**Mise en place du conteneur de l'api**

Les variables ci-dessous pour stocker les valeurs de URL et du password et pour le passer à l'api

***MYSQL\_URL***

***MYSQL\_PASSWORD***

**Test unitaire de l'api**

Pour tester unitairement l’api vous pouvez :

**1/ Dans le dossier archive exécuter les commandes ci-dessous :**

*docker network create cicd\_network*

*docker container run -d --network cicd\_network datascientest/mysql-k8s:1.0.0*

*docker image build . -t api\_test:latest*

*docker container run --rm -p 8000:8000 --env-file credentials.env --network cicd\_network --name api\_test api\_test*

**ou**

**2/ Utiliser l'image que nous avons uploadé dans notre compte dockerhub :**

Le fichier **credentials.env** contient la valeur de nos deux variables ci-dessus (a adapter avec le vôtre)

*docker image build . -t fidzongoformation/fast-api:1.0.0*

*docker image push fidzongoformation/fast-api:1.0.0*

*docker container run --rm -p 8000:8000 --env-file credentials.env --network cicd\_network fidzongoformation/fast-api:1.0.0*

**Test Global dans kubernetes**

Pour ce faire vous pouvez exécuter les commandes ci-dessous dans l’ordre ci-dessous :

Le mot de passe dans le fichier yml est codé en base64

1/

***kubectl create -f my-config-map-eval.yml***

- Pour vérifier la création exécuter cette commande :

***kubectl get configmap |grep my-config-map-eval***

2/

***kubectl create -f my-secret-eval.yml***

- Pour vérifier la création exécuter cette commande :

***kubectl get secret |grep my-secret-eval***

3/

***kubectl create -f my-deployment-eval.yml***

- Pour vérifier la création exécuter cette commande :

***kubectl get deployment |grep my-deployment-eval***

4/

***kubectl create -f my-service-eval.yml***

- Pour vérifier la création exécuter cette commande :

***kubectl get service |grep my-service-eval***

5/

***kubectl create -f my-ingress-eval.yml***

- Pour vérifier la création exécuter cette commande :

***kubectl get ingress |grep my-ingress-eva***

Vous pouvez également effectuer ces étapes en mode graphique et vérifier visuellement leur bonne exécution.

