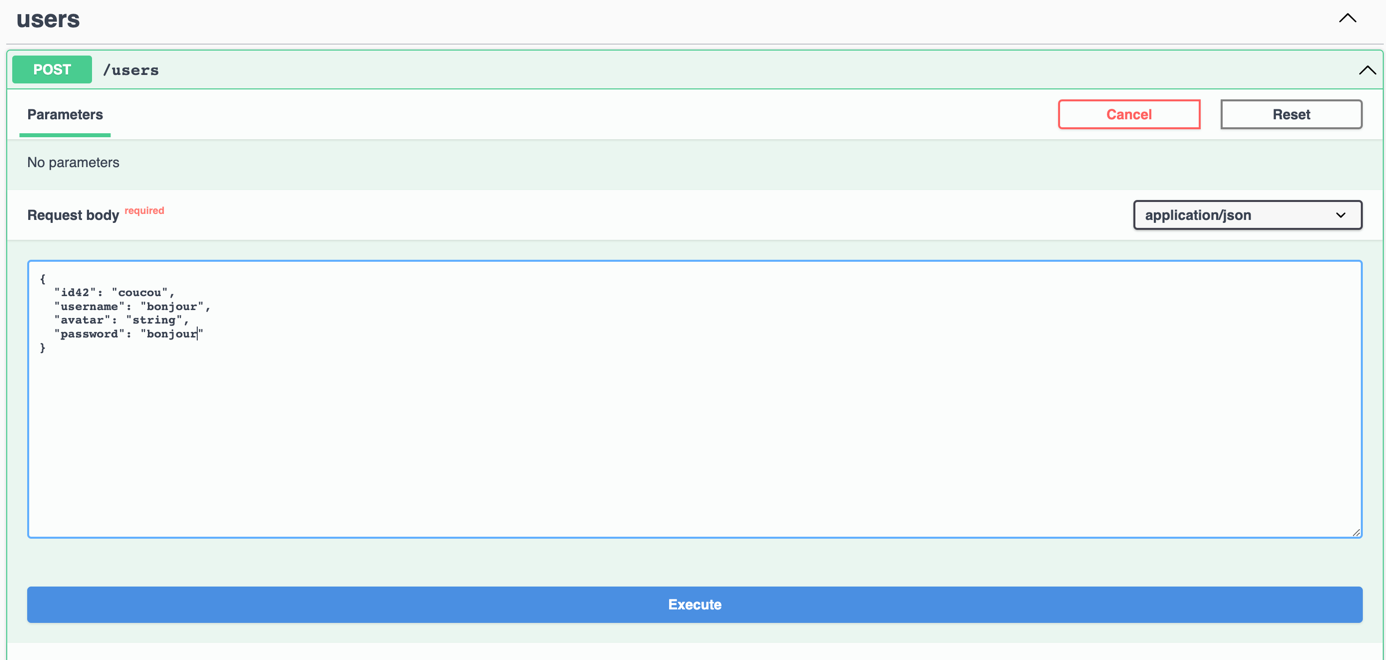
* Comme dhab, pour lancer le pong, aller à la racine du projet et faire `docker-compose up –build`, ouvrir <http://localhost:3000/>, <http://localhost:4000/api> et <http://localhost:8080/>
* Créer des users a partir de Swagger et obtenir leur token dans Postman :
  + Dans swagger :
    - Cliquer sur POST /users (en vert) puis Try it out puis remplir les champs id42, username, avatar, password. Tu mets ce que tu veux, tu peux meme laisser « string ». Pour l’avatar normalement c’est une image mais la on peut laisser une string. Il faut juste bien que tu notes le mot de passe correspondant a l’username car ensuite il sera chiffré dans la database.



* + - Cliquer sur execute. Ensuite tu peux aller sur adminder (localhost8080) et tu vas voir dans la table User, l’user que tu viens de créer s’est affiché. Tu peux en créer autant que tu veux bien sur sur swagger
  + Dans postman :
    - Faire une nouvelle request, sélectionner POST et entrer écrire <http://localhost:4000/auth/login> . Juste en dessous tu selectionne Body, puis encore juste en dessous tu selectionnes Raw et JSON. Tu rentres l’id42 et le password du user. Tu click sur send. En reponse, tu auras le token !

