Maïmouna Sarah Diakité  
Gnanda Paule Axelle Kouamé  
Kponaho Anne-Laure Magnané0  
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  
25 septembre 2023

Rapport du projet MongoDB

Lien vers le dépôt github

Lien : https://github.com/Paulaxelle12/projet2\_exploration.git

# Description de l'API

L’API converter.js permet de faire cinq types de conversions d’unité de mesures.

Les endpoints de conversions sont : ‘’feettometer’’, ‘’metertofeet’’, ‘’kilotopound’’, ‘’poundtokilo’’, ‘’celsiustofahrenheit’’

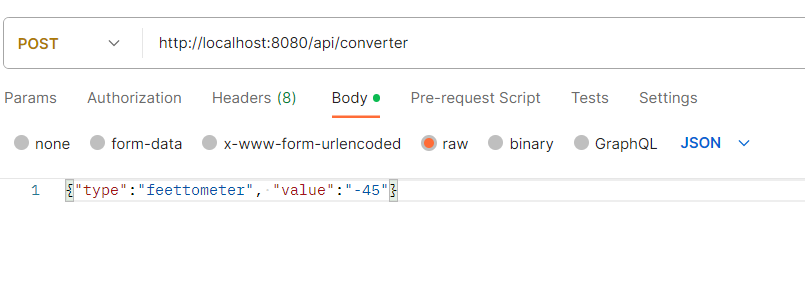
Seule les requêtes POST sont acceptées par l’api, celles-ci doivent contenir le type de conversion à faire donc l’endpoint et la valeur à convertir dans leur « body ». Cette valeur doit être un chiffre qui peut être positif ou négatif.

Les valeurs envoyées par l’utilisateur dans le corps de la requête doivent être en format JSON.

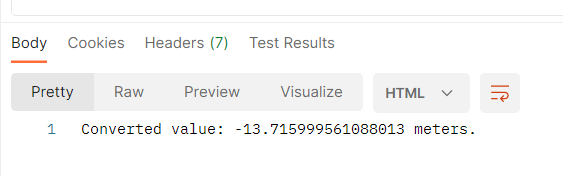
Lorsque le type de la requête, le type de conversion et le chiffre envoyé par l’utilisateur est valide, l’API répond à la requête avec la valeur convertie. Dans le cas contraire, un message d’erreur est renvoyé à l’utilisateur. L’API renvoie aussi un message d’erreur quand le format envoyer n’est pas en JSON

Exemple:

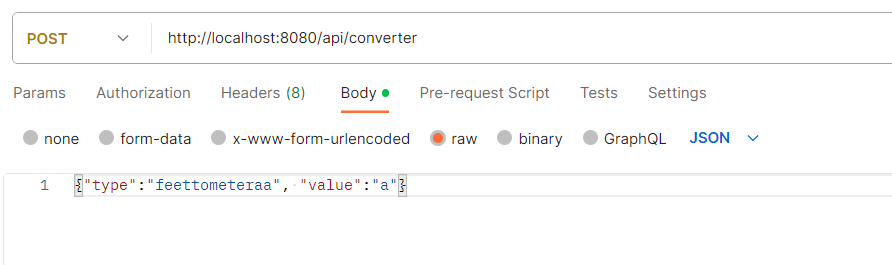
Requête valide :



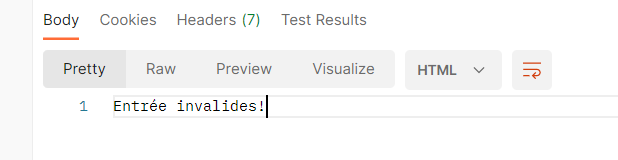
Réponse :



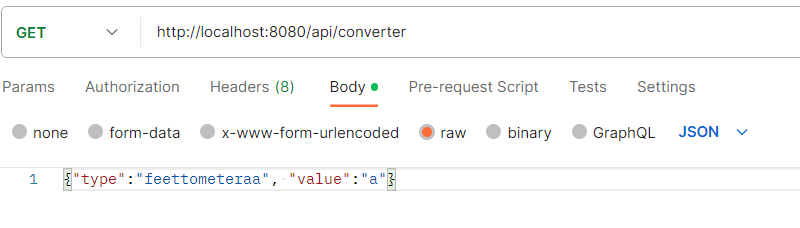
Requête invalide :



Réponse :



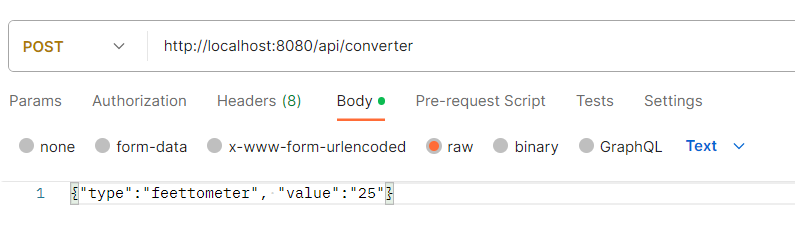
Requête invalide (Type de requête GET) :



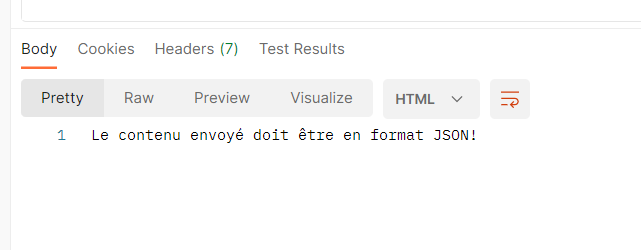
Réponse :



Requête invalide (Le format du body est TEXT) :



Réponse :



# Docker

# MongoDb

Code d’intégration :

// Middleware pour enregistrer les journaux d'accès dans MongoDB

app.use(async (req, res, next) => {

  // Création d'un nouvel objet AccessLog avec les informations de la requête

  const accessLog = new AccessLog({

    ip: req.ip,         // IP source de la requête

    method: req.method, // Méthode de la requête (GET, POST, etc.)

    timestamp: req.timestamp, // Date et heure de la requête

  });

  try {

    // Enregistrement du journal d'accès dans la base de données

    await accessLog.save();

    next(); // Passer au middleware suivant

  } catch (err) {

    // Gestion des erreurs lors de l'enregistrement dans la base de données

    console.error("Erreur lors de l'enregistrement du journal d'accès : ${err}");

    res.status(500).send("Erreur serveur interne");

  }

});

# Conclusion