

Manuel d'utilisation du projet web

Partie utilisateur

Groupe 5

LIU Jianying
SHEN Xinyi
WANG Qi

Master 2 TAL IM
INALCO 2020/2021

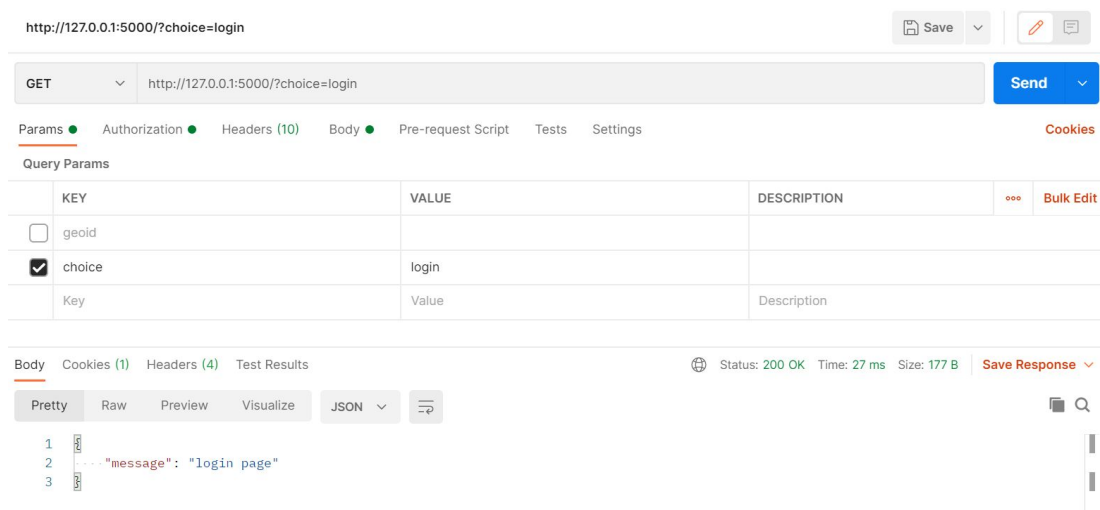
Sommaire

1. Requêtes sur le comptes via Postman	3
1.1 Dans le homepage	3
1.2 Créer un compte	4
1.3 Se connecter au compte	5
1.4 Se déconnecter au compte	6
2. Requêtes sur les données via Postman	7
2.1 afficher des données	7
2.2 rechercher des données	7
2.3 créer de nouvelles données	9
2.4 modifier des données	10

