Techniques Web
Elvis MBONING

Manuel d'utilisation du projet Web
(Partie I serveur)

Groupe 3

Pierre ROCHET
Siyu WANG
Chen SUN
Shuai GAO
Alexandre IVANOV

Master TAL M2
INALCO

Sommaire

1. Lancement de l'application

Cette application, hébergée sur un server Nginx, permet d'exploiter les données clients et les rendre accessibles sur un Postman (web client).

1.1. Système Linux

Requis :

- ➢ nginx : http://hg.nginx.org/nginx.org
- ➢ gunicorn : `pip install gunicorn`
- ➢ postman : https://www.postman.com/

Etape 0 : Créer un environnement virtuel python (optionnel).
Commandes : `source venv/bin/activate`

Etape 1 : Installer les dépendances.
Commandes : `pip3 install -r requirements.txt`

Etape 2 : Mettre le fichier *ceptyconsultant.conf* dans le dossier sites-available de nginx puis créer le chemin symbolique dans le dossier sites-enabled.

Redémarrer ensuite nginx.
Commandes : `sudo service nginx start`

Etape 3 : Lancer le fichier gunicorn_cmd.sh qui contient la commande de lancement de l'application par gunicorn.
Commandes : `sh gunicorn_cmd.sh`

Etape 4 : Aller sur postman pour faire les requêtes.

1.2. Système macOS

Requis :

- ➢ homebrew pour installer nginx : `brew install nginx`
- ➢ gunicorn : `pip3 install gunicorn`
- ➢ postman : téléchargable via le site https://www.postman.com/downloads/

Étape 0 : Créer un environnement virtuel python **:** `pipenv shell`

Étape 1 : Installer les dépendances **:** `pip3 install -r requirements.txt`

Étape 2 : Mettre le fichier `ceptyconsultant.conf` dans le dossier sites-available de nginx: le chemin par défaut de nginx pour Mac OS est : `usr/local/etc/nginx` et il faut créer manuellement le dossier `sites-available` et `sites-enabled`.

Puis créer le chemin symbolique dans le dossier sites-enabled :
`ln -s /usr/local/etc/nginx/sites-available/ceptyconsultant.conf /usr/local/etc/nginx/sites-enabled/ceptyconsultant.conf`

Étape 3 : Tester les configurations de nginx avant de démarrer **:** `sudo nginx -t`
Reload après chaque modification de conf : `sudo nginx -s reload`
Redémarrer nginx : `sudo brew services restart nginx`
Pour regarder le status de nginx : `ps aux | grep nginx`

Étape 4 : Lancer le fichier gunicorn_cmd.sh qui contient la commande de lancement de l'application par gunicorn : `sh gunicorn_cmd.sh`

Étape 5 : Démarrer Postman pour faire des requêtes

2. Requêtes via postman

Toutes les requêtes s'exécutent à l'adresse http://www.localhost/ceptyconsultant.local/.

Pour effectuer une requête, il faut au préalable se connecter avec des identifiants valides.
Des identifiants différents sont disponibles pour chaque collaborateur présent dans le fichier `LISTE_COLLABORATEURS.txt`
Les démonstrations nous avons faites au plus simple pour créer les comptes des utilisateurs :

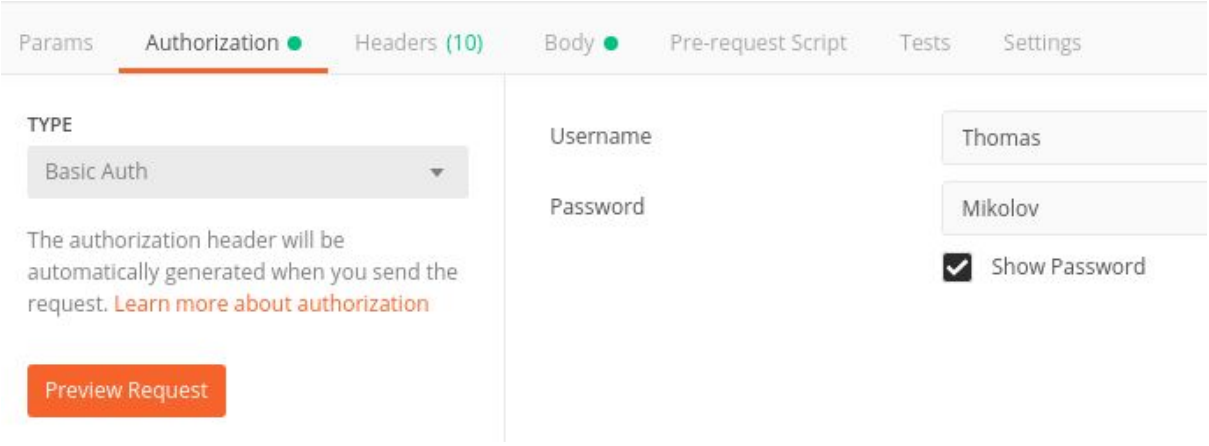
- ➔ Le Username est le prénom de l'utilisateur
- ➔ Le Password est le nom de l'utilisateur

