

**AGRHYMET CCR-AOS**



Documentation APIs

**Table des matières**

[**I.** **Généralités** 1](#_Toc181008151)

[II. Documentation des APIs 2](#_Toc181008152)

[1. Production agricole 2](#_Toc181008153)

[2. Gestion des indicateurs 12](#_Toc181008154)

[3. Division administrative 20](#_Toc181008155)

[4. Gestion des paramètres 21](#_Toc181008156)

[5. Gestion des utilisateurs 26](#_Toc181008157)

# **Généralités**

Un API est un ensemble de définitions et de protocoles qui facilite la création et l'intégration d'applications.  On compare parfois les API à un contrat entre un utilisateur d'informations et un fournisseur d'informations, qui permet de définir le contenu demandé par le consommateur (l'appel) et le contenu demandé au producteur (la réponse).

Dans l’implémentation des APIs, l'architecture REST est généralement retenue pour obtenir des données à partir du web en raison de sa flexibilité, de sa rapidité et de sa simplicité.

Bien qu'elle soit théoriquement compatible avec n'importe quel protocole ou format de données, l'architecture REST utilise le plus souvent le protocole HTTP et transfère les données en utilisant [JSON](https://www.w3schools.com/js/js_json_intro.asp) (JavaScript Object Notation).

Le générateur d’indicateurs de la nouvelle plateforme ECOAGRIS est une application web, ainsi pour la sécurité de cette application nous avons choisit le mode **Stateless** pour la sécurité.

Les échanges entre le frontend et le backend se font par des APIs REST privés, on utilise le JSON WEB TOKEN a cet effet. A chaque communication une authentification par clé privé s’effectue entre le backend et le frontend.

Le développement du générateur d’indicateur de la nouvelle plateforme ECOAGRIS est toujours en cours alors l’authentification par token n’est pas activé.

# **Documentation des APIs**

## **Production agricole**

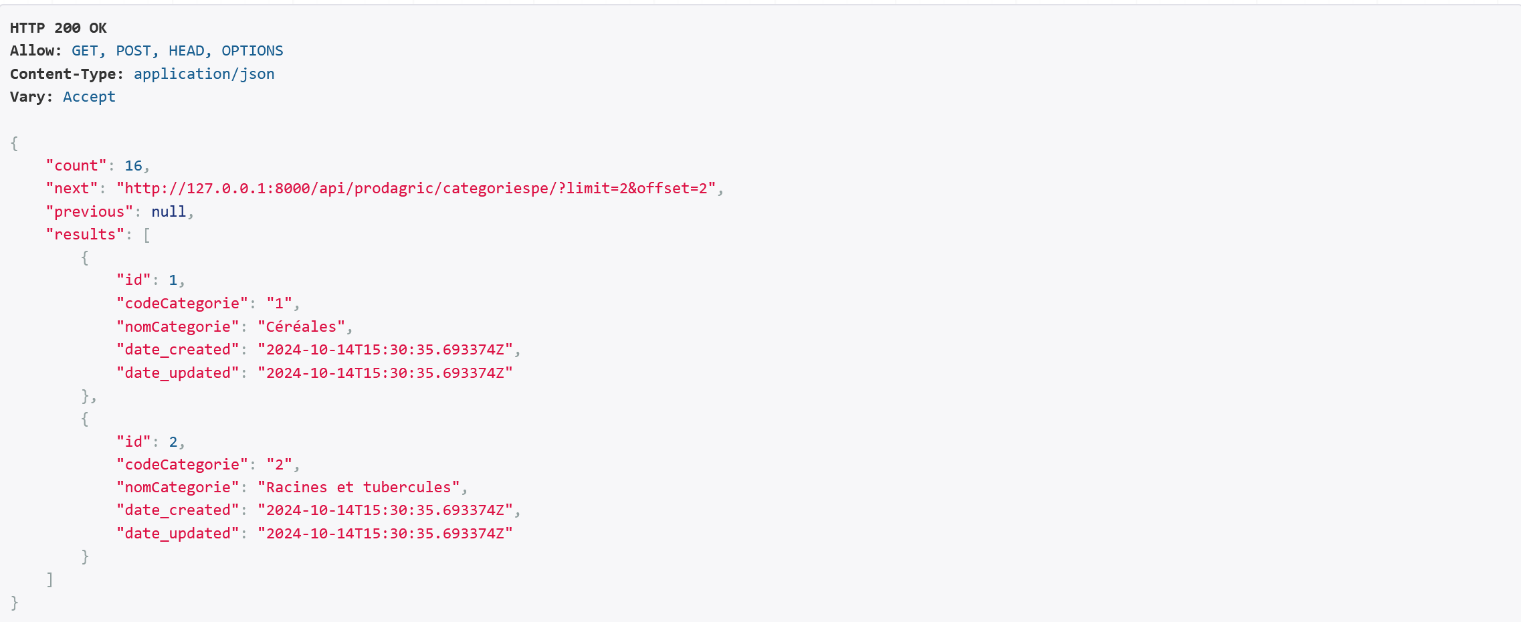
**Speculation :** [**http://127.0.0.1:8000/api/prodagric/speculation/**](http://127.0.0.1:8000/api/prodagric/speculation/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les spéculations



**categoriespe:** [**http://127.0.0.1:8000/api/prodagric/categoriespe/**](http://127.0.0.1:8000/api/prodagric/categoriespe/):

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les categories des speculatiions