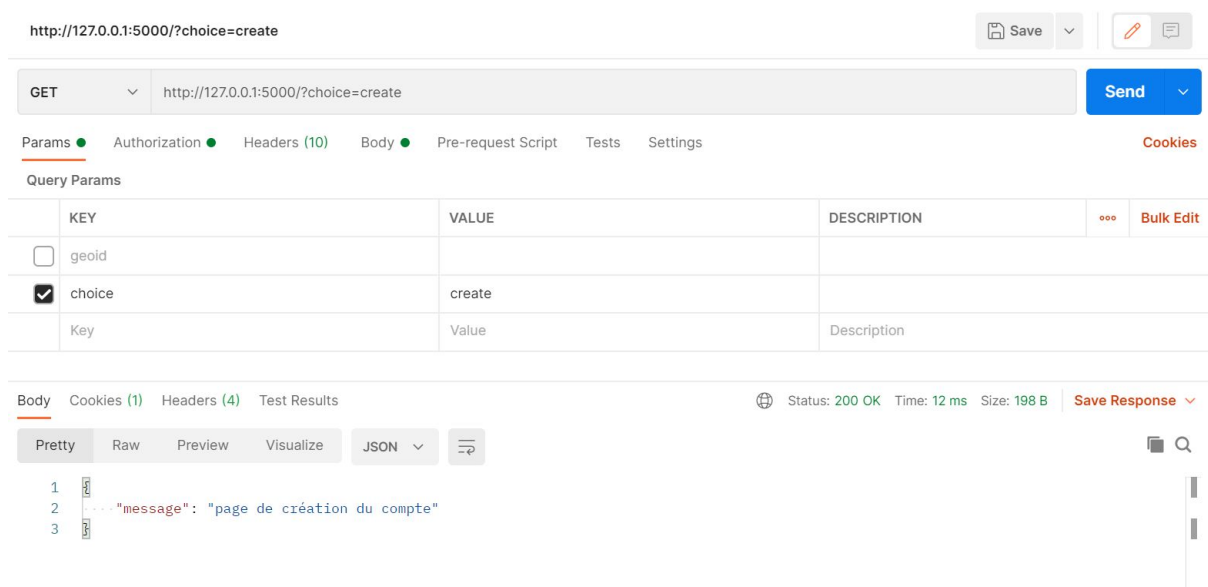
1. Requêtes sur le comptes via Postman

1.1 Dans le homepage

Afin d'aller créer un compte ou se connecter au compte, d'abord, il faut entrer dans la homepage vers le chemin “/” ou “/home” avec la requête “GET”. Dans le bloc Param, si nous choisissons la valeur “login”, il envoie un message “login page” et aussi va à la page “login”.



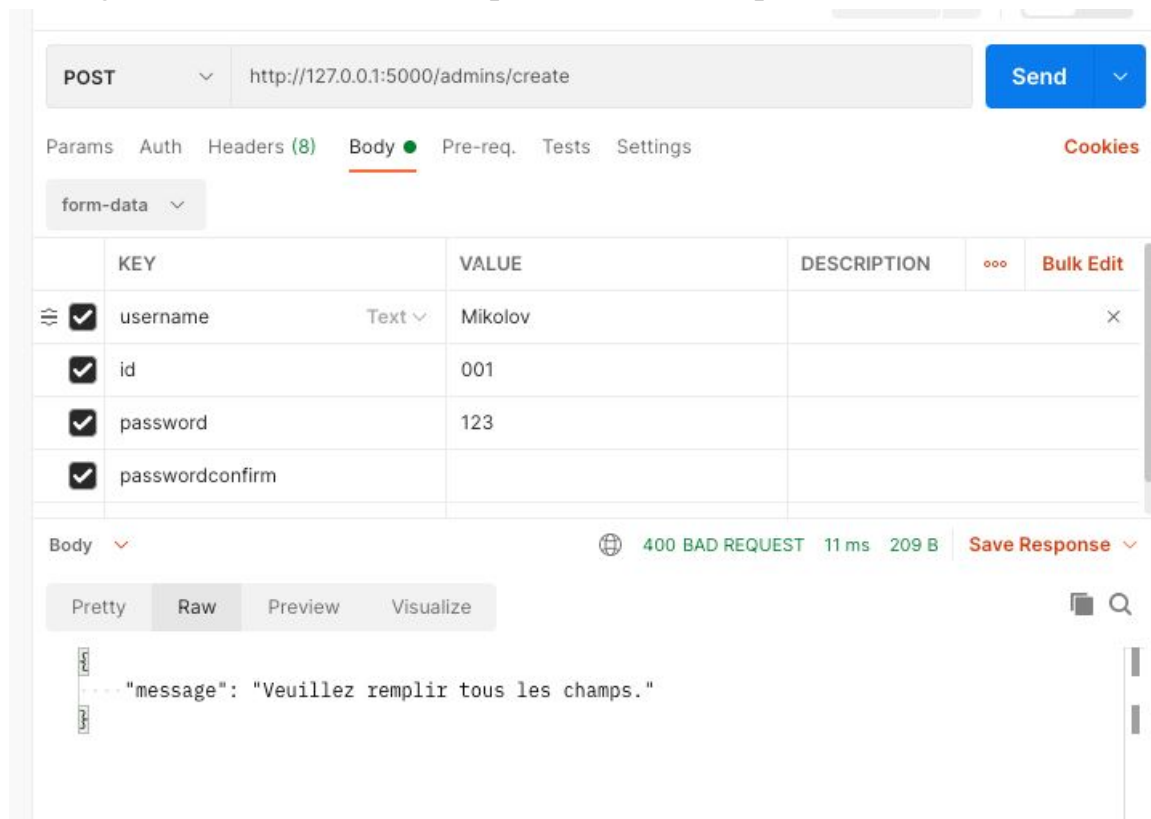
Si nous choisissons la valeur “create”, il envoie un message “page de création du compte” ainsi qu’entre à la page “create”.



