:41218/api/users/validate>

Docketu : <http://docketu.iutnc.univ-lorraine.fr:41218/api/users/validate>

Utilisation de la méthode http GET, Auth Bearer

Dans notre exemple :

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpc3MiOiJhdXRoLnBpenphLXNob3AiLCJpYXQiOiJlE3MDQ3NTMzMzksImV4cCI6MTcwNDc2ODMzMzOswidXB5Ijp7ImVtYWlsIjoiRnJhbGx1MDBIN29pc2VGaXNjaGVyQHNmci5mcilsluVzZXJ1YWw1IjoiRnJhbGx1MDBIN29pc2VGaXNjaGVyIiwicmVmcmVzaF90b2tlbil6ImI0MTk3ZDY4MmJmMGE4ZTEwODA5MTFhMDYyMGFmMzVkODIyMjYxNTkzM2M4MTJmNWYwZWl1ZmUwOGlyMDQ1NTMifX0.xmy4ECARO0Z2uMh-aEBoxQuhZl4ZVROJ67C4MQptj3-L10yQsROWkSEuHn5NQXtAE1FUeF1ml1kvffriqufaXA

Réponse :

```
{
  "user": {
    "email": "FrançoiseFischer@sfr.fr",
    "username": "FrançoiseFischer",
    "refresh_token":
      "b4197d682bf0a8e1180911a0620af35d8222615933c812f5f0eb5fe08b204553"
  }
}
```

Message d'erreur lorsque le token de vérification ne correspond pas :

```
{
  "message": "401 Unauthorized",
  "exception": [
    {
      "type": "Slim\\Exception\\HttpUnauthorizedException",
      "message": "Token incorrect",
      "code": 401,
      "file": "/var/www/src/domain/services/ServiceAuth.php",
      "line": 61
    }
  ]
}
```

Message d'erreur lorsque le token est arrivé à expiration :

```
{
  "message": "401 Unauthorized",
  "exception": [
    {
      "type": "Slim\\Exception\\HttpUnauthorizedException",
      "message": "Token expired",
      "code": 401,
      "file": "/var/www/src/domain/services/ServiceAuth.php",
      "line": 61
    }
  ]
}
```


Refresh token :

Local : <http://localhost:41218/api/users/refresh>

Docketu : <http://docketu.iutnc.univ-lorraine.fr:41218/api/users/refresh>

Utilisation de la méthode http POST, Auth Bearer

Authentication bearer dans notre exemple (refresh_token):

- 15b30d86bf8b31ea1357367716cd08c54468e99a84f901b281a0b655854d2d2d

Réponse :

```
"access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9
    .eyJpc3MiOiJhdXRoLnBpenphLXNob3AiLCJpYXQiOiJlM0E3MDAwNjI4NDAsImV4cCI6MTcwM
    DA3Nzg0MCwidXByIjp7ImVtYWlsIjoiQWxpeFBlcnJvdEBmcmVlLmZyIiwidXNlc5hbwU
    ioiJBbG14UGVycm90IiwicmVmcmVzaF90b2t1biI6IjI1NDk0YTdiMjhY2I0MzYxZWU0M
    WRmNmYwYTU2NGMxNWQyM2VmYWI2YzYyM2U2NTZkNTI5NDU3MzNiZnQ4NTYifX0
    .cwCH1vBbK-42LWQt9eA7JXYZZp01PCL0eIJayH8hrZXRgd_ClnZjapTydZv
    -HQaIsug8qB8hK4N-Dr1Bhuna2A",
"refresh_token": "25494a7b28acb4361ee41df6f0a564c15d23efab6c623e656d5297e7
    33b74856"
```

Message si le token est incorrect :

```
"message": "401 Unauthorized",
"exception": [
    {
        "type": "Slim\\Exception\\HttpUnauthorizedException",
        "message": "Token incorrect",
        "code": 401,
        "file": "/var/www/src/domain/auth/AuthProvider.php",
        "line": 51
    }
]
```

Inscription :

