Table des matières

[**Settings** 3](#_Toc23862672)

[**How to create First API Request** 4](#_Toc23862673)

[**Collections :** 5](#_Toc23862674)

[**Exécuter une collection** 6](#_Toc23862675)

[**Les variables** 8](#_Toc23862676)

[**Variable niveau collection** 8](#_Toc23862677)

[**Variable niveau Globale** 9](#_Toc23862678)

[**Afficher la console de postman** 10](#_Toc23862679)

[**Utilisation des variables à partir des scripts en javascript** 11](#_Toc23862680)

[**Les environnements** 12](#_Toc23862681)

[**Création des scripts rapidement avec Postman :** 16](#_Toc23862682)

[**Comment créer le premier test dans Postman** 17](#_Toc23862683)

[**Workspace** 18](#_Toc23862684)

[**Data file , comment récupération des données à partir d’un fichier CSV, JSON et faire les tests** 20](#_Toc23862685)

[**Comment récupérer les données à partir d’un fichier csv** 23](#_Toc23862686)

[**Comment récupérer les données à partir d’un fichier Json** 24](#_Toc23862687)

[**Créer des tests** 25](#_Toc23862688)

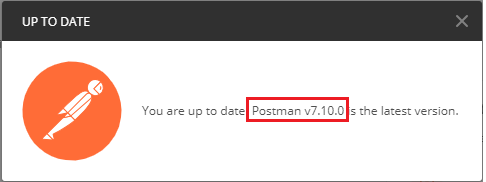
[**Exécuter Postman à partir de l’invite de commande (Newmane)** 25](#_Toc23862689)

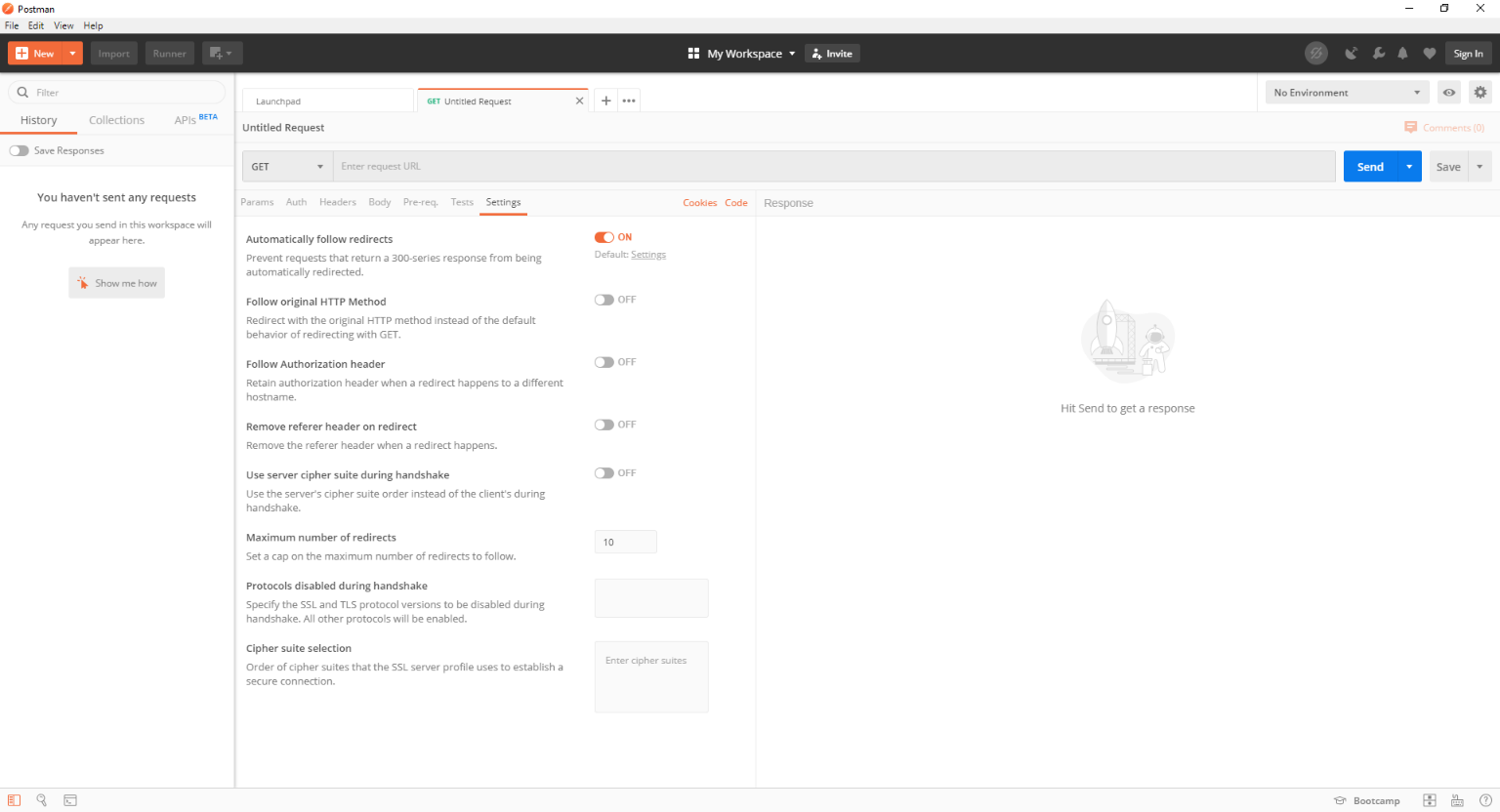
[**Exécuter une collection à distance par une Url** 26](#_Toc23862690)

[**Exécuter avec Jenkins** 28](#_Toc23862691)

**Postman**

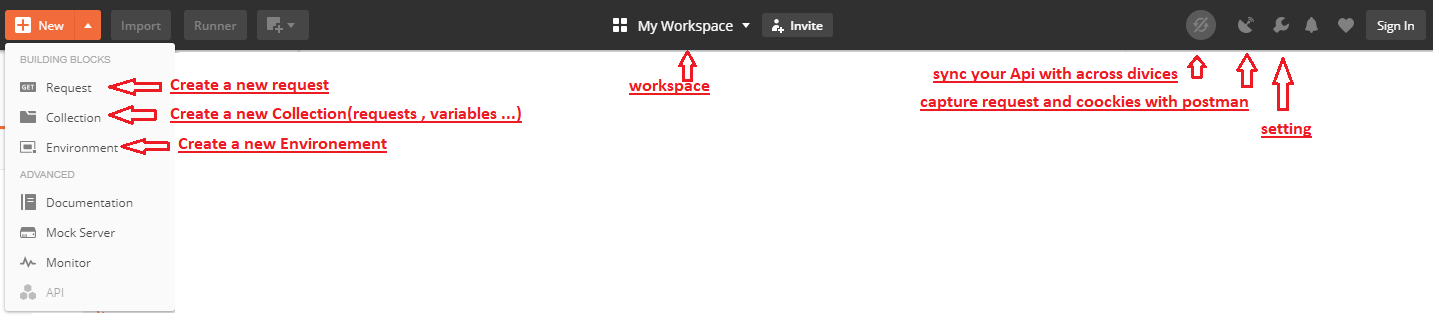
Dans cette documentation on va utiliser Postman version 7.0.0