1.2 Créer un compte

Afin de créer un compte, il faut tout d'abord aller au chemin vers “/admins/create” avec la requête “POST”. Avec les clés demandées dans la liste d'informations, il faut remplir “username”, “userid” (les deux informations sont dans le fichier “user.json”), “password”, “passwordconfirm”(les deux sont créés par le client) dans le bloc “Body”.

Si nous n'avons pas rempli toutes les informations demandées, il envoie un message d'erreur “ Veuillez remplir tous les champs.”



Si c'est un client qui n'est pas dans la liste du “user”, il envoie un message d'erreur “Désolée, vous n'êtes pas notre collaborateur, vous ne pouvez pas créer un compte.”

Si le collaborateur a déjà eu un compte, il envoie un message d'erreur “ Vous avez déjà un compte.”

Si les deux mots de passe ne sont pas identiques, il envoie un message d'erreur “Les deux mots de passe remplis doivent être identiques.”

Enfin, s'il n'y a aucune erreur, le compte va être bien créé et enregistré dans la base de données, et nous voyons aussi un message “Votre compte admin a bien été créé.”

1.3 Se connecter au compte

Afin de se connecter au compte, il faut tout d’abord aller au chemin vers “/admins/login” avec la requête “POST”. Avec les clés demandées dans la liste d’informations , il faut remplir “id” et “username” (les deux informations sont dans le fichier user.json) et “password” créé par le client dans le bloc “Body”. Si tous les champs sont bien remplis, il envoie un message “Username: {user}”.

