Maïmouna Sarah Diakité  
Gnanda Paule Axelle Kouamé  
Kponaho Anne-Laure Magnané  
Techniques de l’informatique  
groupe 4318

Projet 2: Développement d’une api de conversion

Travail présenté à  
M. Nicolas Payre  
Département des Techniques de l’informatique   
Pour le cours  
*Exploration de nouvelles technologies*

Cégep de Sherbrooke  
18 novembre 2023

Table des matières

[Lien vers le dépôt GitHub 3](#_Toc151073838)

[Description de l'API 3](#_Toc151073839)

[Tâches au choix 5](#_Toc151073840)

[Docker 5](#_Toc151073841)

[MongoDB 5](#_Toc151073842)

[Conclusion 6](#_Toc151073843)

# Lien vers le dépôt GitHub

Le dépôt contient toute les copies de nos fichier des configuration et annexe.

Lien : <https://github.com/Paulaxelle12/projet2_exploration.git>

## Description de l'API

L'API Converter.js offre une solution complète pour la conversion de cinq types d'unités de mesure différentes. Les Endpoint disponibles sont soigneusement définis pour garantir une utilisation précise et efficace. Les conversions prises en charge comprennent :

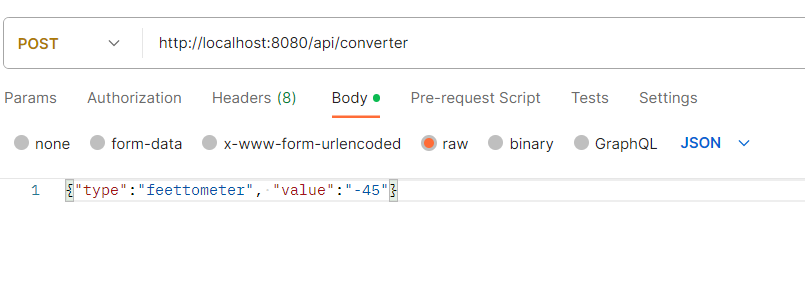
* **feettometer** : Convertit les pieds en mètres.
* **metertofeet** : Convertit les mètres en pieds.
* **kilotopound** : Convertit les kilogrammes en livres.
* **poundtokilo** : Convertit les livres en kilogrammes.
* **celsiustofahrenheit** : Convertit les degrés Celsius en degrés Fahrenheit.

Seules les requêtes de type POST sont autorisées par l'API Converter.js. Chaque requête doit contenir les informations suivantes dans le corps de la requête au format JSON :

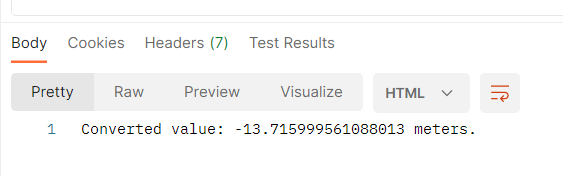
* **type** (type de conversion) : L'endpoint spécifique correspondant à la conversion souhaitée.
* **value** (valeur à convertir) : Un nombre, positif ou négatif, représentant la valeur à convertir.

Exemple:

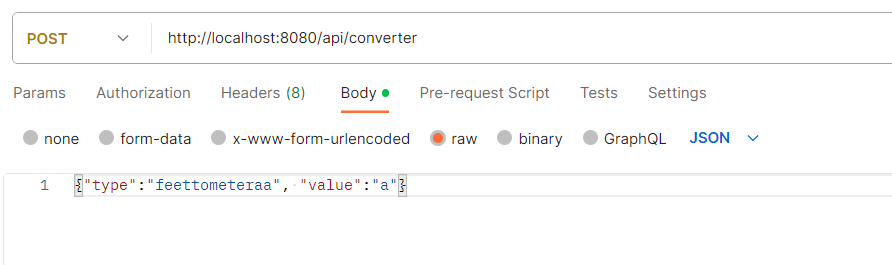
Requête valide :



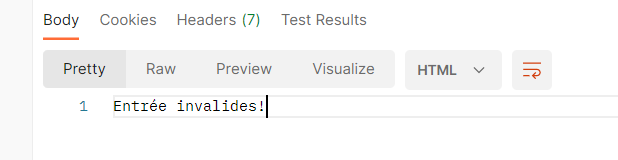
Réponse :