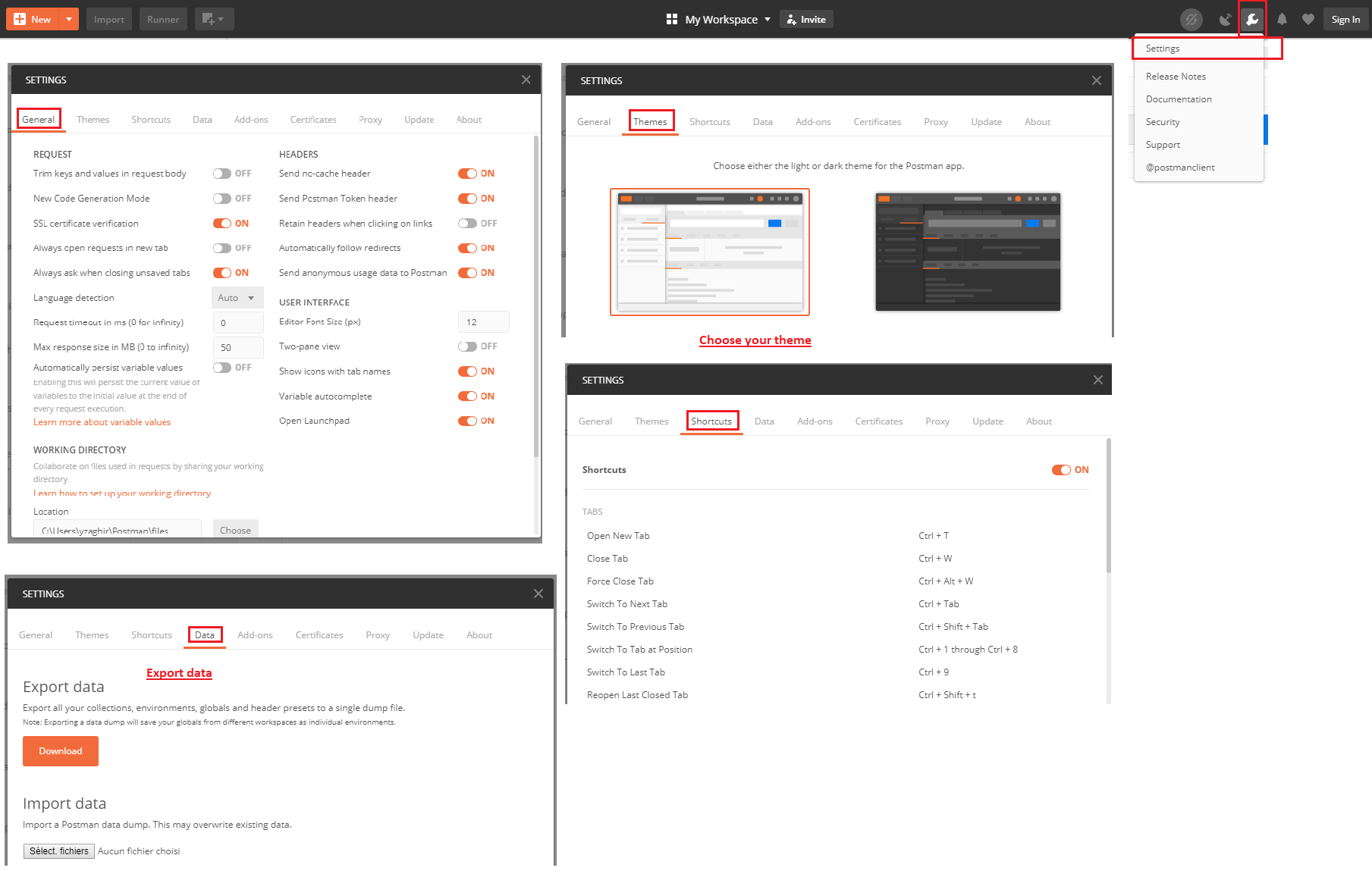
Interface UI postman

Menu

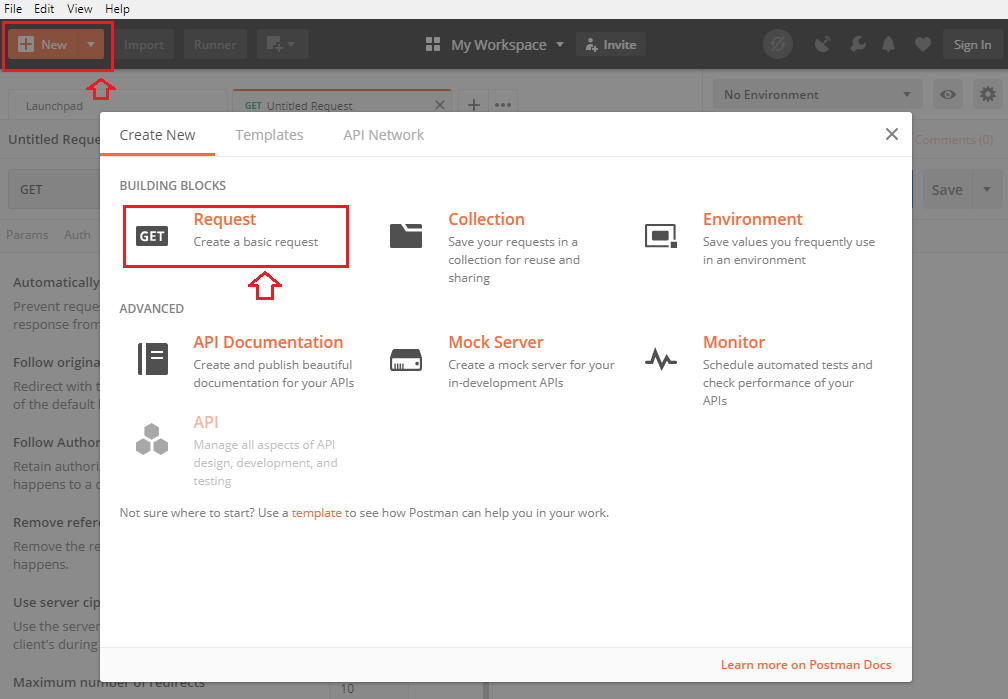


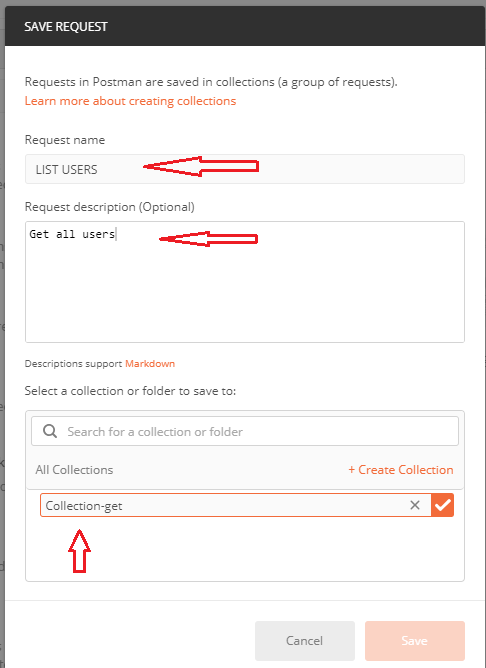


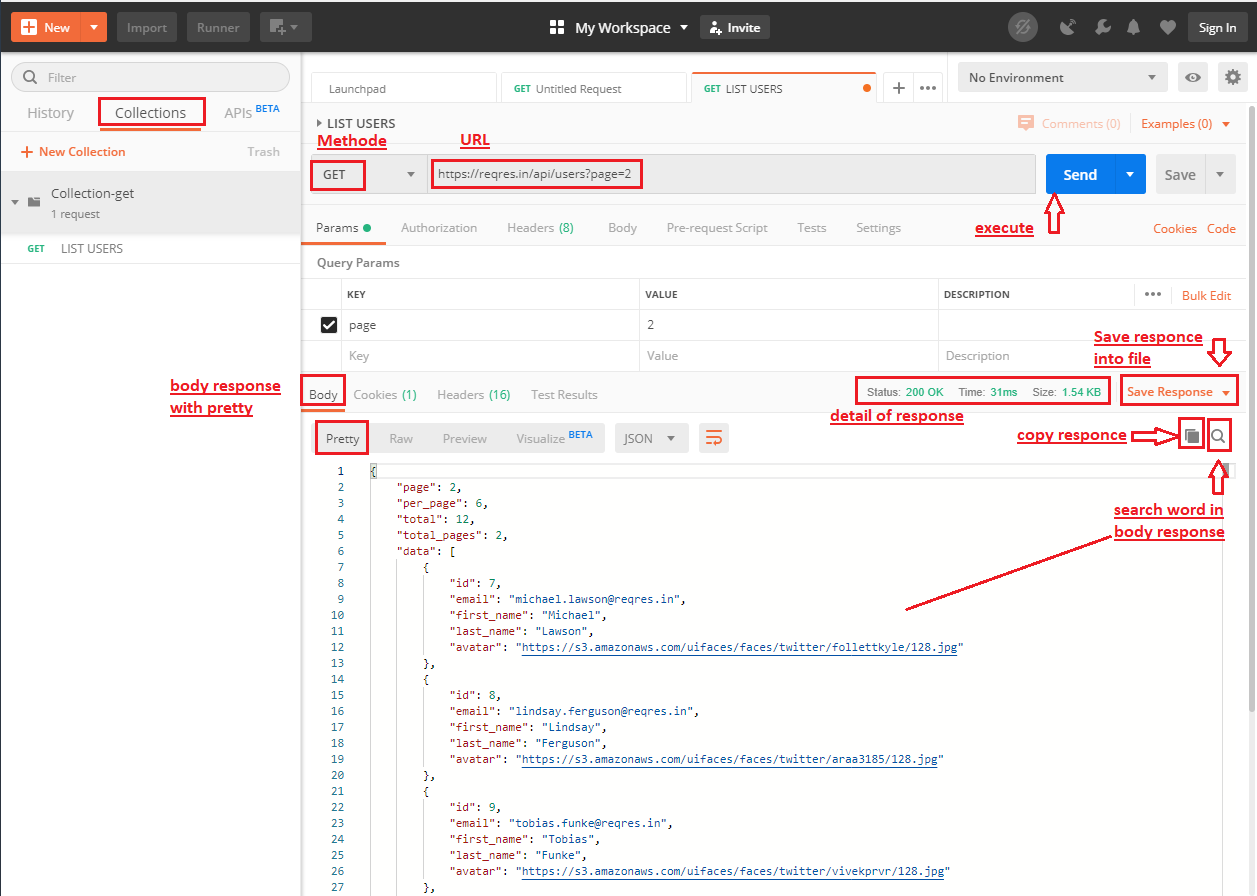
**Settings**

****

**How to create First API Request**



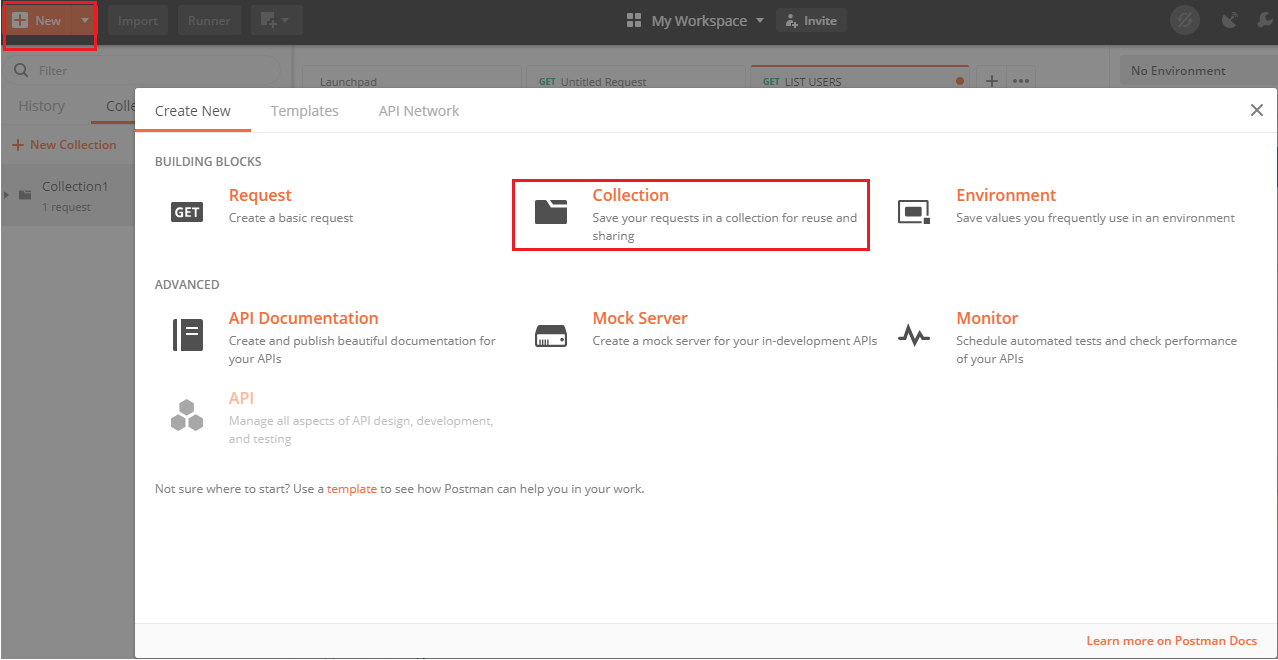


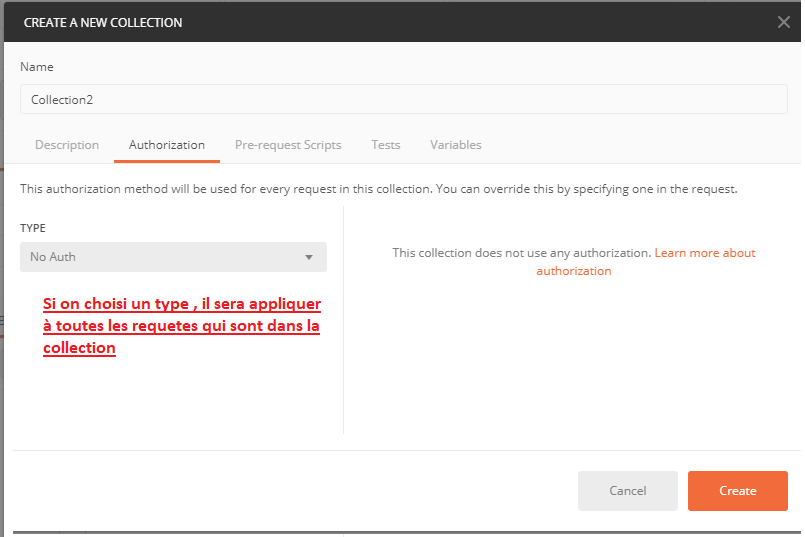


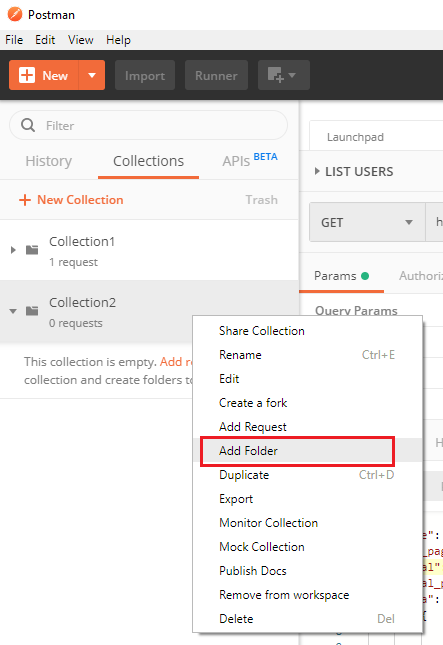
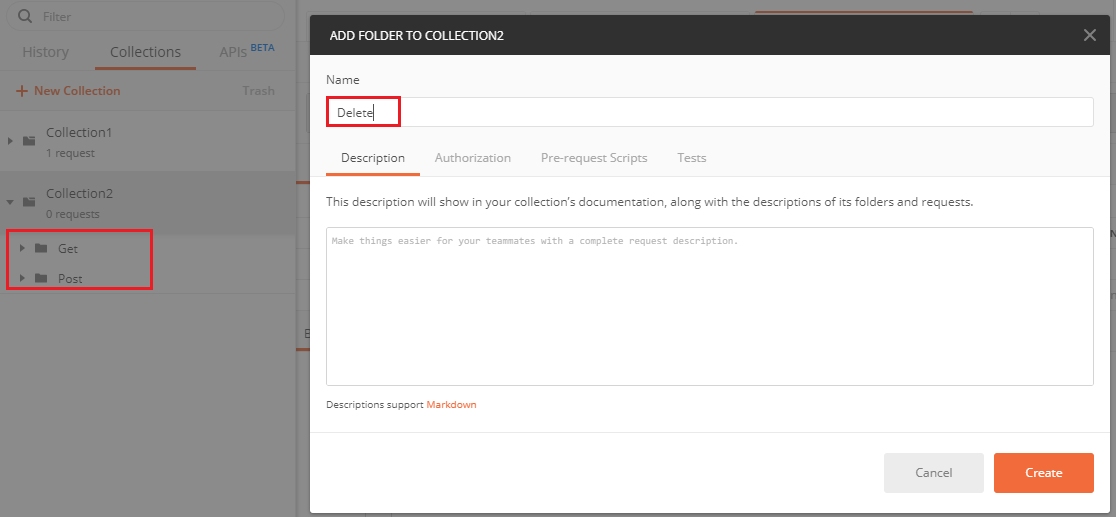
**Collections :**

Collection c’est un groupe de requêtes d’apis , qui peuvent être stocker et enregistrer dans un arrangement logique

Créer une nouvelle collection