Local : <http://localhost:41218/api/users/signup>

Docketu : <http://docketu.iutnc.univ-lorraine.fr:41218/api/users/signup>

Utilisation de la méthode http POST, Body Json

Body de la requête :

```
{  
  "email": "test@gmail.com",  
  "username": "nzo",  
  "password": "test"  
}
```

Réponse :

En cas de réussite une erreur 201 est renvoyé

En cas de compte déjà existant :

```
"error": "Compte déjà existant avec cet email",  
"code": 409
```

Partie Commande :

Accéder à une commande :

Local : <http://localhost:41218/api/commandes/ce852d5f-4845-354e-8ef8-c86a1a0d8736>

Docketu : <http://docketu.iutnc.univ-lorraine.fr:41218/api/commandes/ce852d5f-4845-354e-8ef8-c86a1a0d8736>

Utilisation de la méthode http GET

Réponse :

```
{
  "type": "ressource",
  "commande": {
    "id": "ce852d5f-4845-354e-8ef8-c86a1a0d8736",
    "date_commande": "2023-08-28 14:35:31",
    "type_livraison": 2,
    "etat": 3,
    "mail_client": "LaurenceLambert@live.com",
    "montant": 106.89,
    "delai": 0,
    "items": [
      {
        "numero": "10",
        "libelle": "panna cotta",
        "taille": 2,
        "quantite": 4,
        "tarif": 5.99,
        "libelle_taille": "grande"
      },
      {
        "numero": "1",
        "libelle": "Margherita",
        "taille": 1,
        "quantite": 3,
        "tarif": 8.99,
        "libelle_taille": "normale"
      },
      {
        "numero": "3",
        "libelle": "Savoyarde",
        "taille": 2,
        "quantite": 4,
        "tarif": 13.99,
        "libelle_taille": "grande"
      }
    ]
  },
  "links": {
    "self": {
      "href": "/api/commandes/ce852d5f-4845-354e-8ef8-c86a1a0d8736/"
    },
    "valider": {
      "href": "/api/commandes/ce852d5f-4845-354e-8ef8-c86a1a0d8736/"
    }
  }
}
```

Message d'erreur lorsque l'identifiant de commande est introuvable :

```
{  
  "error": "Commande introuvable",  
  "code": 404  
}
```

Valider une commande :

Local : <http://localhost:41218/api/commandes/eedeea63-4032-3b84-af42-3c257d5c3c33>

Docketu : <http://docketu.iutnc.univ-lorraine.fr:41218/api/commandes/eedeea63-4032-3b84-af42-3c257d5c3c33>

Utilisation de la méthode http PATCH

Message d'erreur lorsque la commande en question a déjà été validée :

```
{  
  "error": "Commande déjà validée",  
  "code": 400  
}
```

Message d'erreur lorsque la commande est introuvable :

```
{  
  "error": "Commande introuvable",  
  "code": 404  
}
```

Connexion depuis l'API commande vers l'API user :

Local : <http://localhost:41218/api/connection>

Docketu : <http://docketu.iutnc.univ-lorraine.fr:41218/api/connection>

Utilisation de la méthode http POST, Auth Basic

E-mail : FrançoiseFischer@sfr.fr

Mot de passe : FrançoiseFischer

Réponse :

```
{
  "access_token": "eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9
    .eyJpc3MiOiJhdXRoLnBpenphLXNob3AiLCJpYXQiOiE3MDAwNzU3NDIsImV4cCI6MTcwM
    DA5MDc0MiwiZXByIjp7ImVtYWlsIjoiQWxwaG9uc2VGbGV1cn1Ac2ZyLmZyIiwidXNlcm5
    hbWUiOiJBbHBob25zZUZsZXVyeSIsInJlZnJlc2hfdG9rZW4iOiIwY2E5MGFhY2M0ODc5O
    WFjZmRjZTcxM2Q0Nzk3YThkZTZlYWVmYmE2ZWE2ZTAxZThiZmVhYjYyYjY1NTQ3In1
    9.5QBnMOZCLBP27fT_1kmQ06Xr4qkAKGvjBtdI_bfbCtVkYY5BoEX2RBUMS40jiArbVCwT
    fgxD7g-G1B4cqE2pAw",
  "refresh_token": "0ca90aacc48799acfdce713d4797a8de6eaeafb6ea6e01e8bfeab022
    89b65547"
}
```