mais les identifiants peuvent être modifiés facilement avec des mots de passe beaucoup plus complexes.

Allez dans l'onglet **Authorization** choisissez le type **Basic Auth** puis entrez les identifiants suivant :

- ➔ Username : Thomas
- ➔ Password : Mikolov

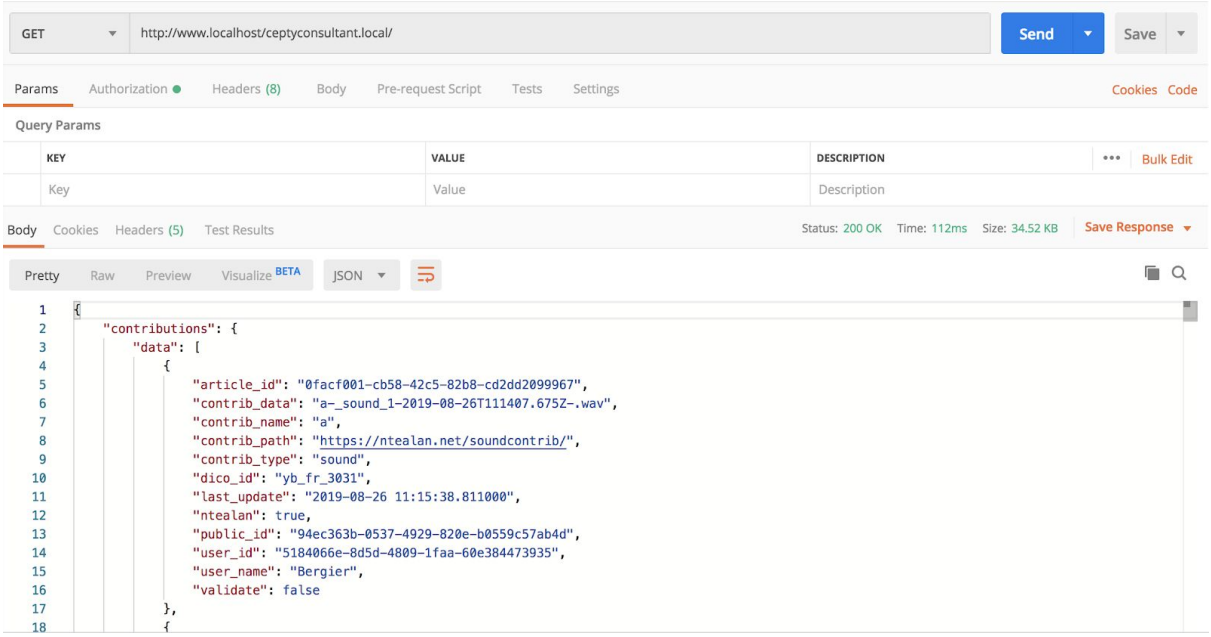
On obtient le résultat ci-dessous :



2.1. Afficher des données

❑ Requête GET

Il s'agit de la requête la plus simple, elle renvoie simplement le contenu de nos données clients c'est à dire le contenu du fichier `DONNEES_CLIENT.txt`