Ajouter un dossier pour mieux structurer les types de requêtes

**Exécuter une collection**

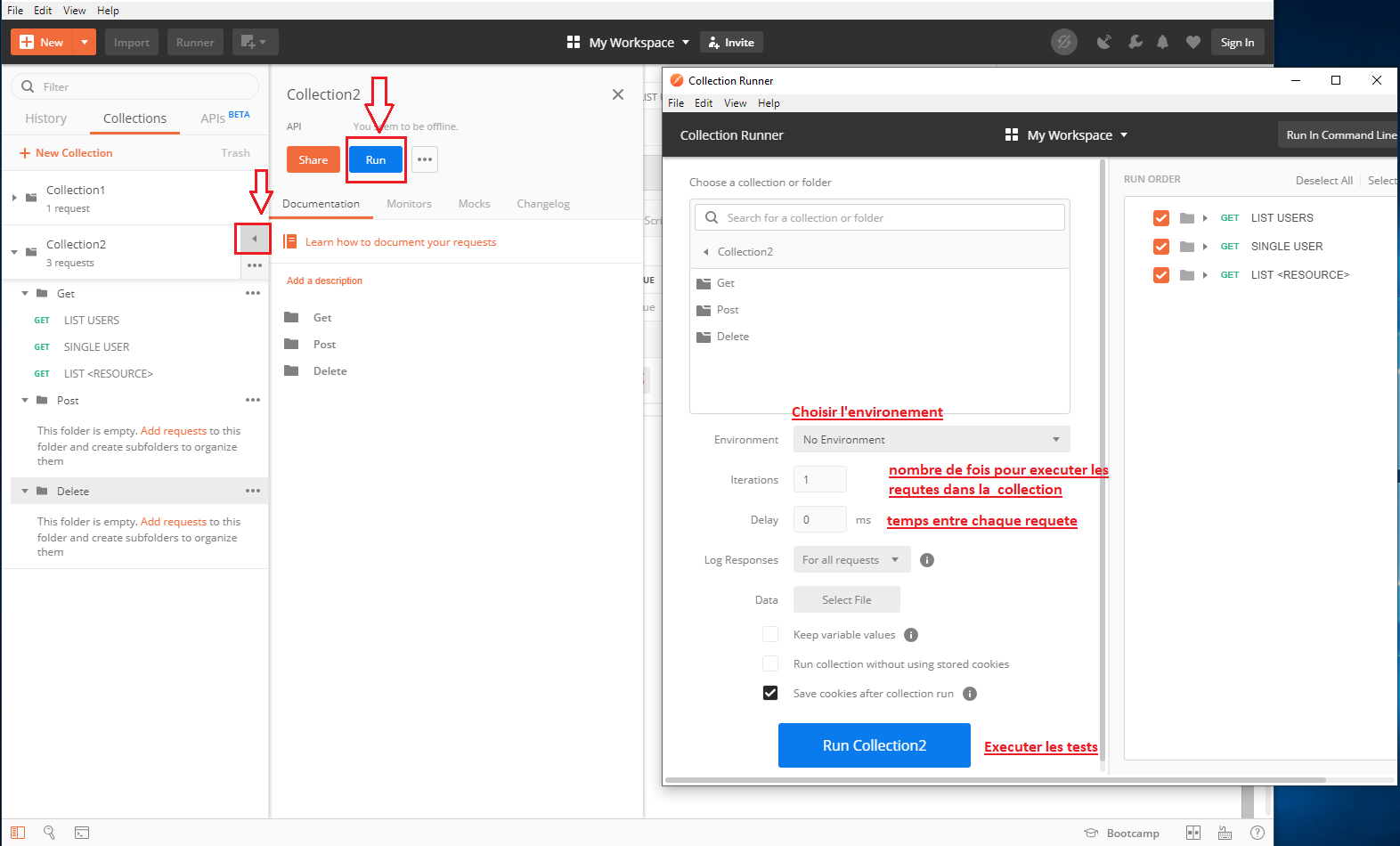
Le site <https://reqres.in/> propose plusieurs ressources

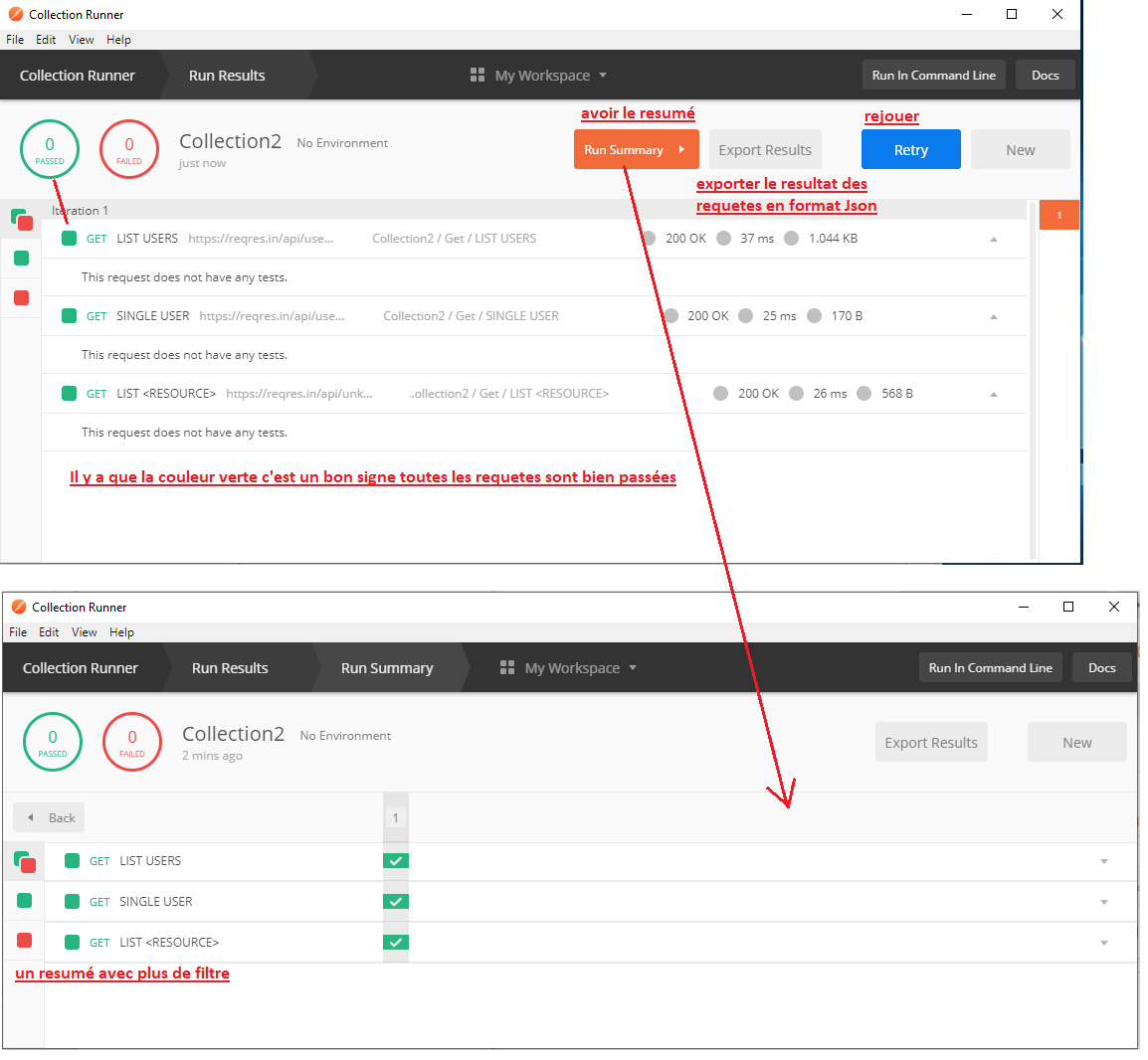
On va ajouter quelques uri dans une collection

LIST USERS : <https://reqres.in/api/users?page=2>

SINGLE USER : <https://reqres.in/api/users/2>

LIST <RESOURCE> : <https://reqres.in/api/unknown>





**Les variables**

C’est quoi les variables ?

Les variables sont des éléments (data store) qui peuvent prendre plusieurs valeurs

Pourquoi ?