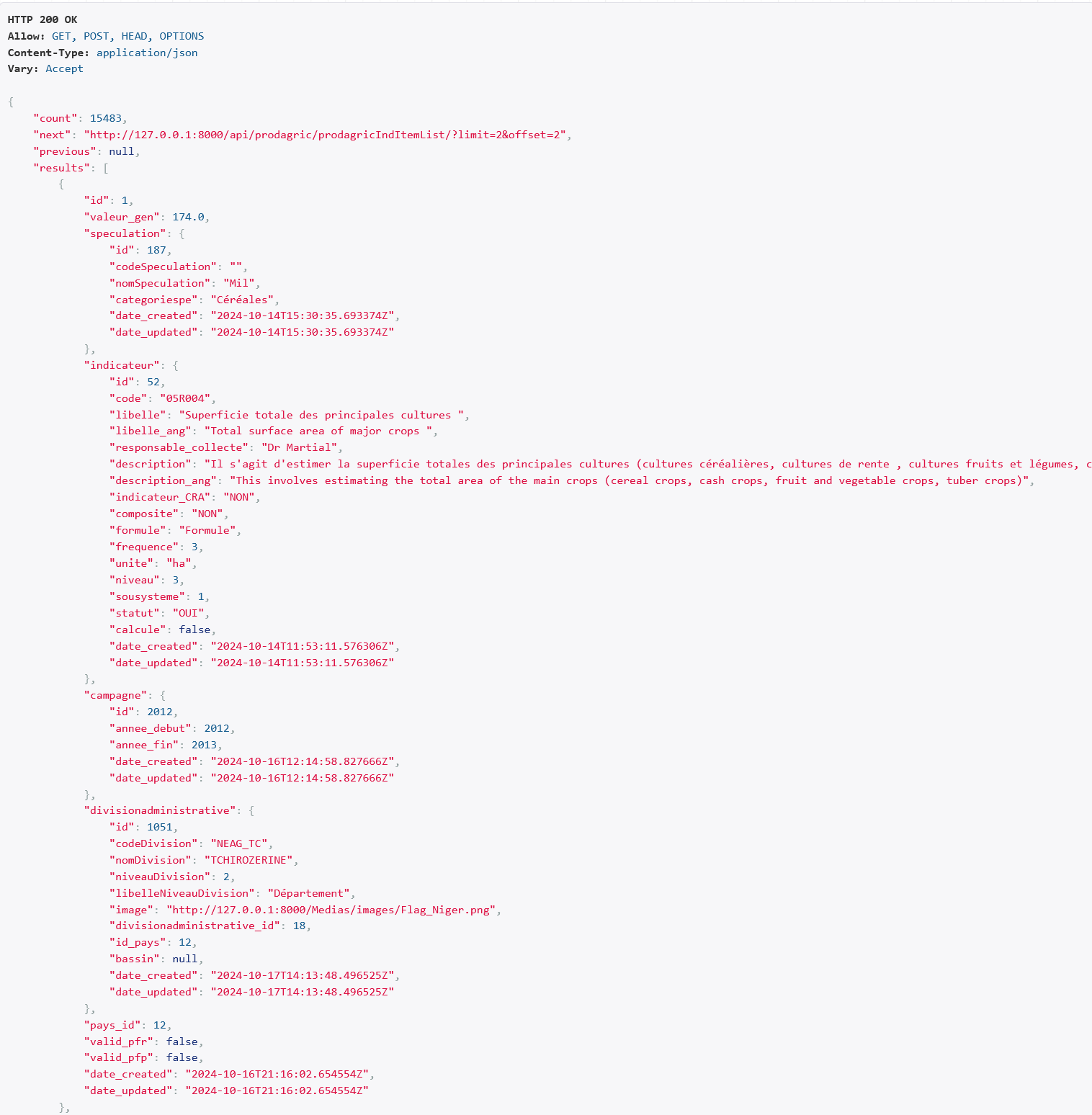
**prodagricIndItem :** [**http://127.0.0.1:8000/api/prodagric/prodagricIndItem/**](http://127.0.0.1:8000/api/prodagric/prodagricIndItem/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les données des indicateurs



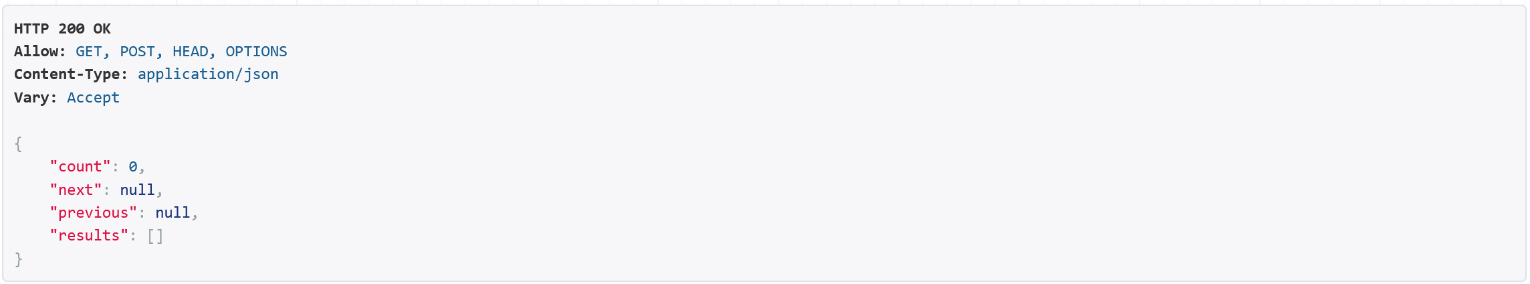
**prodagricIndItemList:** [**http://127.0.0.1:8000/api/prodagric/prodagricIndItemList/**](http://127.0.0.1:8000/api/prodagric/prodagricIndItemList/)**:**

API qui permet de récupérer la liste des données des indicateurs avec leurs détails



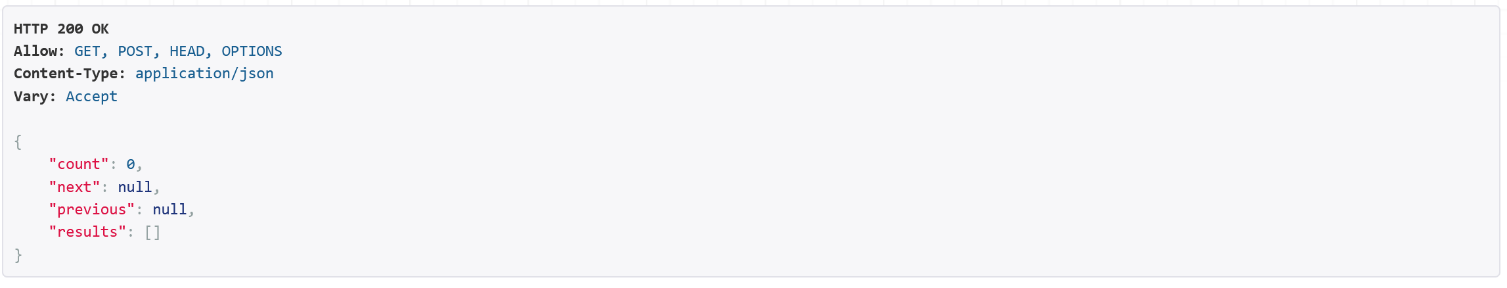
**prodagricVarItem :** [**http://127.0.0.1:8000/api/prodagric/prodagricVarItem/**](http://127.0.0.1:8000/api/prodagric/prodagricVarItem/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les données des variables