**Quelques exemples d’appels api**

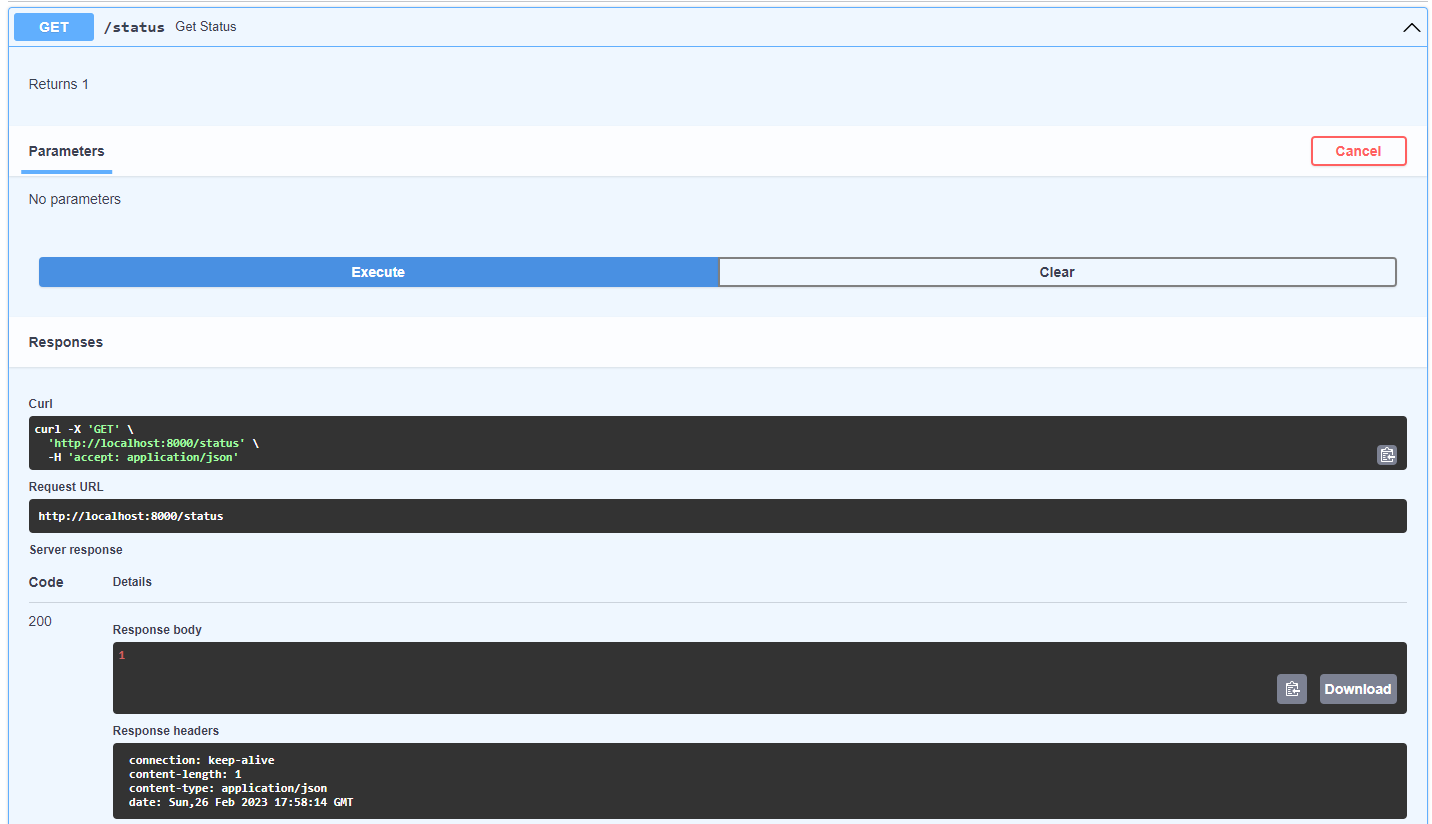


**curl -X 'GET' \**

**'http://localhost:8000/status' \**

**-H 'accept: application/json'**

Où

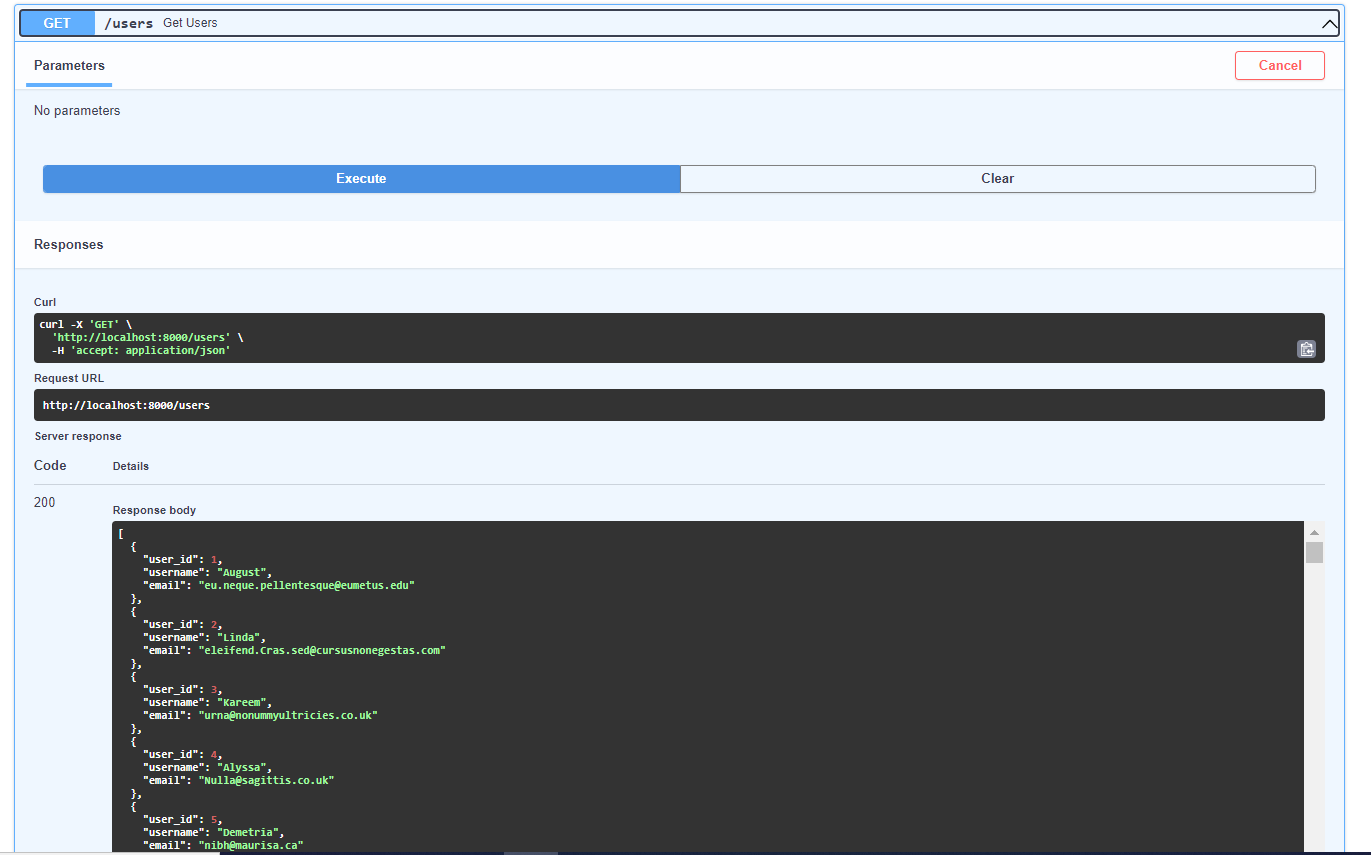


**curl -X 'GET' \**

**'http://localhost:8000/users' \**

**-H 'accept: application/json'**

Ou



**curl -X 'GET' \**

**'http://localhost:8000/users/5' \**

**-H 'accept: application/json'**

Où