Message d'erreur en cas de mot de passe ou email incorrect :

```
{
  "error": "Email ou mot de passe incorrect",
  "code": 401
}
```

Créer une commande :

Local : <http://localhost:41218/api/commandes>

Docketu : <http://docketu.iutnc.univ-lorraine.fr:41218/api/commandes>

Utilisation de la méthode http POST

Token de vérification de notre exemple : **(Vous devez ajouter un token dans Auth Bearer)**

eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzUxMiJ9.eyJpc3MiOiJhdXRoLnBpenphLXNob3AiLCJpYXQiOiJlbnQ1NjAsImV4cCI6MTcwNDc2OTU2MCwidXByljp7ImVtYWlsIjojQWxwaG9uc2VGbGV1cnlAc2ZyLmZyIiwidXNlcm5hbWUiOiJBbHBob25zZUZsZXVyeSIsInJlZnJlc2hfdG9rZW4iOiI0OGFmODc3OWU2NGRiMWMxOWNhYzAwMTI3NjIxZDgyYzUwMGQ2OTk3OGY5NDVjZTM5NzE0NjExMDBIMGE1OGEzIn19.vVG_6bmfbVa6nGYiCRjzuWYwmleR8DSGsKDDBXrOvhLtdNBmM7rrDCoUDKvNcxvynfg35bU2pRLSj3HTkSda9w

Body de la commande à ajouter dans notre exemple :

```
{
  "type_livraison": 1,
  "items": [
    {
      "numero": 2,
      "taille": 1,
      "quantite": 1
    },
    {
      "numero": 3,
      "taille": 1,
      "quantite": 3
    }
  ]
}
```

Réponse :

```
{
  "type_livraison": 1,
  "items": [
    {
      "numero": 2,
      "taille": 1,
      "quantite": 1
    },
    {
      "numero": 3,
      "taille": 1,
      "quantite": 3
    }
  ]
}
```

Réponse en cas de token incorrect :

```
{
  "message": "401 Unauthorized",
  "exception": {
    "type": "Slim\\Exception\\HttpUnauthorizedException",
    "message": "Token incorrect",
    "code": 401,
    "file": "/var/www/src/domain/services/ServiceAuth.php",
    "line": 62
  }
}
```


Partie Catalogue :

Accéder à tous les produits :

Local : <http://localhost:41218/api/produits>

Docketu : <http://docketu.iutnc.univ-lorraine.fr:41218/api/produits>

Utilisation de la méthode http GET

Réponse :

```
[
  {
    "numero_produit": 1,
    "libelle_produit": "Margherita",
    "libelle_categorie": "Pizzas",
    "URI": "/api/produits/1"
  },
  {
    "numero_produit": 2,
    "libelle_produit": "Reine",
    "libelle_categorie": "Pizzas",
    "URI": "/api/produits/2"
  },
  {
    "numero_produit": 3,
    "libelle_produit": "Savoyarde",
    "libelle_categorie": "Pizzas",
    "URI": "/api/produits/3"
  },
  {
    "numero_produit": 4,
    "libelle_produit": "Pepperoni",
    "libelle_categorie": "Pizzas",
    "URI": "/api/produits/4"
  },
  {
    "numero_produit": 5,
    "libelle_produit": "cola",
    "libelle_categorie": "Boissons",
    "URI": "/api/produits/5"
  },
]
```

Accéder à un produit :

Local : <http://localhost:41218/api/produits/10>

Docketu : <http://docketu.iutnc.univ-lorraine.fr:41218/api/produits/10>

Utilisation de la méthode http GET

Réponse :

```
{
  "numero_produit": 10,
  "libelle_produit": "panna cotta",
  "libelle_categorie": "Desserts",
  "URI": "/api/produits/10",
  "tailles": [
    {
      "id_taille": 1,
      "libelle_taille": "normale",
      "tarif": 4.99
    },
    {
      "id_taille": 2,
      "libelle_taille": "grande",
      "tarif": 5.99
    }
  ]
}
```