**prodagricVarItemList:** [**http://127.0.0.1:8000/api/prodagric/prodagricVarItemList/**](http://127.0.0.1:8000/api/prodagric/prodagricVarItemList/)**:**

API qui permet de récupérer la liste des données des variables avec leurs détails



**get\_prodagric:** [**http://127.0.0.1:8000/api/prodagric/get\_prodagric/**](http://127.0.0.1:8000/api/prodagric/get_prodagric/)**:**

API qui permet de récupérer la liste des données des indicateurs agricoles



Cette API peut prendre comme paramètres : **campagne**, **pays**, **indicateurs**

* **campagne :** année de production
* **pays :** pays de production
* **indicateurs :** indicateur agricole (rendement, superficie…)

**Exemple :** <http://127.0.0.1:8000/api/prodagric/get_prodagric/2017-12-53/>

[**http://127.0.0.1:8000/api/prodagric/varQuantiteProd**](http://127.0.0.1:8000/api/prodagric/varQuantiteProd) **:**

API qui permet de récupérer la variable Quantité de production dans la plateforme PREGEC Bilan alimentaire

Cet API prend en paramètres : **campagne**, **speculation**, **indicateurs**

* **campagne :** année de production
* **pays :** pays de production
* **indicateurs :** indicateur agricole (rendement, superficie…)

**Exemple :**

<http://127.0.0.1:8000/api/prodagric/varQuantiteProd?campagne=2024&speculation=9&pays=12>



[**http://127.0.0.1:8000/api/prodagric/varSuperficieCult**](http://127.0.0.1:8000/api/prodagric/varSuperficieCult) **:**

API qui permet de récupérer la variable Superficie de production dans la plateforme PREGEC Bilan alimentaire.

Cet API prend en paramètres : **campagne**, **speculation**, **indicateurs**

**Exemple :**

<http://127.0.0.1:8000/api/prodagric/varSuperficieCult?campagne=2024&speculation=9&pays=12>



[**http://127.0.0.1:8000/api/prodagric/indRendement**](http://127.0.0.1:8000/api/prodagric/indRendement) **:**

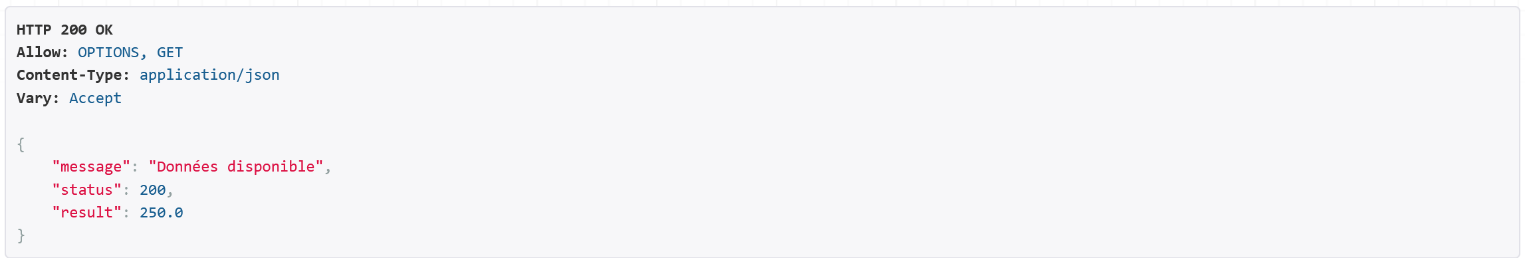
API qui permet de calculer l’indicateur Rendement agricole

Cet API prend en paramètres : **quantite**, **superficie**

* **quantite**: quantité de production des principales cultures (mil, riz)
* **superficie :** superficie de production des principales cultures (mil, riz)

**Exemple :**

[**http://127.0.0.1:8000/api/prodagric/indRendement?quantite=250000&superficie=1000**](http://127.0.0.1:8000/api/prodagric/indRendement?quantite=250000&superficie=1000)



[**http://127.0.0.1:8000/api/prodagric/indicateurItemByAnne**e](http://127.0.0.1:8000/api/prodagric/indicateurItemByAnnee) :

API qui permet de récupérer les données des indicateurs agricoles par année

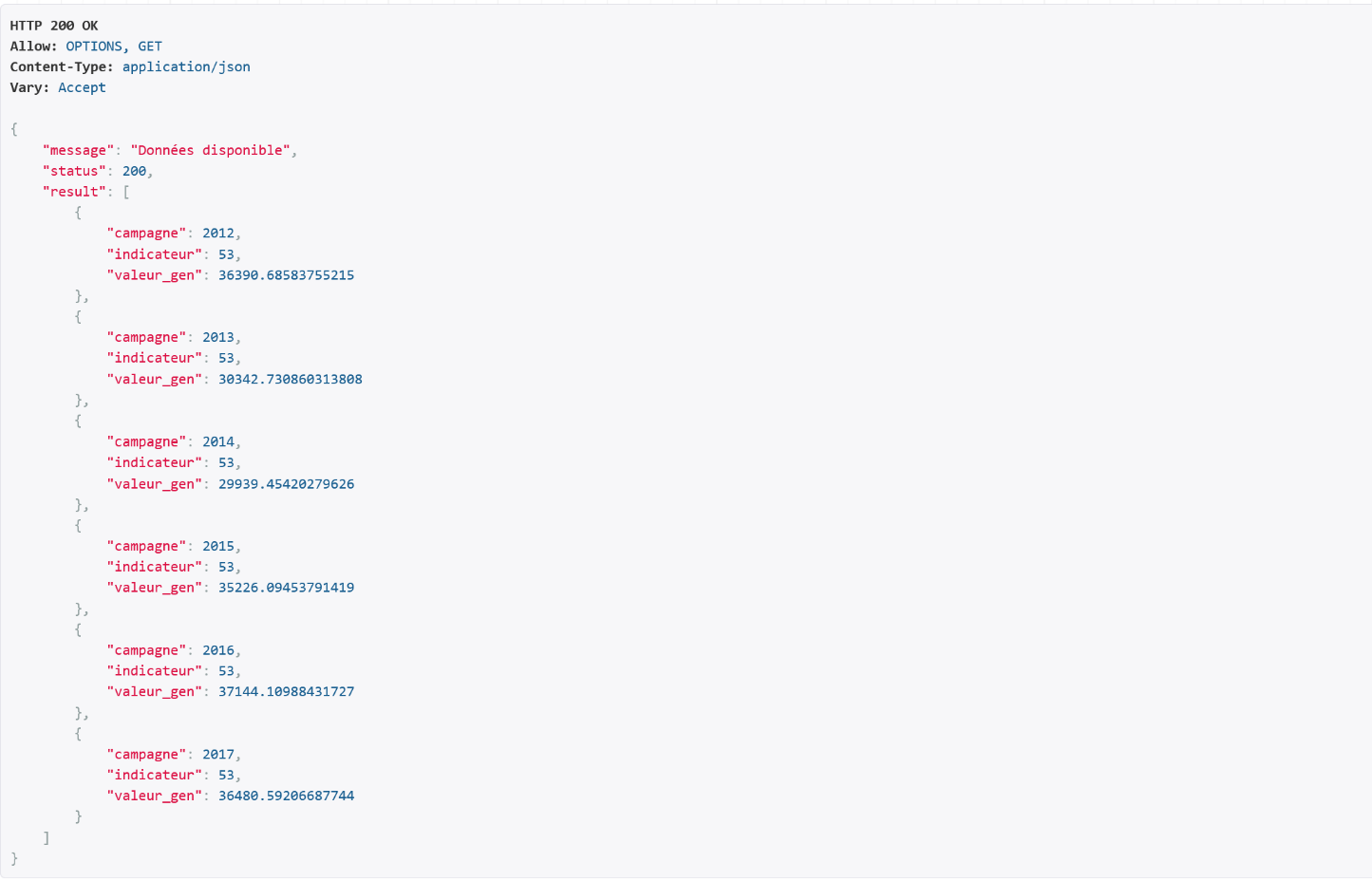
Ce APIs retourne les données d’un indicateur par spéculation par pays.