Pour la réutilisation des valeurs dans plusieurs endroit

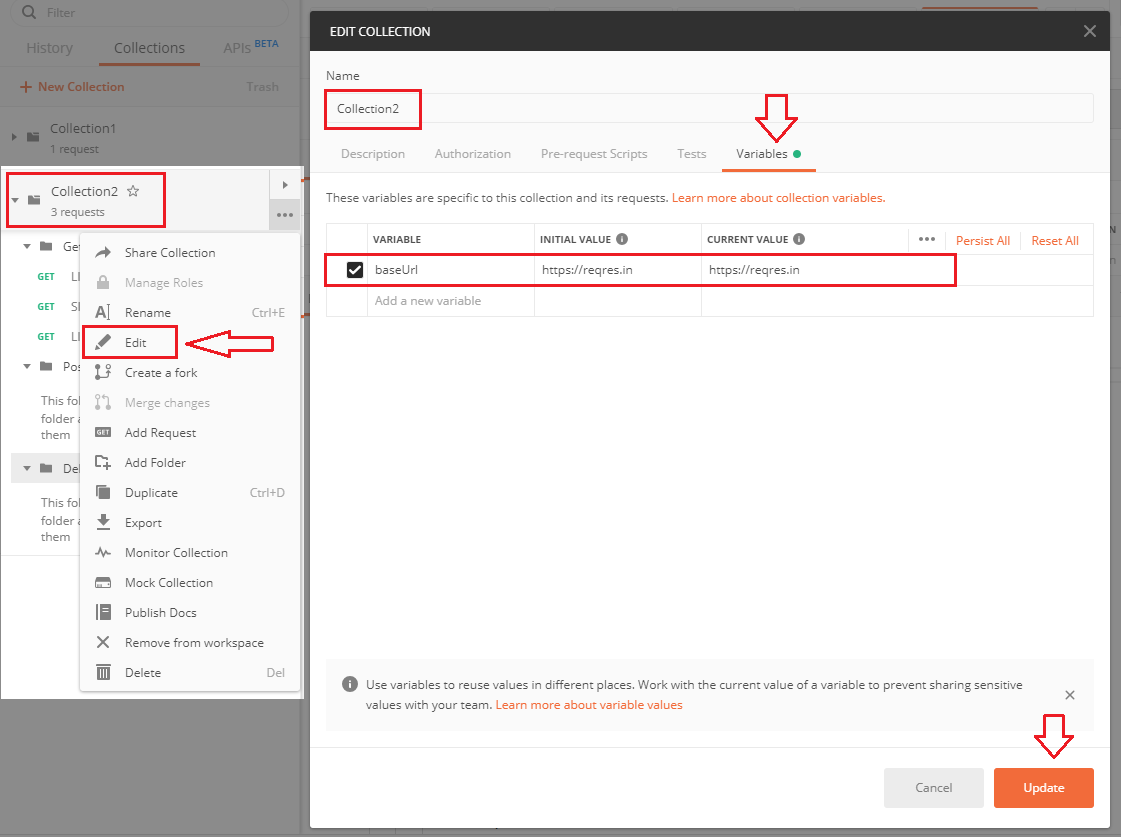
Interdire la répétition

Interdire de refaire la configuration des requêtes quand la valeur change

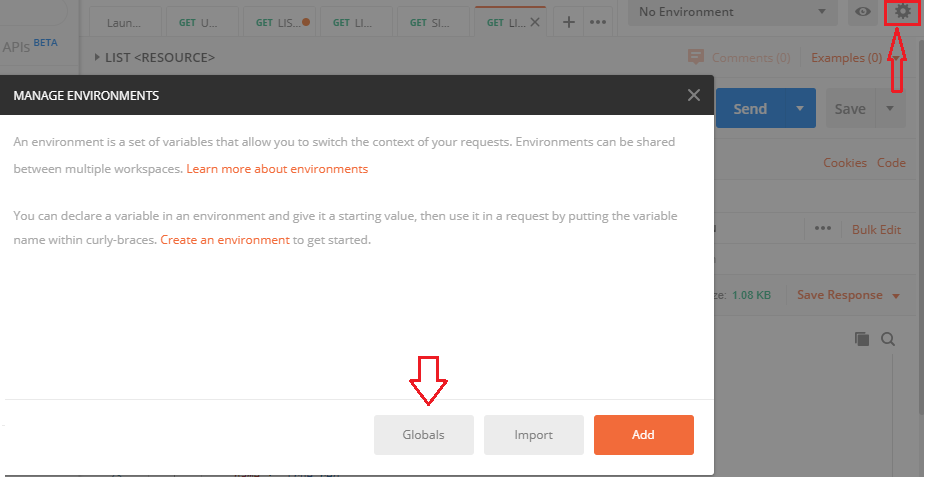
Il y a 3 scopes pour une variable. Elle peut être lié à la collection, à l’environnement, ou elle est reconnue d’une manière globale.

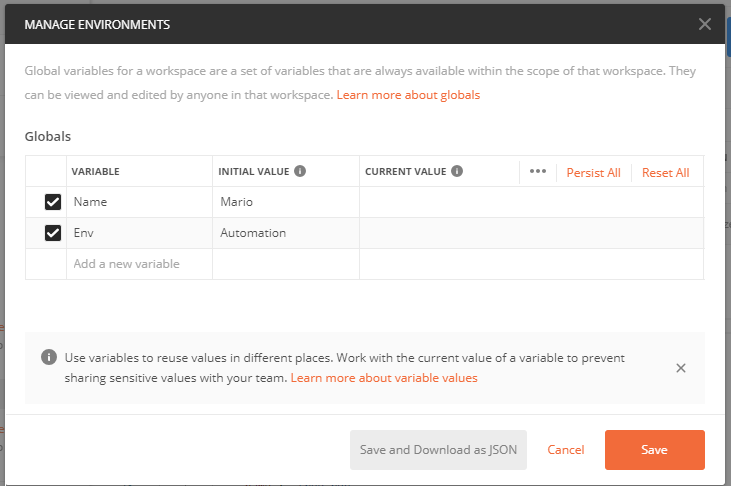
Dans notre exemple on va modifier nos uri pour que la partie static (url du site = <https://reqres.in>) soit variabiliser. Donc le format de notre request soit sous la forme **{{baseUrl}}/path**

**Variable niveau collection**

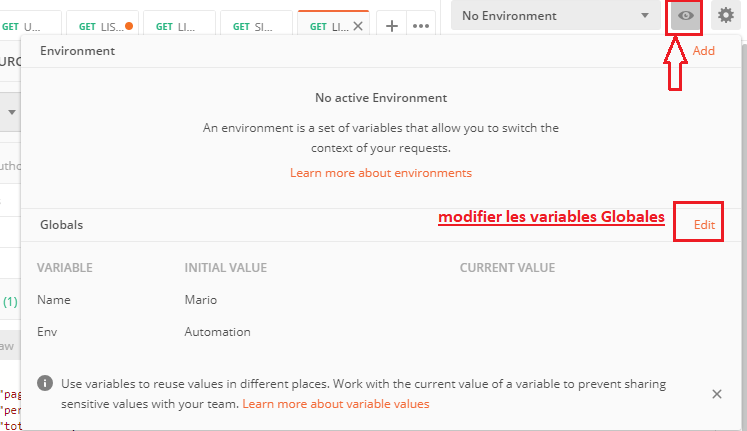


**Variable niveau Globale**

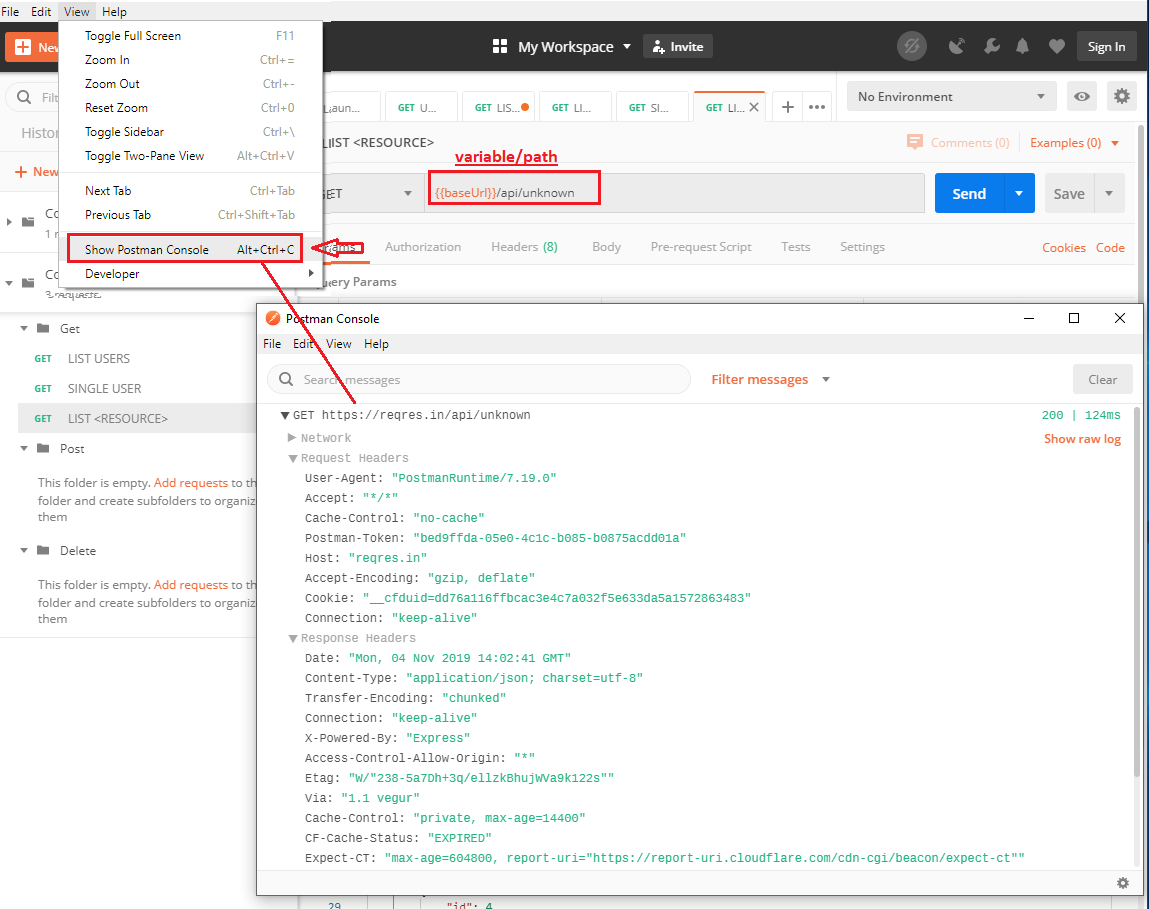




Pour voir les variables Globales



**Afficher la console de postman**



**Utilisation des variables à partir des scripts en javascript**

Dans onglet tests on peut passer des scripts en javascript

console.log("test Single User");

let baseUrl = pm.variables.get("baseUrl");

console.log("Value for url variable is :=> ", baseUrl);

// modifier la valeur d'une variable

pm.variables.set("baseUrl", "Postman") ;

let envVariable = pm.globals.get("Env");

console.log("Env avant :=> " , envVariable);