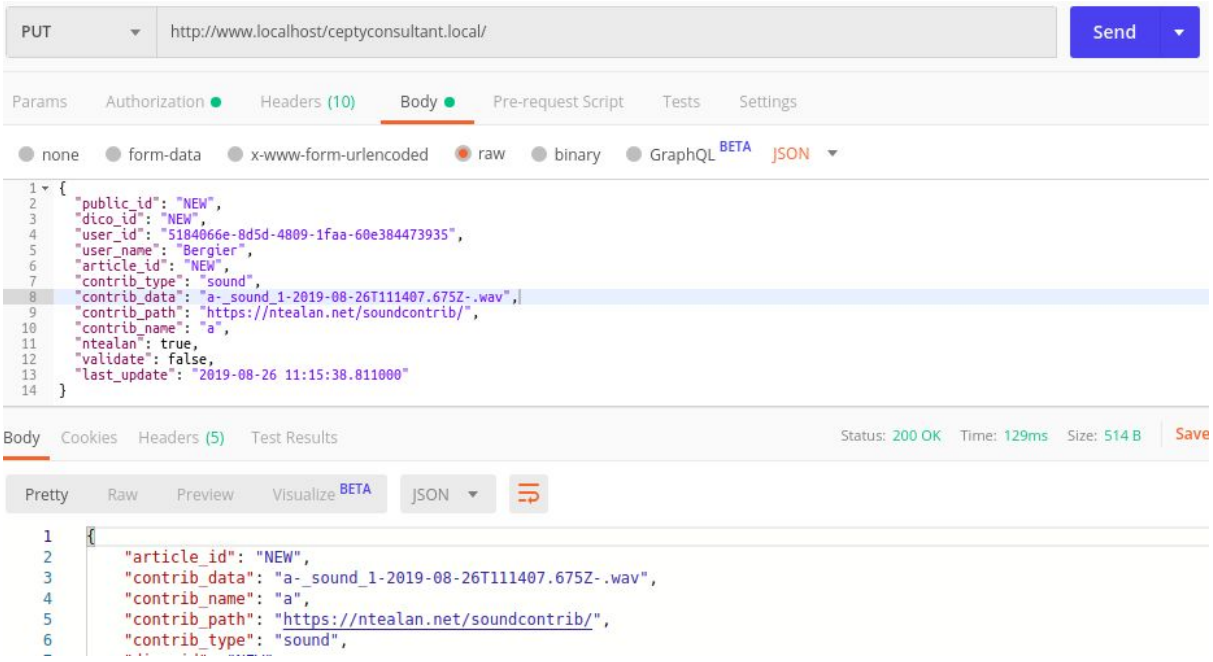
2.2. Ajouter des données

❑ Requête PUT

Cette requête permet d'ajouter de nouvelles données saisies par l'utilisateur.

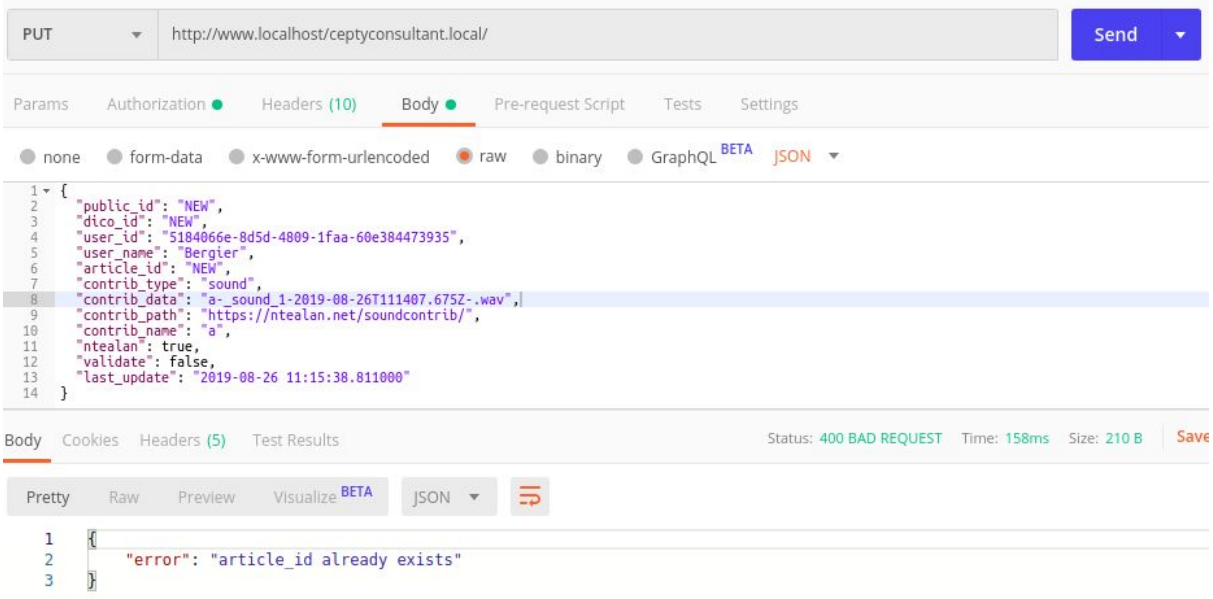
- sélectionner le type de requête **PUT**
- aller dans l'onglet **body**
- choisir le langage json et le format **raw**
- écrire le contenu à ajouter sous format **json**
- lancer la requête

Remarque : Seule la clé **article_id** est obligatoire.
Si tout se passe bien, les nouvelles données seront ajoutées à la liste **data** de nos données clients et les données ajoutées seront renvoyées.