Il prend en paramètres : **indicateur**, **speculation**, **pays**

* **indicateur** : indicateur agricole
* **speculation**: principale culture (mil, riz, sorgho)
* **pays :** pays de production

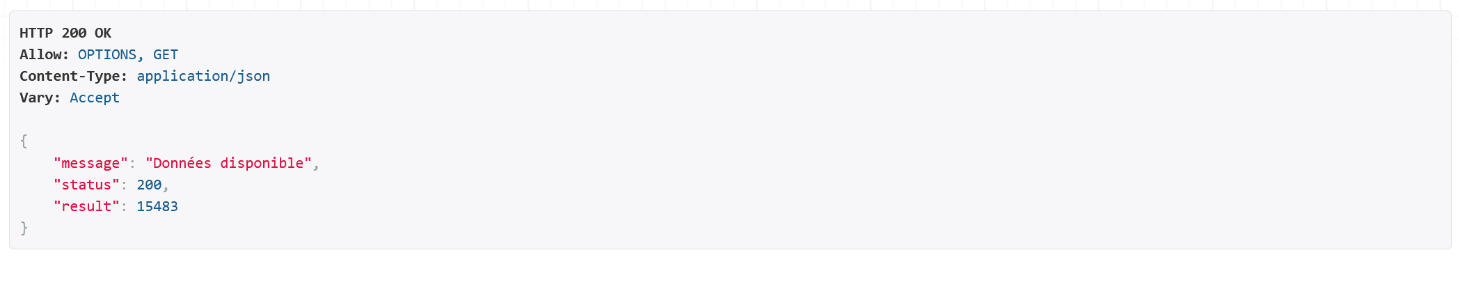
**Exemple :**

<http://127.0.0.1:8000/api/prodagric/indicateurItemByAnnee?params_indicateur=53&params_speculation=187&params_pays=12>



[**http://127.0.0.1:8000/api/prodagric/nbr\_data\_indicateur**](http://127.0.0.1:8000/api/prodagric/nbr_data_indicateur) **:**

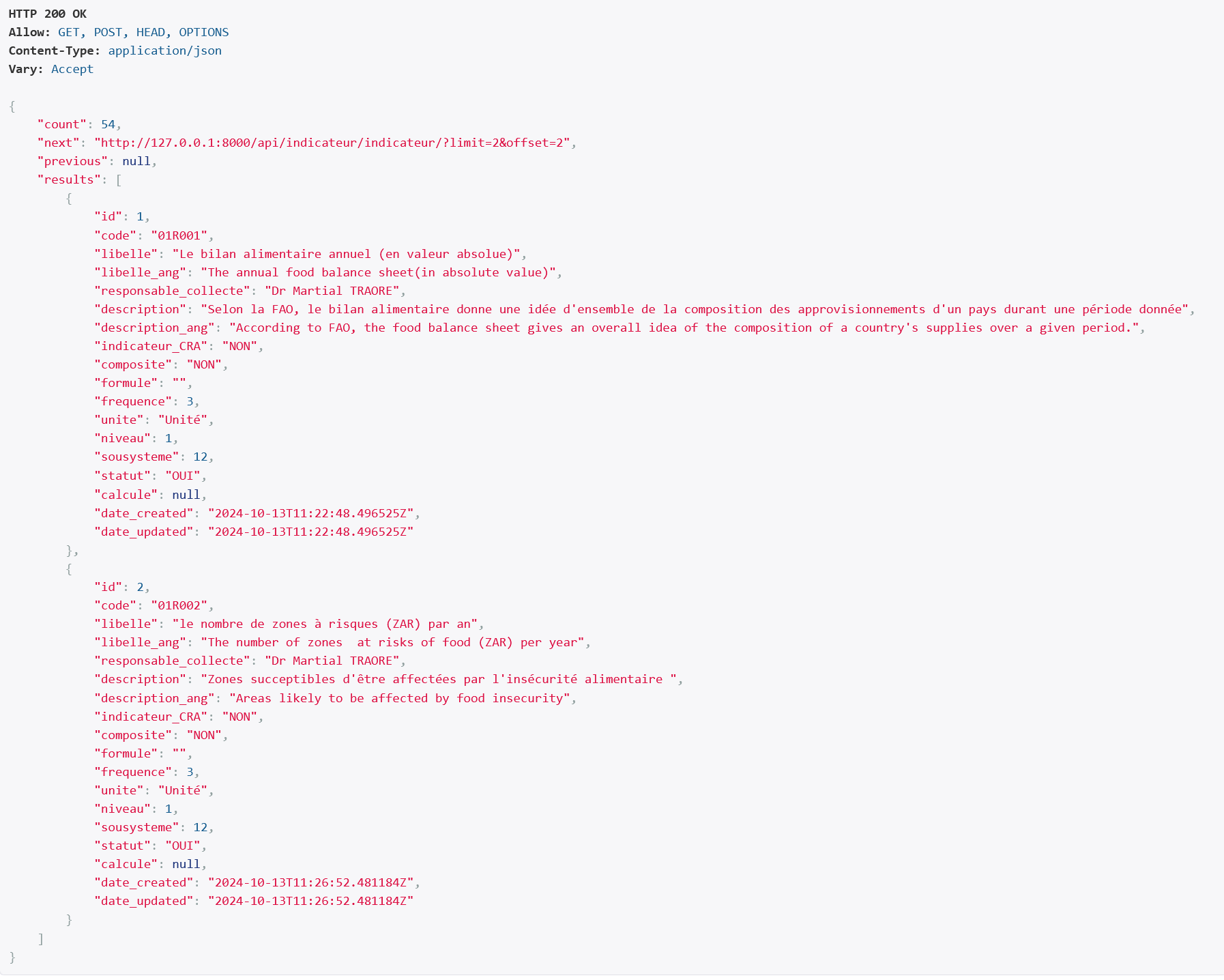
API qui permet d’avoir le nombre total des données des indicateurs