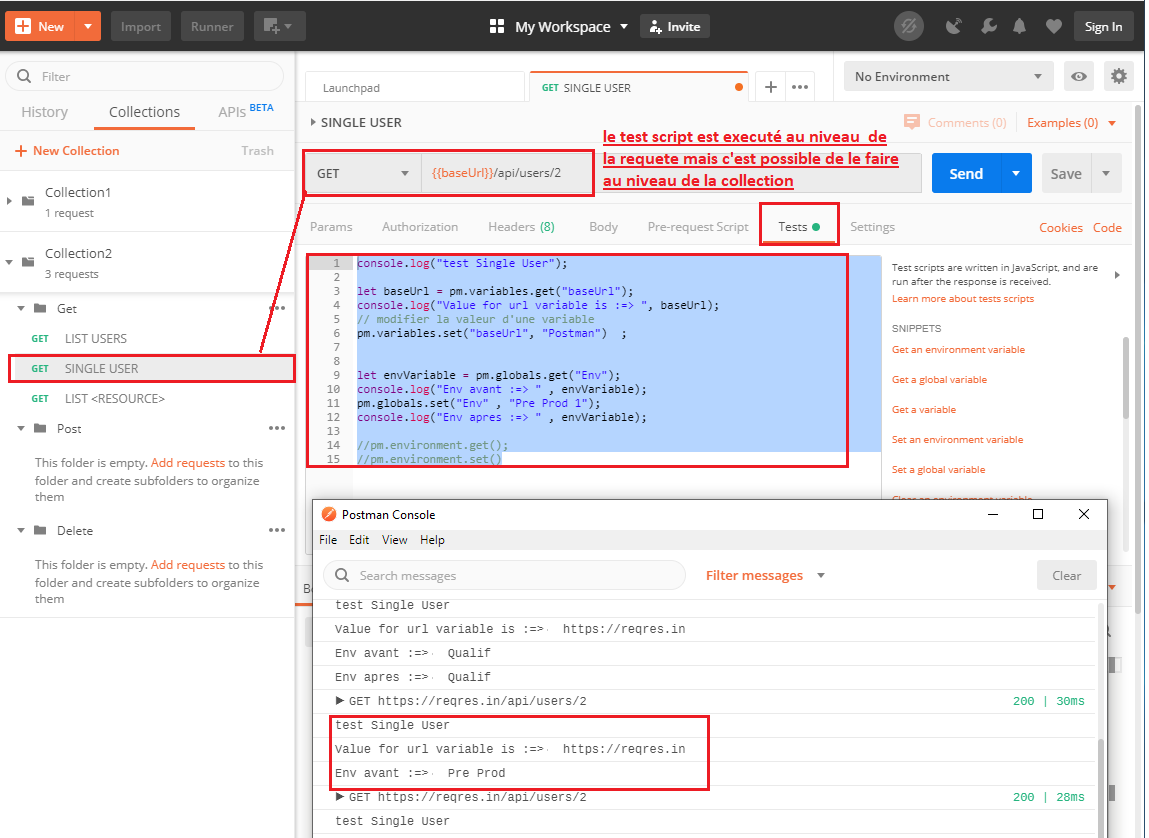
pm.globals.set("Env" , "Pre Prod 1");

console.log("Env apres :=> " , envVariable);

//pm.environment.get();

//pm.environment.set()

Il faut activer la console postman pour voir le resultat



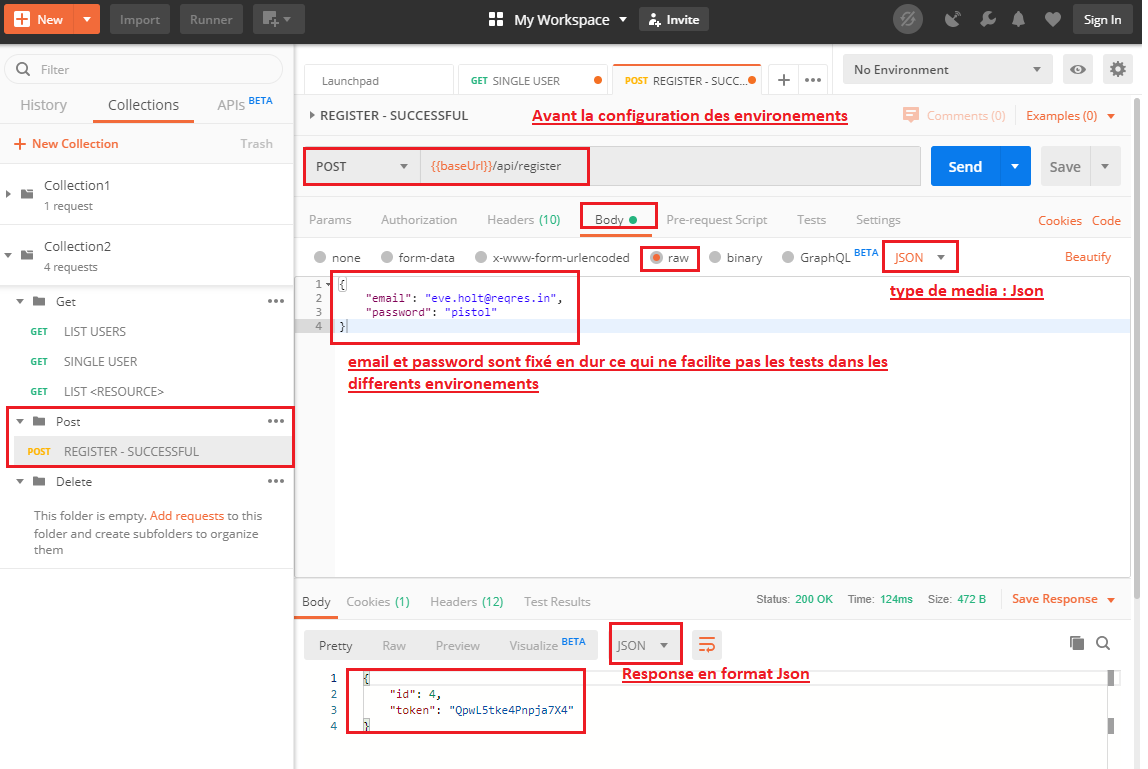
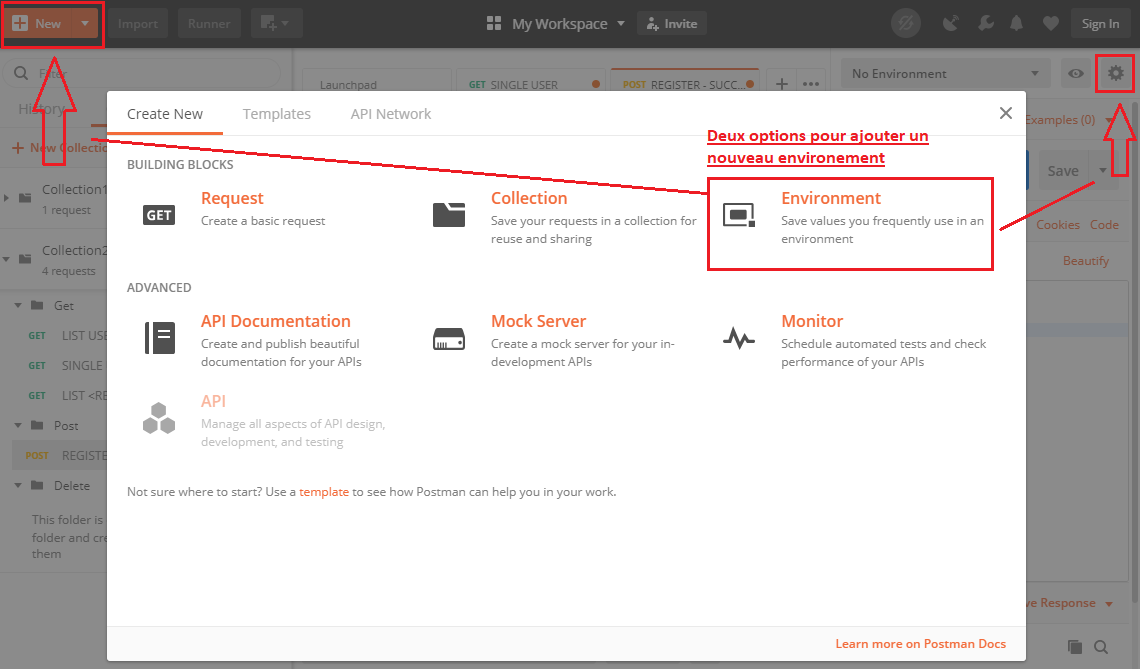
**Les environnements**

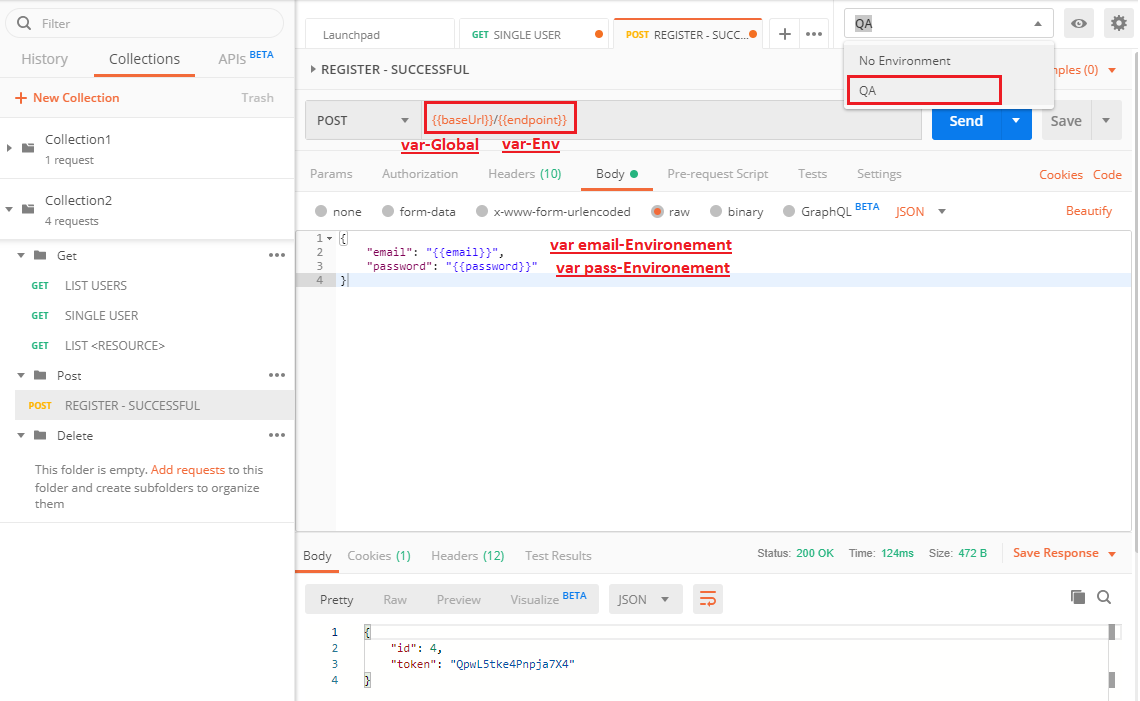
C’est quoi ?