http://127.0.0.1:5000/admins/login

POST http://127.0.0.1:5000/admins/login

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

Field	Value
modification_date	h
username	Bergier
id	002
password	54321

Body Cookies (1) Headers (6) Test Results Status: 200 OK Time: 1102 ms Size: 561 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "username": "Bergier"
3 }
```

Dans la situation de connexion, nous voulons aller aux chemins “/geonames” et les autres chemins “/geonames/{autres}”, nous avons le droit d’y aller.

GET http://127.0.0.1:5000/geonames?name=Paris

Params Authorization Headers (10) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

Field	Value
modification_date	h
username	Bergier
id	002
password	54321

Body Cookies (1) Headers (5) Test Results Status: 200 OK Time: 221 ms Size: 175.17 KB Save Response

Pretty Raw Preview Visualize JSON

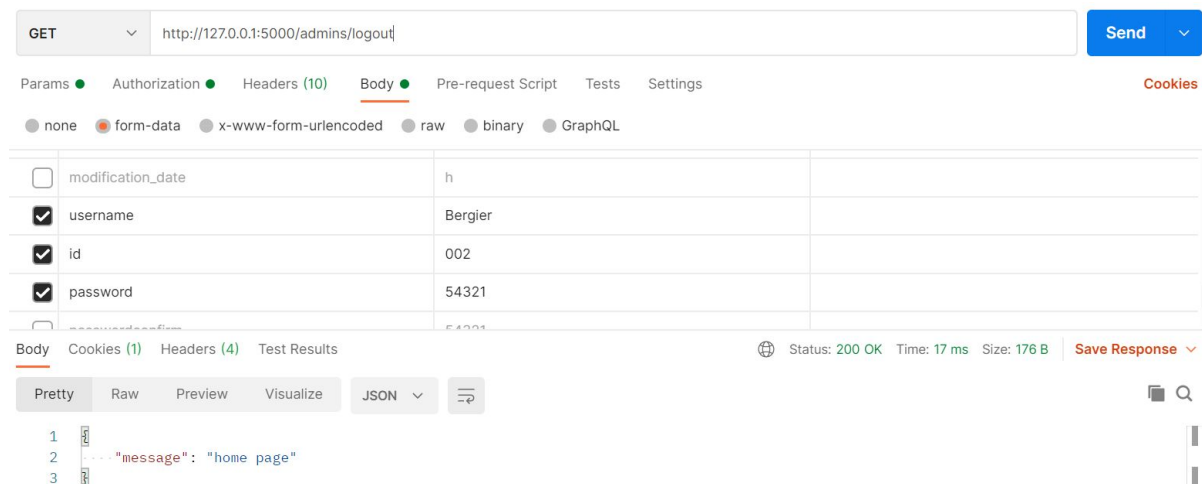
```
1 {
2   "2968496": {
3     "geonameid": "2968496",
4     "name": "Villeparisis",
5     "latitude": "48.94208",
6     "longitude": "2.61463"
7   },
8   "2968496": {
9     "geonameid": "2968496",
10    "name": "Villeparisis",
11    "latitude": "48.94208",
12    "longitude": "2.61463"
13   }
14 }
```

Par contre, si le client ne remplit pas tous les champs, il envoie un message d'erreur : "Remplissez tous les champs, s'il vous plaît!"

Si "username", "id" ou "password" a un problème (par exemple, le mot de passe n'est pas correct), il envoie un message d'erreur : "Vérifiez votre nom, votre id ou votre mot de passe, s'il vous plaît !"

1.4 Se déconnecter au compte

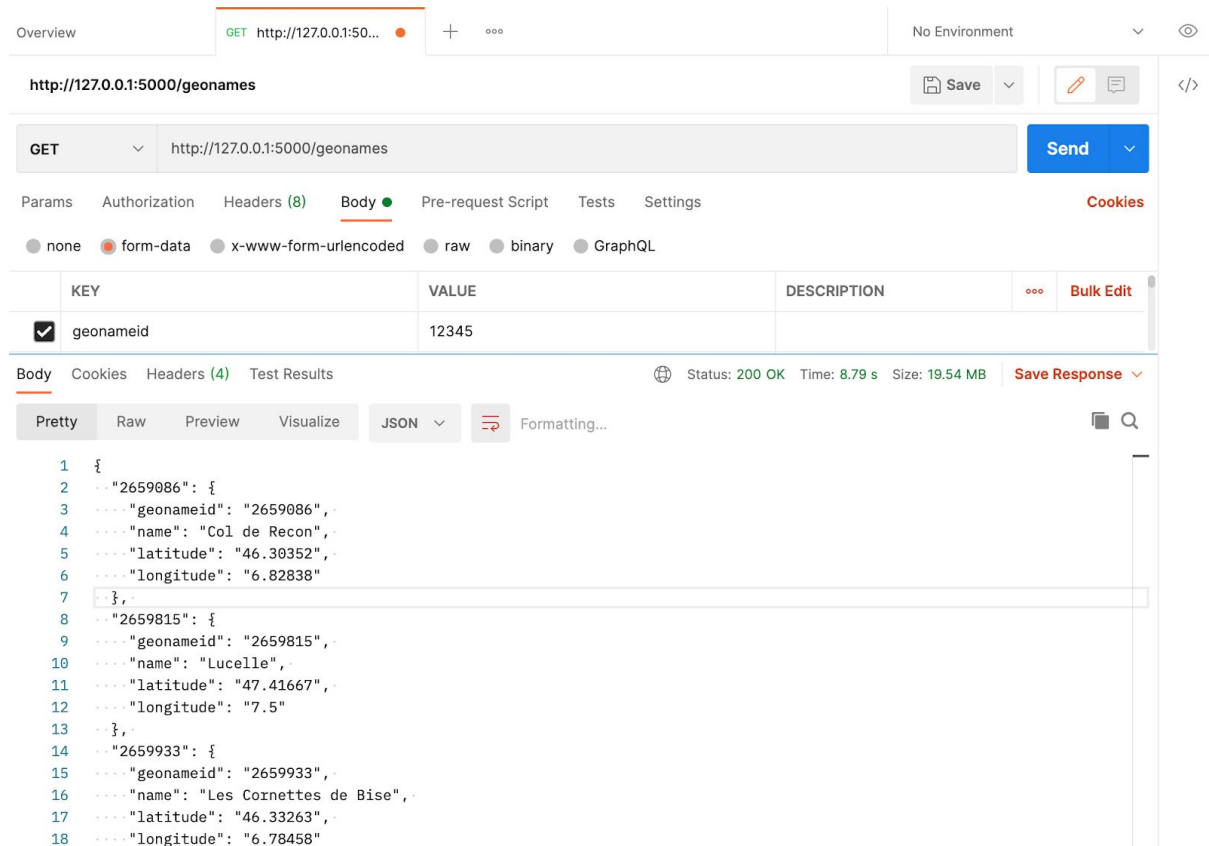