## **Gestion des indicateurs**

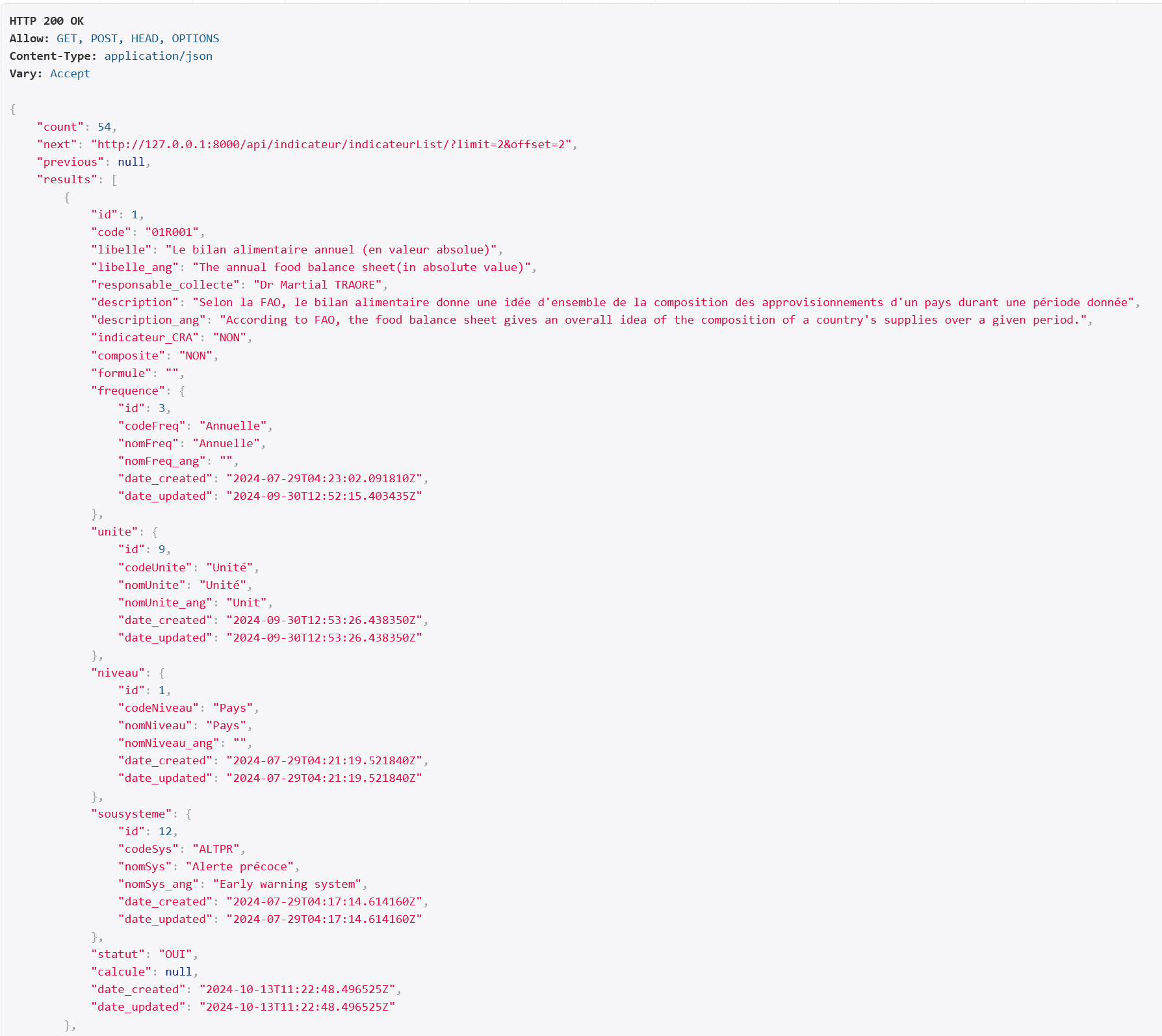
**indicateur:** [**http://127.0.0.1:8000/api/indicateur/indicateur/**](http://127.0.0.1:8000/api/indicateur/indicateur/):

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les indicateurs



**indicateurList:** [**http://127.0.0.1:8000/api/indicateur/indicateurList/**](http://127.0.0.1:8000/api/indicateur/indicateurList/)**:**

API qui permet de récupérer la liste des indicateurs avec leurs détails



**config\_formule:** [**http://127.0.0.1:8000/api/indicateur/config\_formule/**](http://127.0.0.1:8000/api/indicateur/config_formule/)**:**

API qui permet de mettre a jour les données d’un indicateurs avec sa formule



**variable:** [**http://127.0.0.1:8000/api/indicateur/variable/**](http://127.0.0.1:8000/api/indicateur/variable/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les variables



**variableList:** [**http://127.0.0.1:8000/api/indicateur/variableList/**](http://127.0.0.1:8000/api/indicateur/variableList/)**:**

API qui permet de récupérer la liste des variables avec leurs détails



**indicateurVariable :**[**http://127.0.0.1:8000/api/indicateur/indicateurVariable/**](http://127.0.0.1:8000/api/indicateur/indicateurVariable/)**:**