Erreur en cas de produit introuvable :

```
{
  "error": "Produit introuvable",
  "code": 404
}
```

Accéder à tous les produits d'une catégorie :

Local : <http://localhost:41218/api/categories/5/produits>

Docketu : <http://docketu.iutnc.univ-lorraine.fr:41218/api/categories/5/produits>

Utilisation de la méthode http GET

Réponse :

```
[
  {
    "numero_produit": 1,
    "libelle_produit": "Margherita",
    "libelle_categorie": "Pizzas",
    "URI": "/api/produits/1"
  },
  {
    "numero_produit": 2,
    "libelle_produit": "Reine",
    "libelle_categorie": "Pizzas",
    "URI": "/api/produits/2"
  },
  {
    "numero_produit": 3,
    "libelle_produit": "Savoyarde",
    "libelle_categorie": "Pizzas",
    "URI": "/api/produits/3"
  },
  {
    "numero_produit": 4,
    "libelle_produit": "Pepperoni",
    "libelle_categorie": "Pizzas",
    "URI": "/api/produits/4"
  }
]
```

Erreur en cas de catégorie introuvable :

```
{
  "error": "Catégorie introuvable",
  "code": 404
}
```

Partie RabbitMQ :

Lister les commandes :

Lien : <http://localhost:41630/commandes>

Docketu : <http://docketu.iutnc.univ-lorraine.fr:41630/commandes>

```
[
  {
    "delai": 0,
    "id": "56ad35fa-3122-31d6-86c8-294f907be5d9",
    "date_commande": "1977-05-17T21:38:36.000Z",
    "type_livraison": 1,
    "etape": 1,
    "montant_total": 17.39,
    "mail_client": "paul.susanne@gmail.com"
  },
  {
    "delai": 0,
    "id": "56ad35fa-3122-31d6-86c8-294f907be5d9",
    "date_commande": "1977-05-17T21:38:36.000Z",
    "type_livraison": 1,
    "etape": 1,
    "montant_total": 17.39,
    "mail_client": "paul.susanne@gmail.com"
  },
  {
    "delai": 0,
    "id": "819466e7-bfee-3cd6-bb8a-f675601b867a",
    "date_commande": "1981-03-07T12:43:58.000Z",
    "type_livraison": 1,
    "etape": 1,
    "montant_total": 75.89,
  }
]
```

Afficher une commande :

Lien : <http://localhost:41630/commandes/cc1e6220-774a-37bd-b8cf-e7b9dc5c446a>

Docketu : <http://docketu.iutnc.univ-lorraine.fr:41630/commandes/cc1e6220-774a-37bd-b8cf-e7b9dc5c446a>

```
{
  "delai": 0,
  "id": "cc1e6220-774a-37bd-b8cf-e7b9dc5c446a",
  "date_commande": "2022-10-03T08:39:21.000Z",
  "type_livraison": 3,
  "etape": 3,
  "montant_total": 62.94,
  "mail_client": "GrégoireDupont@tele2.fr"
}
```

Changer l'état d'une commande :

Type **patch** :

Lien : <http://localhost:41630/commandes/cc1e6220-774a-37bd-b8cf-e7b9dc5c446a>

Docketu : <http://docketu.iutnc.univ-lorraine.fr:41630/commandes/cc1e6220-774a-37bd-b8cf-e7b9dc5c446a>

```
{
  "delai": 0,
  "id": "51b570bb-f9cf-3923-ba2b-901192226c02",
  "date_commande": "2023-07-03T09:33:11.000Z",
  "type_livraison": 2,
  "etape": 2,
  "montant_total": 50.92,
  "mail_client": "AlphonseFleury@sfr.fr"
}
```

Affichage de la commande avec son état modifié et modifications dans la base de données, erreur si la commande est déjà finie

La validation d'une commande depuis l'API commande l'envoi directement dans la base de données de notre micro-services ExpressJS.