Pour se déconnecter du compte, il faut que le client soit déjà bien connecté au compte. Vers le chemin "/admins/logout" avec la requête "Post", il va au "homepage" et envoie le message : "homepage". Et si nous voulons nous connecter aux chemins "geonames" ou "geonames/{autres}" dans la situation de déconnexion, il va aussi au "homepage" et envoie le message : "homepage".



2. Requêtes sur les données via Postman

2.1 afficher des données

Afin d’obtenir toutes les données géographiques, il suffit d’envoyer la requête “GET” pour le chemin “/geonames”. La capture ci-dessous montre les informations principales des locations (geonames, name, latitude, longitude).



2.2 rechercher des données

La requête “GET” de la même page consiste à chercher une location avec une adresse indiquée sous le format “/geonames?name=[nom_location]”. Ainsi, nous pouvons insérer une valeur pour la clé “name” pour changer de l’adresse comme montré dessus.

Les locations contenant le mot-clé seront envoyées une fois on envoie la requête “GET”. Et les informations des locations sont aussi principales comme dessus (geonames, name, latitude, longitude).

http://127.0.0.1:5000/geonames?name=Paris

GET http://127.0.0.1:5000/geonames?name=Paris

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION	Bulk Edit
geonameid	12345		

Body Cookies Headers (4) Test Results Status: 200 OK Time: 257 ms Size: 175.16 KB Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "2968496": {
3      "geonameid": "2968496",
4      "name": "Villeparisis",
5      "latitude": "48.94208",
6      "longitude": "2.61463"
7    },
8    "2968815": {
9      "geonameid": "2968815",
10     "name": "Paris",
11     "latitude": "48.8534",
12     "longitude": "2.3486"
13   },
14   "2970479": {
15     "geonameid": "2970479",
16     "name": "Paris 15 Vaugirard",
17     "latitude": "48.8412",
18     "longitude": "2.3003"