API qui permet d’associer des variables a un indicateur



**get\_indicateur\_sousysteme:** [**http://127.0.0.1:8000/api/indicateur/get\_indicateur\_sousysteme/**](http://127.0.0.1:8000/api/indicateur/get_indicateur_sousysteme/)**:**

API qui permet de récupérer les indicateurs d’un sous-système



Il prend en paramètre : **sous\_systeme**

* **sous\_systeme**: identifiant d’un sous-système

**Exemple :**

<http://127.0.0.1:8000/api/indicateur/get_indicateur_sousysteme/1/>

**get\_variable\_sousysteme :<http://127.0.0.1:8000/api/indicateur/get_variable_sousysteme/> :**

API qui permet de récupérer les variables d’un sous-système



Il prend en paramètre : **sous\_systeme**

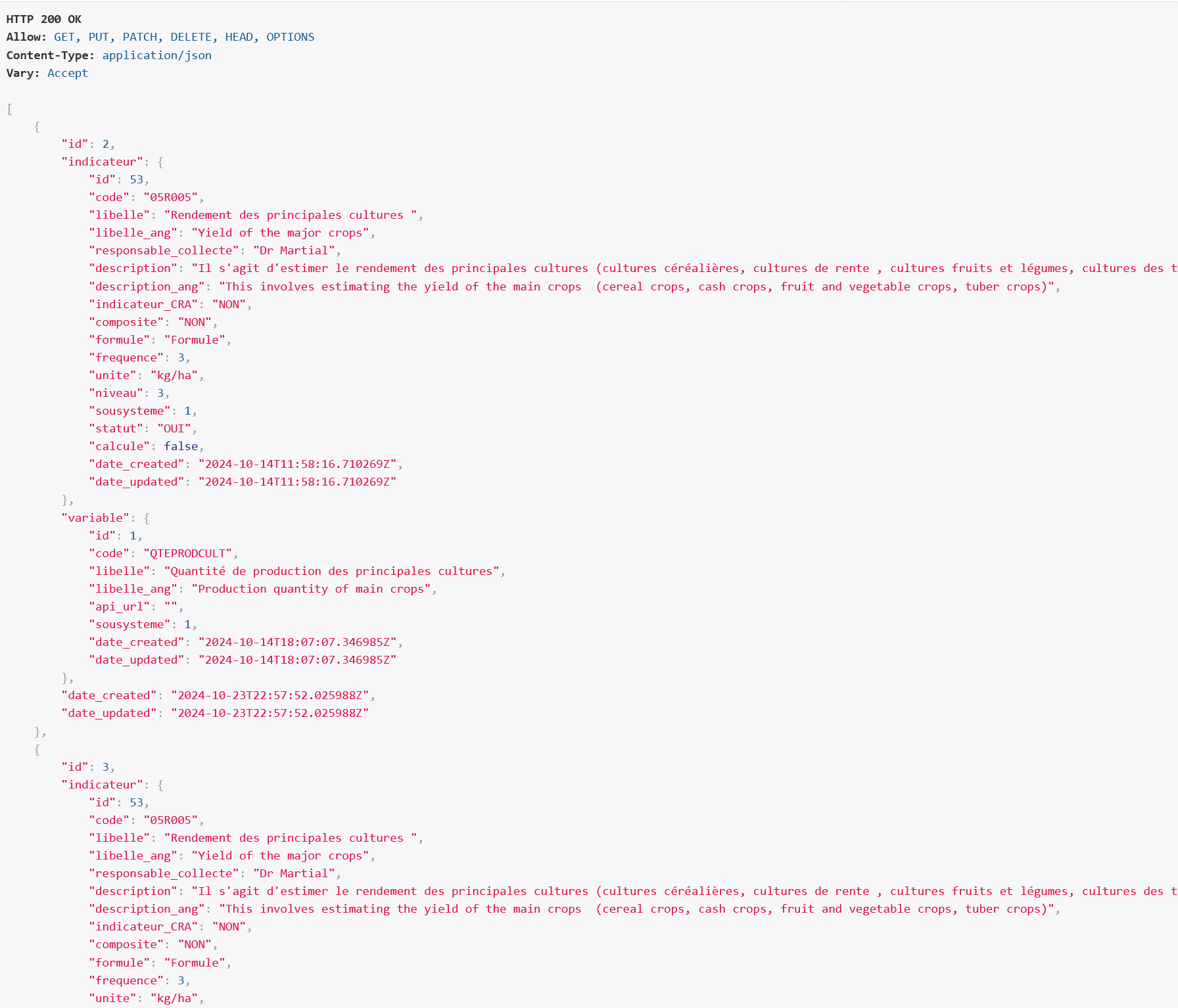
* **sous\_systeme**: identifiant d’un sous-système

**Exemple :**

<http://127.0.0.1:8000/api/indicateur/get_variable_sousysteme/1>

**get\_indicateur\_variable:**[**http://127.0.0.1:8000/api/indicateur/get\_indicateur\_variable/**](http://127.0.0.1:8000/api/indicateur/get_indicateur_variable/)**:**

API qui permet de récupérer les variables d’un indicateur



Il prend en paramètre : **indicateur**

* **indicateur**: identifiant d’un indicateur

**Exemple :**

<http://127.0.0.1:8000/api/indicateur/get_indicateur_variable/53>

## **Division administrative**

**Zones :** [**http://127.0.0.1:8000/api/divadmin/zones/**](http://127.0.0.1:8000/api/divadmin/zones/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les zones

**Bassins :** [**http://127.0.0.1:8000/api/divadmin/bassins/**](http://127.0.0.1:8000/api/divadmin/bassins/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les bassins

**divisionsAdministratives :** [**http://127.0.0.1:8000/api/divadmin/divisionsAdministratives/**](http://127.0.0.1:8000/api/divadmin/divisionsAdministratives/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les divisions administratives



**zonesDivisionsAdministratives :**[**http://127.0.0.1:8000/api/divadmin/zonesDivisionsAdministratives/**](http://127.0.0.1:8000/api/divadmin/zonesDivisionsAdministratives/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les zones divisions administratives

**getonedivision:** [**http://127.0.0.1:8000/api/divadmin/getonedivision/**](http://127.0.0.1:8000/api/divadmin/getonedivision/)**:**

API qui permet de récupérer une division administrative par le nom

**pays-list:** [**http://127.0.0.1:8000/api/divadmin/pays-list/**](http://127.0.0.1:8000/api/divadmin/pays-list/)**:**

API qui permet de récupérer uniquement les pays dans les divisions administratives