Web sockets

S'abonner à une commande :

Changer la connexion de http à websockets sur postman,

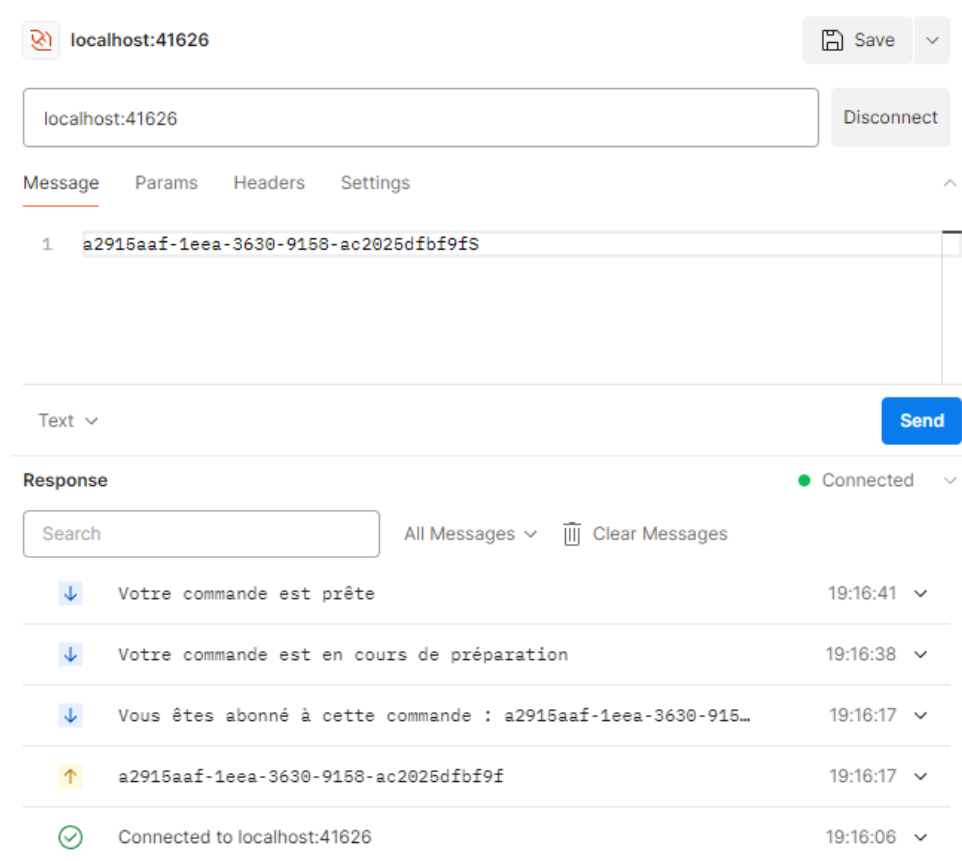
Lien : localhost:41626

Docketu : docketu.iutnc.univ-lorraine.fr:41626

Après la connexion envoyé un message avec votre numéro de commande, par exemple :

a2915aaf-1eea-3630-9158-ac2025dfbf9f

Ensuite à chaque fois que le status de la commande changera avec la partie « Changer l'état d'une commande ci-dessus vous recevrez une notification dans le chat avec l'état de votre commande :



Ports du projet :

API :

- shop.pizza-shop : 41215:80
- catalogue.pizza-shop : 41216:80
- api.pizza-auth : 41217:80
- gateway.pizza-auth : 41218:80

DB :

- pizza-shop.commande.db : 41219:3306
- pizza-shop.catalogue.db : 41220:5432
- pizza-shop.auth.db : 41221:3306

Adminer :

- adminer : 41222:8080

RabbitMQ :

- rabbitmq : 15672:15672
- rabbitmq.php : 15673:80

ExpressJS :

- API express : 41630:3000
- BDD Express : 41625:3306

WebSocket Js :

- WebSocket : 41626:3000