```

Si on souhaite d'obtenir toutes les informations de cette location, il suffit de changer le chemin de "name" par "geonameid", soit "/geonames?geoid=[geonameid]" avec la requête "GET".

Postman

File Edit View Help

Home Workspaces Reports Explore Search Postman Upgrade

GET http://127.0.0.1:5000/geonames?geoid=2968496

GET http://127.0.0.1:5000/geonames?geoid=2968496

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

name	Paris	
geoid	2968496	
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 27 ms Size: 840 B Save Response

Pretty Raw Preview Visualize JSON

```

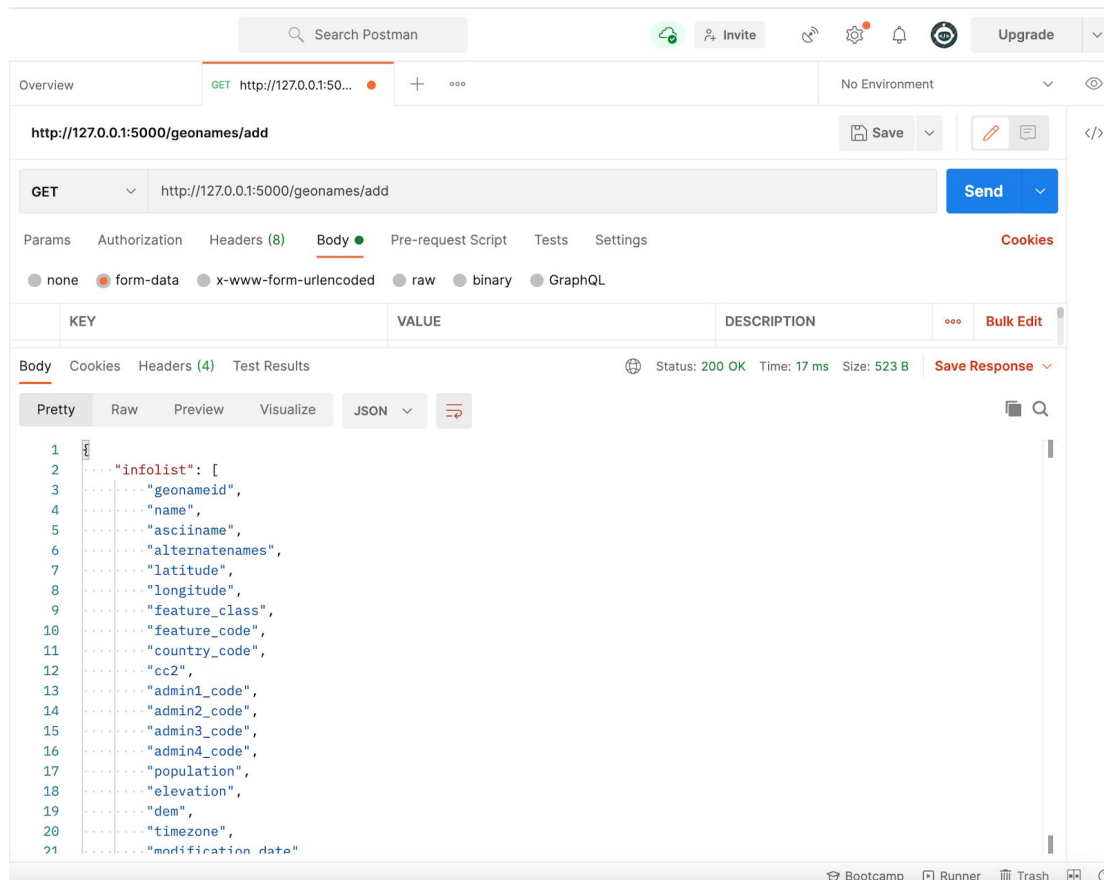
1  {
2    "2968496": {
3      "Geoname ID": "2968496",
4      "Name": "Villeparisis",
5      "Ascii Name": "Villeparisis",
6      "Alternate Names": "Vil'parizi,Villeparisis,Вильпаризи",
7      "Latitude": "48.94208",
8      "Longitude": "2.61463",
9      "Feature class": "P",
10     "Feature code": "PPL",
11     "Country code": "FR",
12     "CC2": ""

```

Find and Replace Console Bootcamp Runner Trash

2.3 créer de nouvelles données

Afin de pouvoir ajouter de nouvelles données, il faut d’abord changer du chemin vers “/geonames/add” avec la requête “GET”, et il va renvoyer automatiquement une liste d’informations (infolist) à remplir.



Avec les clés demandées dans la liste d’informations (infolist), il faut remplir avec toutes les valeurs correspondantes dans le bloc “Body”. Une fois on réussira à fournir toutes les valeurs d’une nouvelle location, un message va renvoyer indiquant cette location bien ajoutée dans la base de données.