## **Gestion des paramètres**

**Frequences :** [**http://127.0.0.1:8000/api/parametre/frequences/**](http://127.0.0.1:8000/api/parametre/frequences/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les fréquences



**Unites :** [**http://127.0.0.1:8000/api/parametre/unites/**](http://127.0.0.1:8000/api/parametre/unites/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les unités



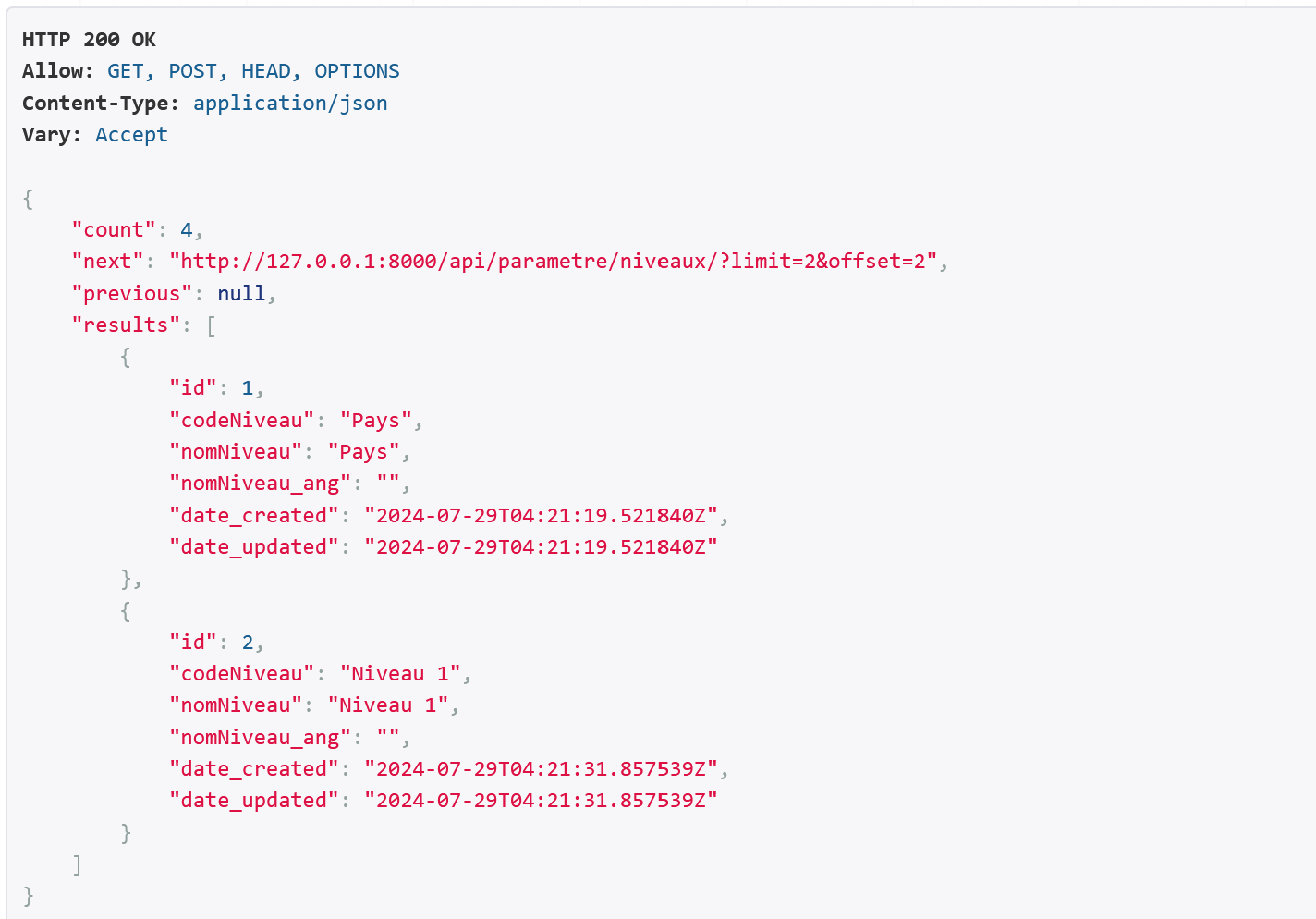
**campagnes: "**[**http://127.0.0.1:8000/api/parametre/campagnes/**](http://127.0.0.1:8000/api/parametre/campagnes/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les campagnes