❑ Requête PUT avec erreur

Si vous tentez d'ajouter du contenu avec un **article_id** déjà présent dans les données alors une erreur est renvoyée.



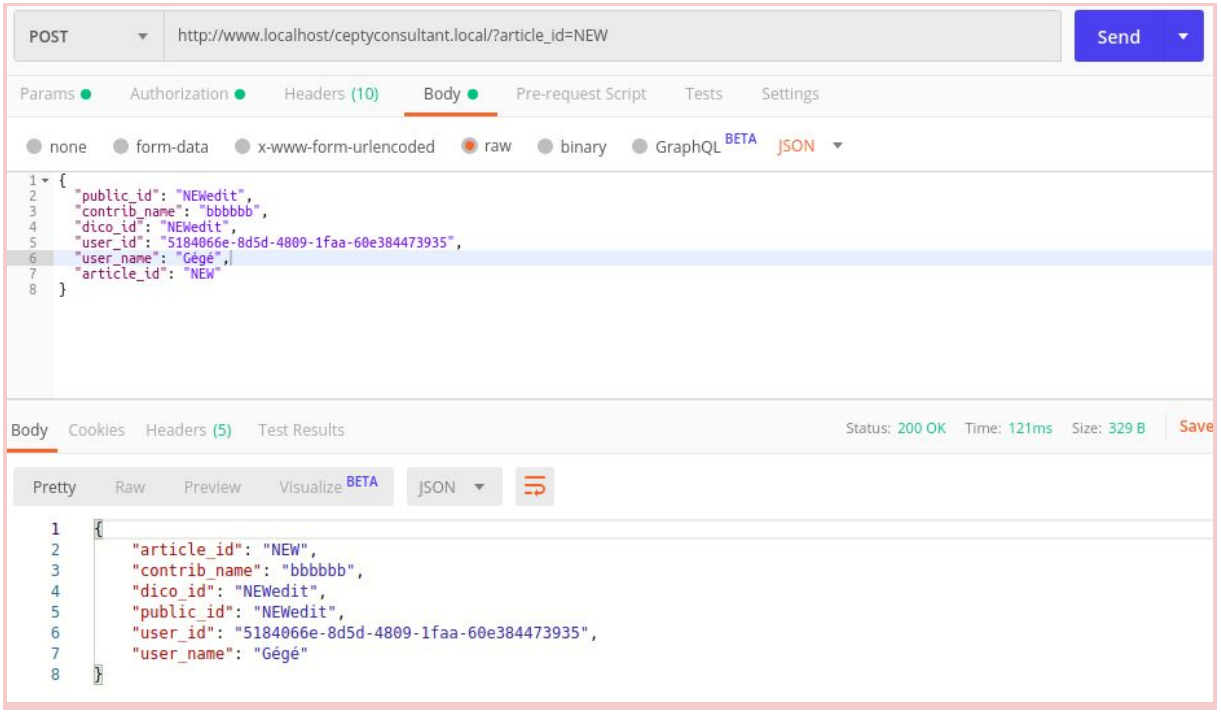
2.3. Modifier des données

❑ Requête POST

Pour modifier des données:

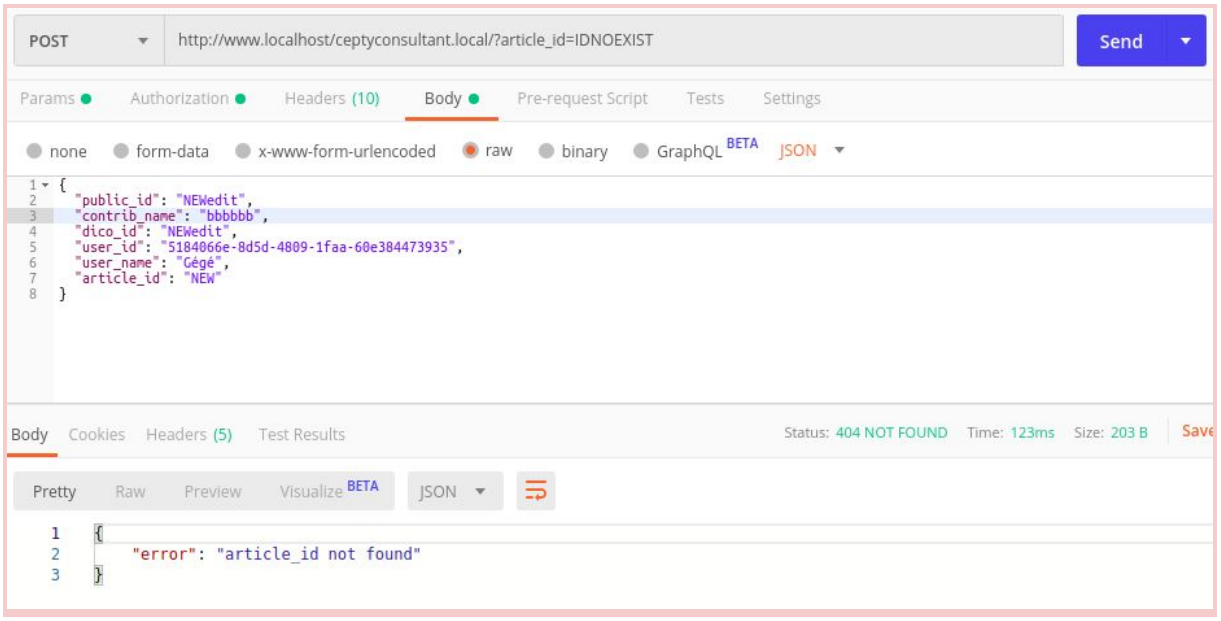
- sélectionner le type de requête **POST**
- aller dans l'onglet **Params**
- le nom de la clé sera toujours **article_id,** la valeur est le nom de l'élément à modifier
- ensuite aller dans l'onglet **body**
- ecrire les données que vous voulez modifier en format **json**.
  Remarque : Si une donnée n'a pas de clé existante alors elle sera créé, si une donnée à une clé existante alors la valeur sera modifiée
- lancer la requête.

Si tout se passe bien, l'élément des données ayant été modifié est renvoyé :



❑ Requête POST avec erreur

Si les données ne contiennent pas l'**article_id** indiqué, une erreur est alors renvoyée.



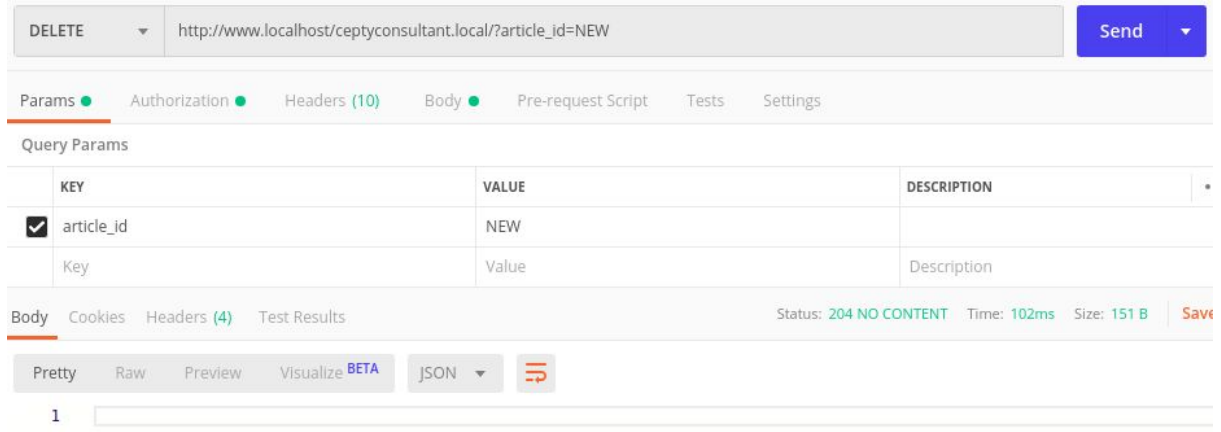
2.4. Supprimer des données

❑ Requête DELETE

Pour supprimer des données :

- sélectionner le type de requête **DELETE**
- aller à l'onglet **Params**
- la clé sera toujours **article_id**, la valeur est la nom de l'**article_id** des données à supprimer
- exécuter la requête

Si rien n'est renvoyé alors la requête a fonctionné et l'élément contenant l'id indiqué est supprimé.



❑ Requête DELETE avec erreur

Si les données ne contiennent pas l'**article_id** indiqué, une erreur est alors renvoyée.