POST http://127.0.0.1:5000/geonames/add

Save Send

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

	KEY	VALUE	DESCRIPTION		Bulk Edit
<input checked="" type="checkbox"/>	geonameid	112233			
<input checked="" type="checkbox"/>	name	Qingdao			
<input checked="" type="checkbox"/>	asciiname	Qingdao			
<input checked="" type="checkbox"/>	alternatenames	Qingdao			
<input checked="" type="checkbox"/>	latitude	36.50			
<input checked="" type="checkbox"/>	longitude	120.20			
<input checked="" type="checkbox"/>	feature_class	CN			
<input checked="" type="checkbox"/>	feature_code	CN			
<input checked="" type="checkbox"/>	country_code	CN			

Body Cookies Headers (4) Test Results Status: 201 CREATED Time: 19 ms Size: 247 B Save Response

Pretty Raw Preview Visualize JSON

```

1  {
2    "message": "****Qingdao(112233) est bien ajoutée dans la base de données.****"
3  }

```

2.4 modifier des données

Il faudrait avoir un identifiant (geonameid) pour consulter toutes les informations de la location avant de modifier ou supprimer des données. Pour accéder à cette page d'informations, on peut changer du chemin vers “/geonames/<geonameid>” avec la requête “GET”.

GET http://127.0.0.1:5000/geonames/112233

Save

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION
Key	Value	Description

Body Cookies Headers (4) Test Results Status: 200 OK Time: 74 ms Size: 680 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "112233": {
3     "Geoname ID": "112233",
4     "Name": "Qingdao",
5     "Ascii Name": "Qingdao",
6     "Alternate Names": "Qingdao",
7     "Latitude": "36.5",
8     "Longitude": "120.2",
9     "Feature class": "CN",
10    "Feature code": "CN",
11    "Country code": "CN",
12    "CC2": "FR",
13    "Admin1 code": "1",
14    "Admin2 code": "2",
15    "Admin3 code": "3",
16    "Admin4 code": "4",
17    "Population": "950",
18    "Elevation": "900",
19    "Dem": "1234",
20    "Timezone": "Asia/Beijing",
21    "Modification date": "2020-02-12"
22  }
23 }
```

Si cet identifiant n'existe pas dans la base de données, il envoie un message d'erreur.

http://127.0.0.1:5000/geonames/303

Save

Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

KEY	VALUE	DESCRIPTION
<input checked="" type="checkbox"/> geonameid	30375	
<input checked="" type="checkbox"/> name	Paris Store	
<input checked="" type="checkbox"/> asciiname	Annecy-le-Vieux	
<input checked="" type="checkbox"/> alternatenames	Annecy-le-Vieux	
<input checked="" type="checkbox"/> latitude	23.45	

Body Cookies Headers (4) Test Results Status: 404 NOT FOUND Time: 234 ms Size: 235 B Save Response

Pretty Raw Preview Visualize JSON

```

1 {
2   "error": "La base de données n'a pas d'infos de cette location (303)"
3 }
```

Maintenant, nous pouvons modifier des informations d'une location existante avec la requête "PUT". Il faut s'assurer que tous les champs sont bien sélectionnés et remplis avec les informations originales et que les informations que nous souhaitons changer soient bien modifiées. Un message va être envoyé pour indiquer le succès de la modification après l'envoi de la requête.

The screenshot shows a REST client interface with a PUT request to `http://127.0.0.1:5000/geonames/112233`. The request body is a JSON object with the following fields:

Field	Value
geonameid	112233
name	Beijing
asciiname	Beijing
alternatenames	Beijing
latitude	36.50
longitude	120.20

The response status is 200 OK. The response body is a JSON object:

```
{  "message": "****Les informations de Beijing(112233) est bien modifiées.****"}
```

Attention, le geonameid ne peut pas être changé. Un message d'erreur va être renvoyé si on modifie le geonameid.

The screenshot shows a REST client interface with a PUT request to `http://127.0.0.1:5000/geonames/3037540`. The request body is a JSON object with the following fields:

Field	Value
geonameid	30375
name	Paris Store
asciiname	Annecy-le-Vieux
alternatenames	Annecy-le-Vieux
latitude	23.45

The response status is 200 OK. The response body is a JSON object:

```
{  "error": "Désolé, le geonameid ne peut pas être changé."}
```

2.5 supprimer des données

On reste à la même page que celle de la modification des données, soit le chemin “/geonames/<geonameid>”, mais avec la requête “DELETE” dans le but de supprimer des données. Comme la suppression est pour toutes les informations d’une location, il suffit d’envoyer la requête “DELETE”, et nous pouvons obtenir le message.