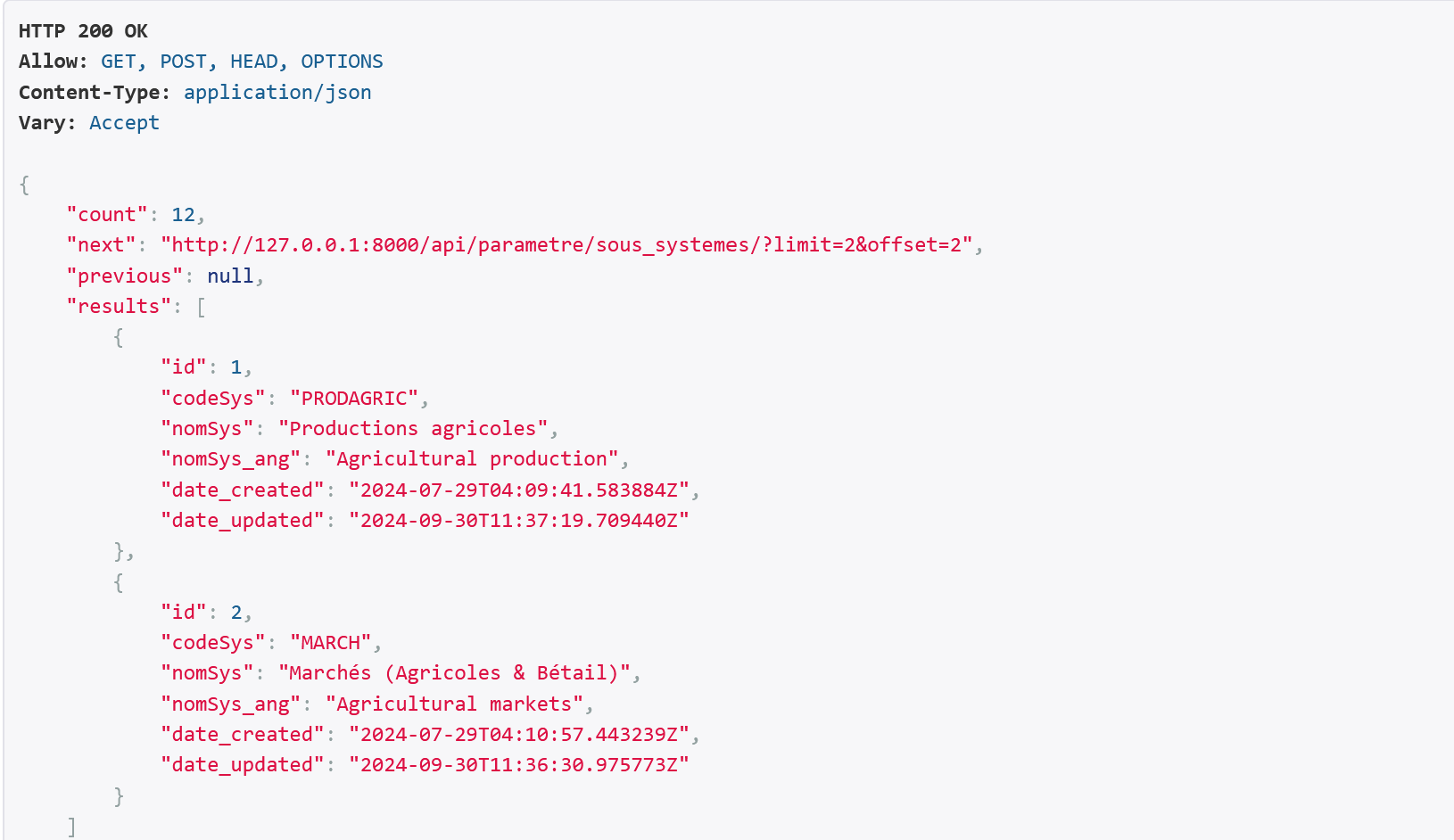
**niveaux:** [**http://127.0.0.1:8000/api/parametre/niveaux/**](http://127.0.0.1:8000/api/parametre/niveaux/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les niveaux



**sous\_systemes:** [**http://127.0.0.1:8000/api/parametre/sous\_systemes/**](http://127.0.0.1:8000/api/parametre/sous_systemes/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les sous-sytème



## **Gestion des utilisateurs**

**Logs :** [**http://127.0.0.1:8000/api/gestuser/logs/**](http://127.0.0.1:8000/api/gestuser/logs/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur l’historique des actions des utilisateurs

**Privileges :** [**http://127.0.0.1:8000/api/gestuser/privileges/**](http://127.0.0.1:8000/api/gestuser/privileges/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les droits d’accès ou privilèges.

**Users :** [**http://127.0.0.1:8000/api/gestuser/users/**](http://127.0.0.1:8000/api/gestuser/users/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les utilisateurs.



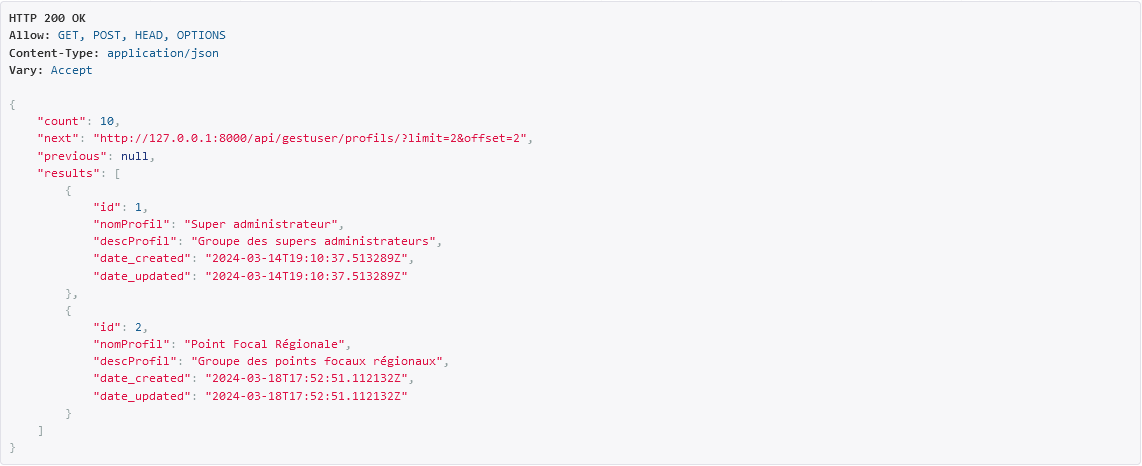
**Fonctionnalites :** [**http://127.0.0.1:8000/api/gestuser/fonctionnalites/**](http://127.0.0.1:8000/api/gestuser/fonctionnalites/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les fonctionnalités.



**Profils :** [**http://127.0.0.1:8000/api/gestuser/profils/**](http://127.0.0.1:8000/api/gestuser/profils/)**:**

API qui permet d’effectuer les opération CRUD (Create, Read, Update, Delete ) sur les profils ou groupe d’utilisateur.



**Getuser :** [**http://127.0.0.1:8000/api/gestuser/getuser/**](http://127.0.0.1:8000/api/gestuser/getuser/)**:**

API qui permet de récupérer les données d’un utilisateur

**Getprivilege :** [**http://127.0.0.1:8000/api/gestuser/getprivilege/**](http://127.0.0.1:8000/api/gestuser/getprivilege/)**:**

API qui permet de récupérer les droits ou privilèges d’un utilisateur

**Getoneprofil :** [**http://127.0.0.1:8000/api/gestuser/getoneprofil/**](http://127.0.0.1:8000/api/gestuser/getoneprofil/)**:**

API qui permet de récupérer les données d’un profil

**Connecteduser :**[**http://127.0.0.1:8000/api/gestuser/connecteduser/**](http://127.0.0.1:8000/api/gestuser/connecteduser/)**:**

API qui permet de mettre a jour l’état de connexion d’un utilisateur

**updateuser:** [**http://127.0.0.1:8000/api/gestuser/updateuser/**](http://127.0.0.1:8000/api/gestuser/updateuser/)

API qui permet de mettre a jour les données d’un utilisateur

**Récupération Token :** [**http://127.0.0.1:8000/api/token/**](http://127.0.0.1:8000/api/token/)

API qui permet de récupérer un token

**Récupération Refresh Token : http://127.0.0.1:8000/api/token/refresh/**

API qui permet de récupérer un refresh token