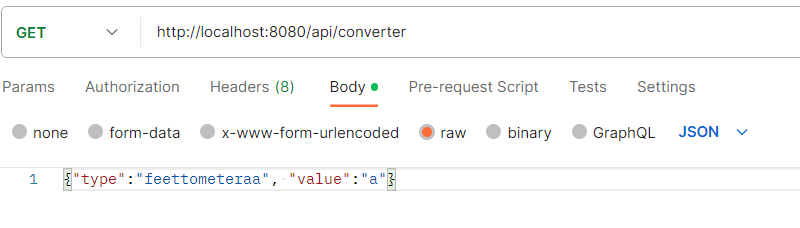
Requête invalide :



Réponse :



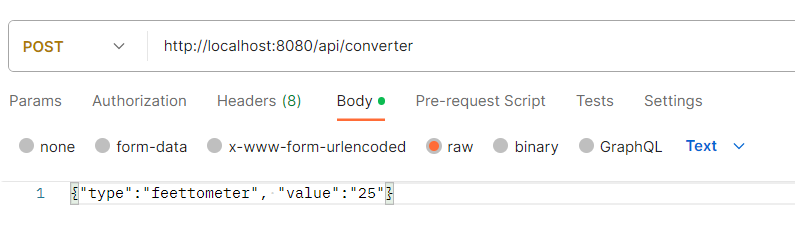
Requête invalide (Type de requête GET) :



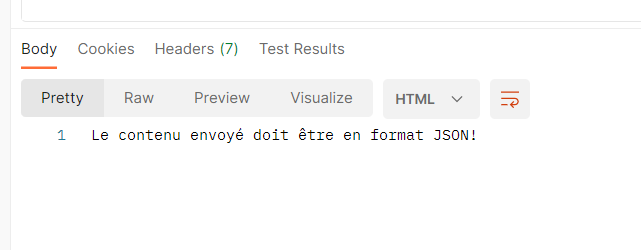
Réponse :



Requête invalide (Le format du body est TEXT) :



Réponse :



# Tâches au choix

## Docker

La mise en œuvre de Docker s'est avérée être une étape relativement aisée, facilitant ainsi le déploiement de notre API sur Internet à l'adresse localhost:8080. L'utilisation de Docker repose sur un script appelé Dockerfile, un composant particulièrement sensible à la casse. Il est crucial de respecter rigoureusement la casse des commandes et des instructions dans ce fichier pour assurer un fonctionnement fluide.

## MongoDB

L'intégration de MongoDB pour enregistrer les journaux d'accès à notre API revêt une importance cruciale pour assurer la traçabilité et permettre une analyse approfondie des interactions. À chaque fois qu'une requête est effectuée à l'API, des données essentielles, telles que l'adresse IP source, la date et l'heure de la requête, ainsi que le type de requête, sont méticuleusement consignées dans une collection spécifique de notre base de données MongoDB Atlas, que nous avons nommée "AccessLog".

Code d’intégration :

Une image contenant texte, capture d’écran

Description générée automatiquement

# Conclusion

En résumé, nous avons réussi à concevoir un API solide baptisée Converter.js, compétente dans la conversion de diverses unités de mesure. L'intégration de Docker a grandement facilité le déploiement de notre API sur Internet, soulignant l'importance cruciale de respecter la casse dans le fichier Dockerfile. L'incorporation de MongoDB a permis de tracer de manière précise les activités de notre API, établissant ainsi une base solide pour l'analyse approfondie de ces interactions.

Le lien vers notre dépôt GitHub simplifie l'accès à tous les fichiers essentiels, tandis que les descriptions détaillées, accompagnées d'exemples simples de requêtes et de réponses, simplifient la compréhension du fonctionnement de l'API.

Les tâches supplémentaires, notamment l'utilisation de Docker et MongoDB, témoignent de notre engagement à rendre le déploiement et la gestion des données plus efficaces, renforçant ainsi la qualité globale de notre projet.