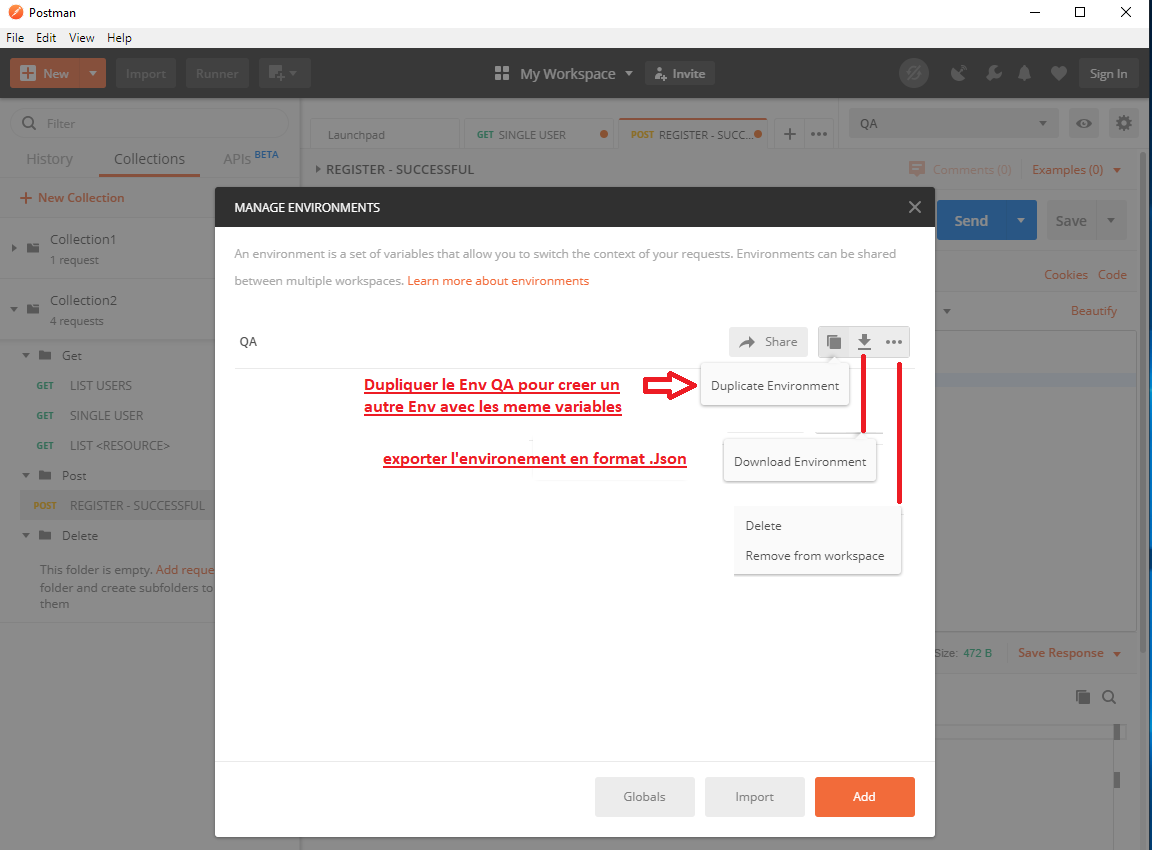
Un environnement c’est un ensemble des paires key value

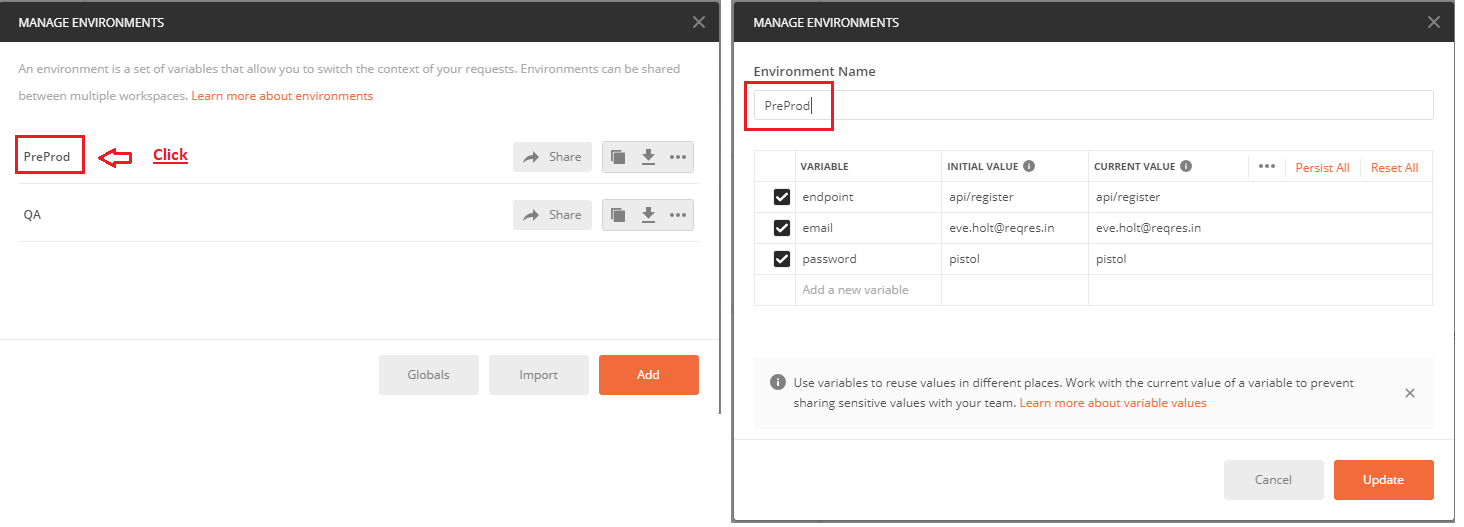
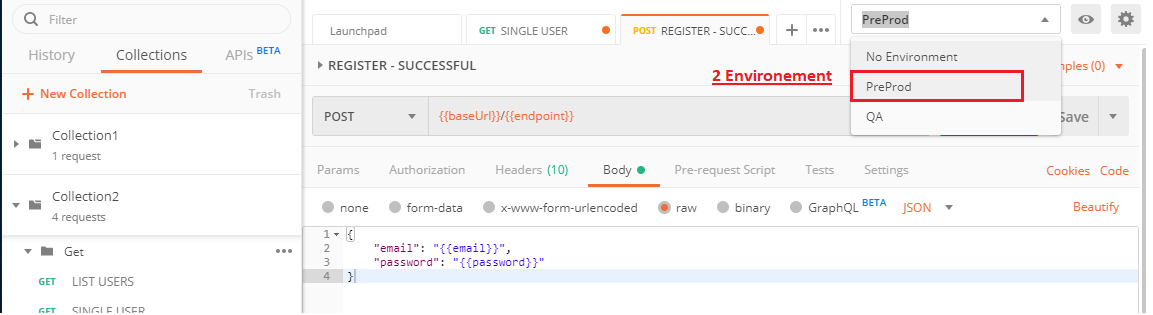
On va ajouter une nouvelle requête qui demande un user et password comme paramètre

Ces deux paramètre changent d’un environnement à un autre, dans ce cas on va utiliser les environnements configurer dans Postman

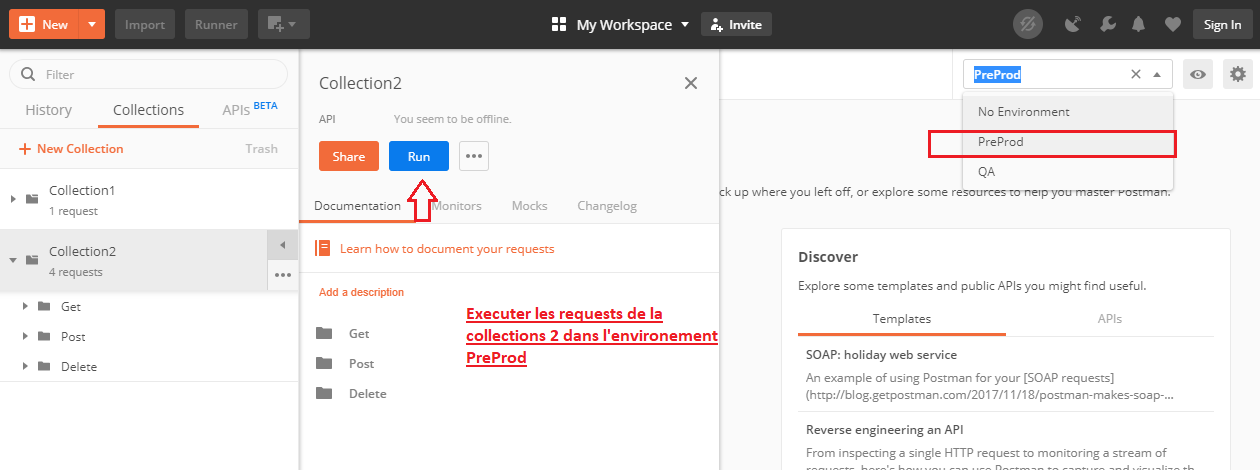
Créer un nouvel environnement :