Une image contenant texte

Description générée automatiquement

Si tu as besoin d’avoir le token d’un autre user, tu fais une **nouvelle** requete exactement de la meme manière mais tu mets le bon id42 + passsword, celui de l’user qui t’interesse

* Dans ton code, tu peux donc copier coller le token donc la partie orange apres access\_token. Cest un bearer token (je sais pas trop pourquoi cest ce type de token, cest Iuri qui a gere ca). Tu le mets dans le code entre des guillemets ` ` .
* Dans postman, tu peux faire pleins d’autre requetes. Pour savoir ce qu’il faut ecrire avant le send, genre <http://localhost:4000/auth/login> , tu peux t’appuyer sur postman. Par exemple si je reprends les requetes que jai mis dans le code. La premiere requete cest GET users. Pour savoir quoi mettre, dans swagger j’ai cliquer sur Get /users puis try it out puis execute. Voila ma response :



Donc dans mq requete sur postman je mets exactement ce qu’il y a dans request url donc on voit <http://localhost:4000/users>. ET je selectionne GEt car cest une requete GET :

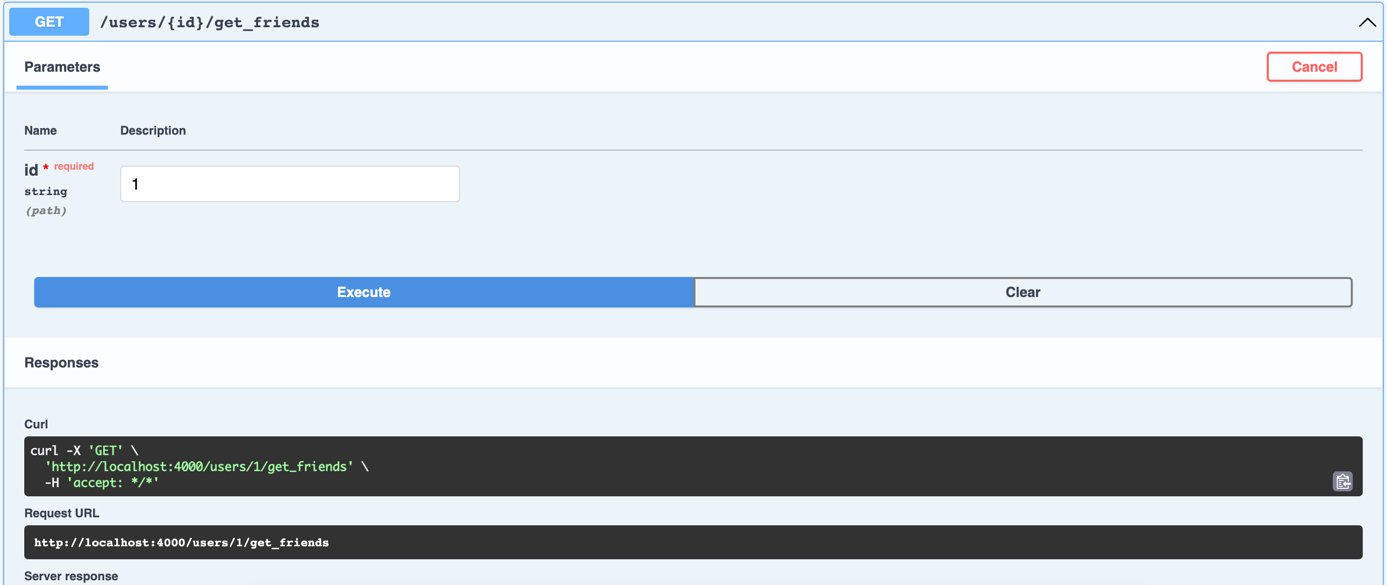
Une image contenant texte

Description générée automatiquement

* Dans mon code, je mets ce qu’il y a apres localhost :4000 donc ici juste /users

Une image contenant texte

Description générée automatiquement

* Dans cette requete /users/ pas besoin de token.
* Faisons un exemple avec token :
* Dans mon exemple de code, je veux avoir la liste des amis de l’user 1.
* Je vais dans swagger pour voir quelle requete pourrait me convenir. Ca semble etre Get friend
* 
* Donc pour l’user 1, je peux rentrer dans postman <http://localhost:4000/users/1/get_friends> (pour un autre user il faudrait changer le 1 en l’id du user qui nous interesse.)
* Ensuite il faut aller dans Autorisation, selectionner bearer token comme type et copier coller le token de l’id qui nous interesse. Ensuite tu peux faire send et tu auras les infosUne image contenant texte

  Description générée automatiquement
* Etc etc pour toutes les requetes GET, POST, PATCH, DELETE. Parfois il faut ajouter des parametre sur postman comme on l’a fait pour obtenir le token
* Erreur courantes de responses
* Error 401 cest que tu n’as pas mis de token alors qu’il en fallait un OU qu’il est expire
* Error 404 cest que la requete mal ecrite (genre faute de frappe, [http://localhost:4000/users/1/get\_friend](http://localhost:4000/users/1/get_friends) al lieu de <http://localhost:4000/users/1/get_friends> par exemple) OU que l’user dont tu essaie d’avoir les infos ou modifier ou supprimer les infos n’existe pas (genre <http://localhost:4000/users/12/get_friends> alors que l’user avec l’id 12 n’existe pas par exemple)