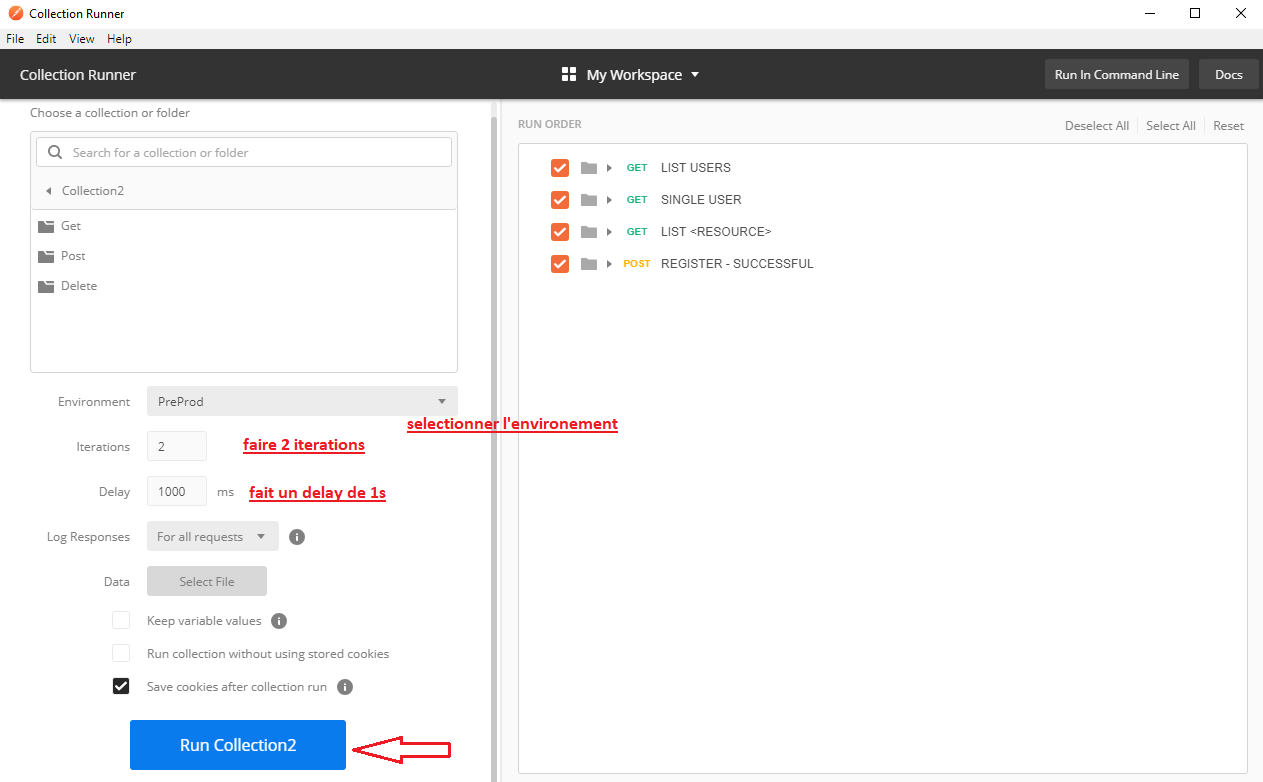


**Créer un autre environnement PreProd**

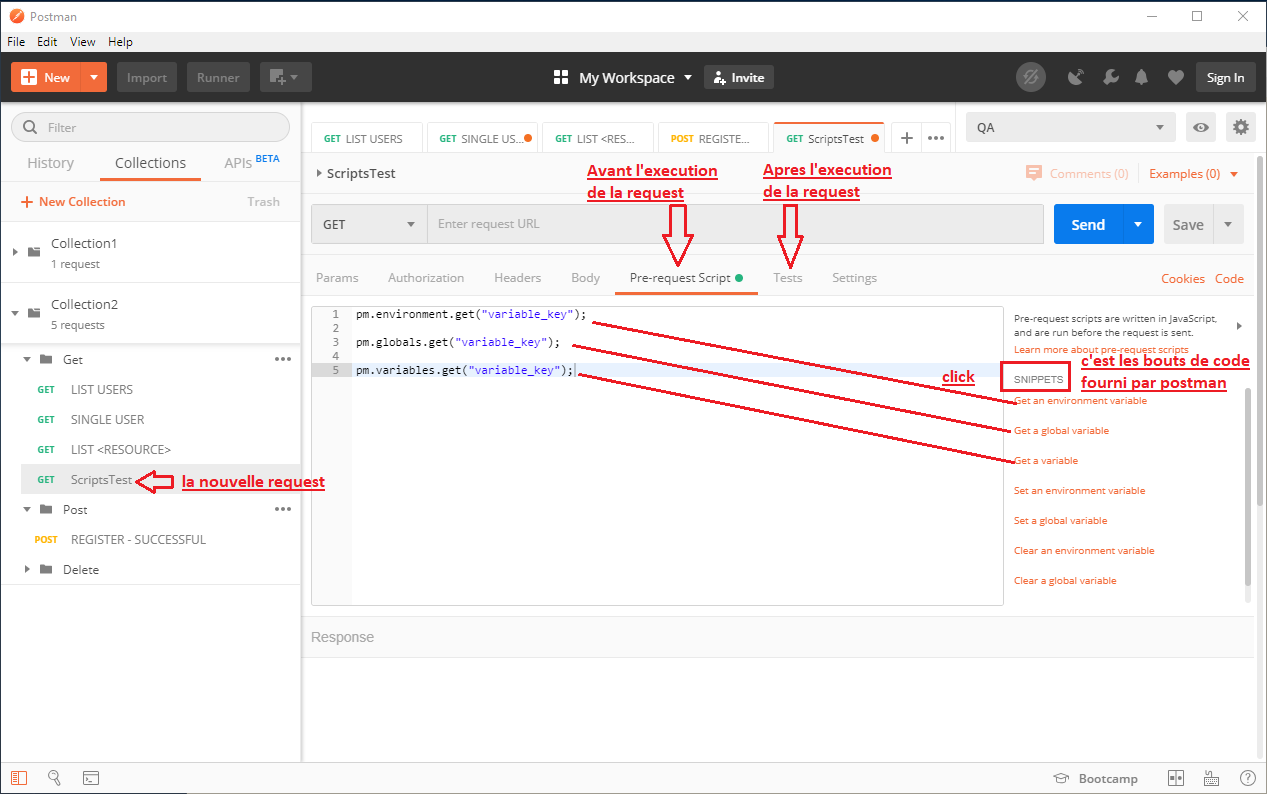


**Exécuter les requests dans un environnement PreProd**





**Création des scripts rapidement avec Postman :**

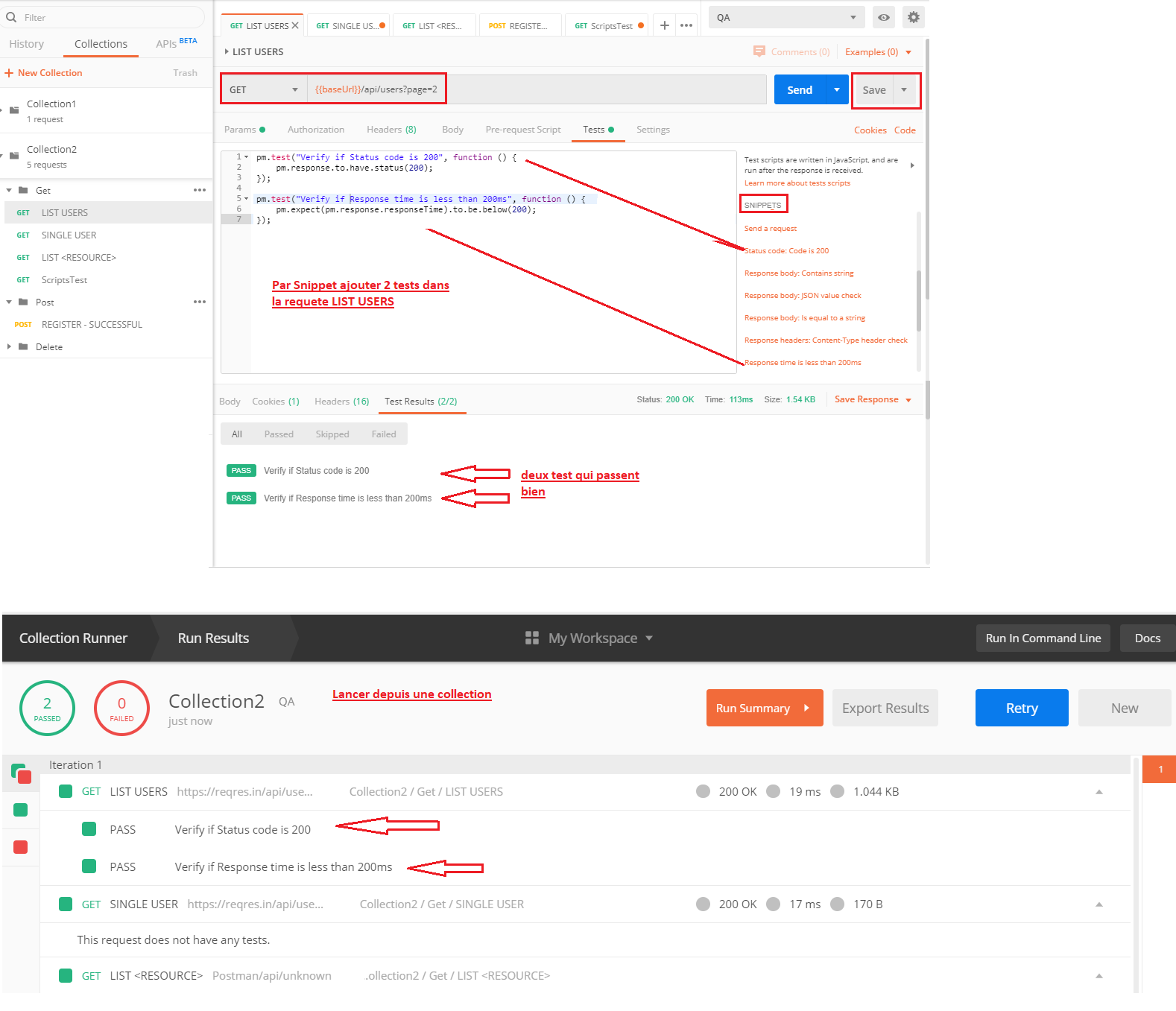
Postman génère des bouts de code ou des Templates pour nos tests

**Comment créer le premier test dans Postman**

C’est quoi un test dans Postman ?

Les tests dans Postman sont des codes javascript exécutés après la réception de la réponse

Les tests peuvent être exécuter au niveau d’une collection, d’un dossier, d’une requête

Ils peuvent être jouer avant la request (Pre-request Script) ou après (Tests)

Code Javascript

pm.test("Verify if Status code is 200", function () {

pm.response.to.have.status(200);

});

pm.test("Verify if Response time is less than 200ms", function () {

pm.expect(pm.response.responseTime).to.be.below(200);

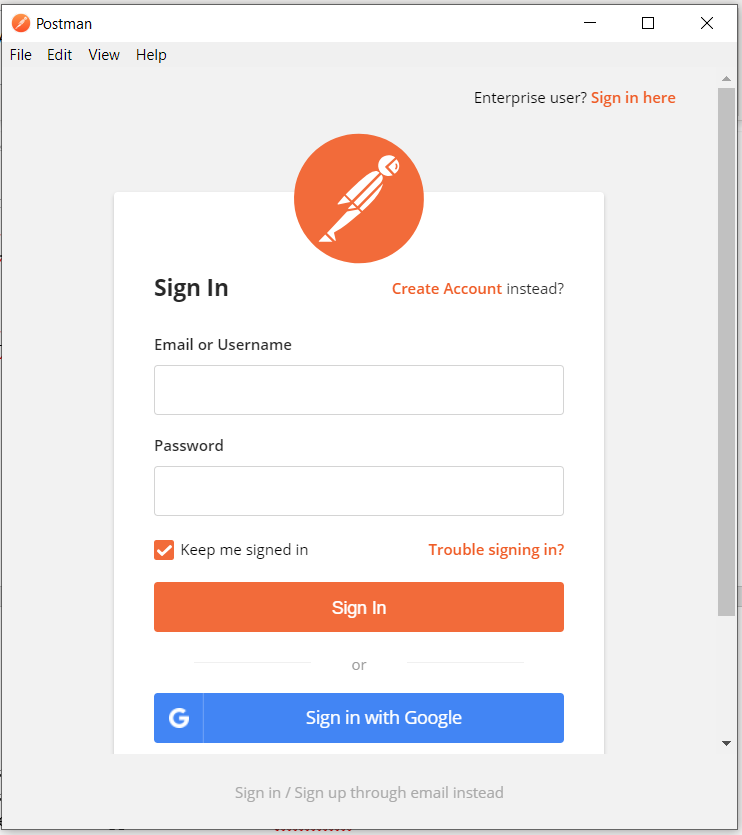
});

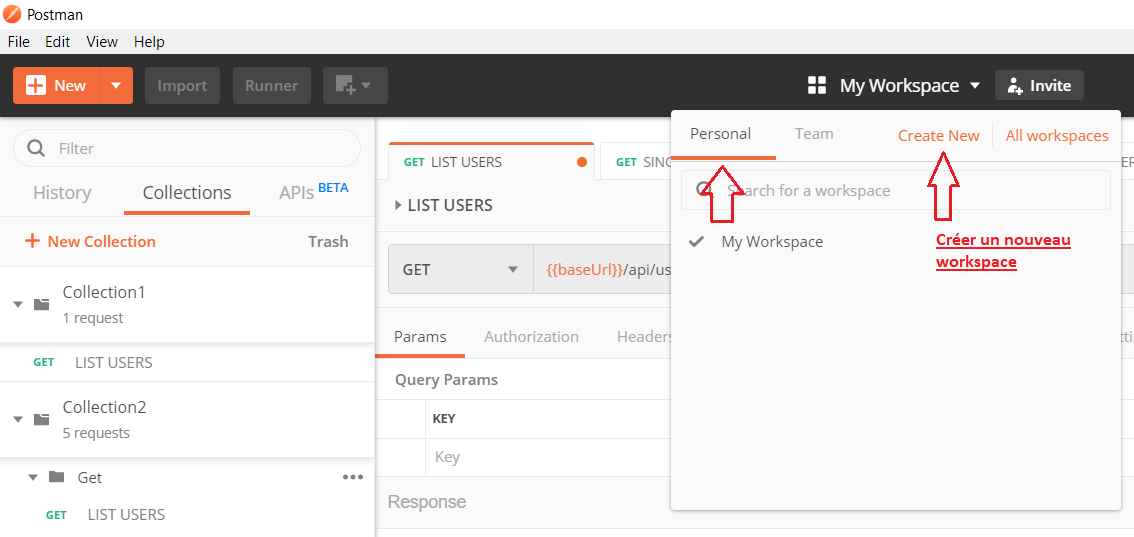
**Workspace**

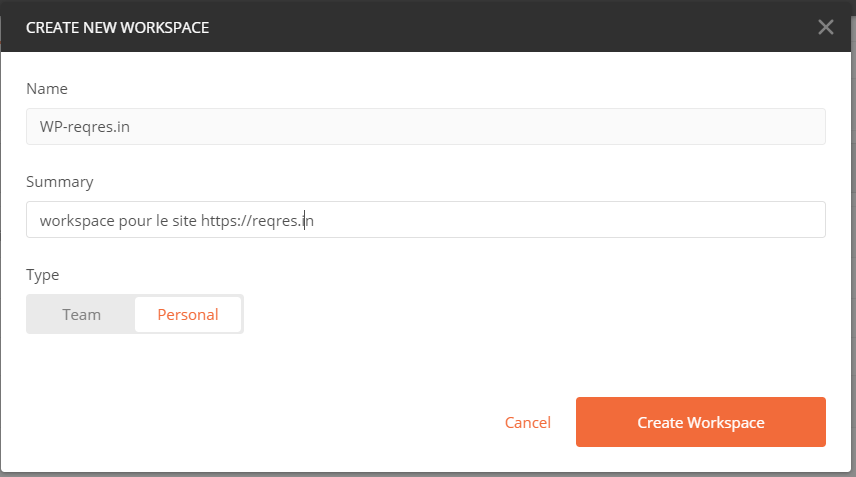
Workspace c’est quoi ?

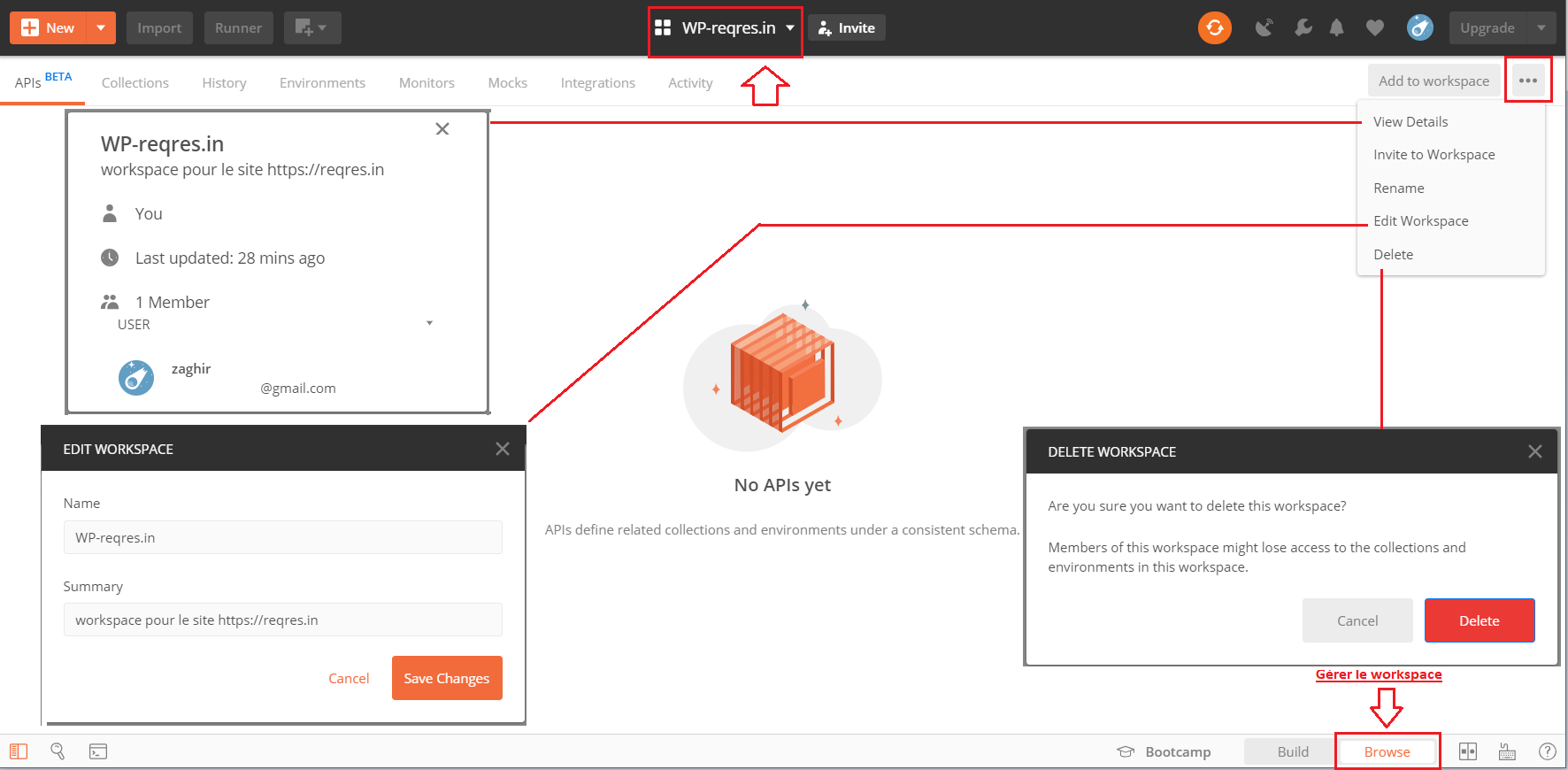
Workspace c’est une zone ou on peut grouper organiser et manager nos collections

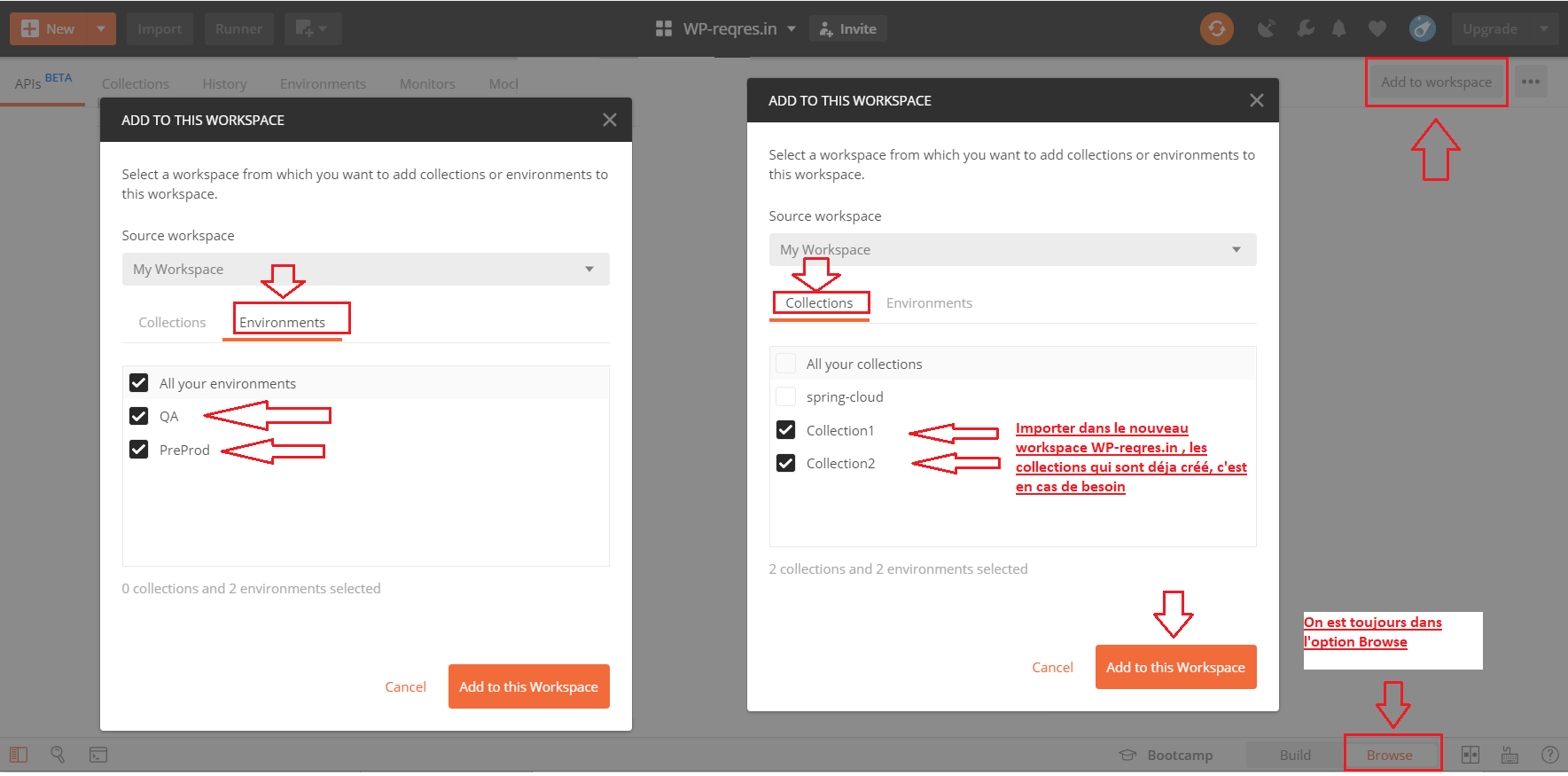
Pour commencer il faut ce logger avec un mail et password

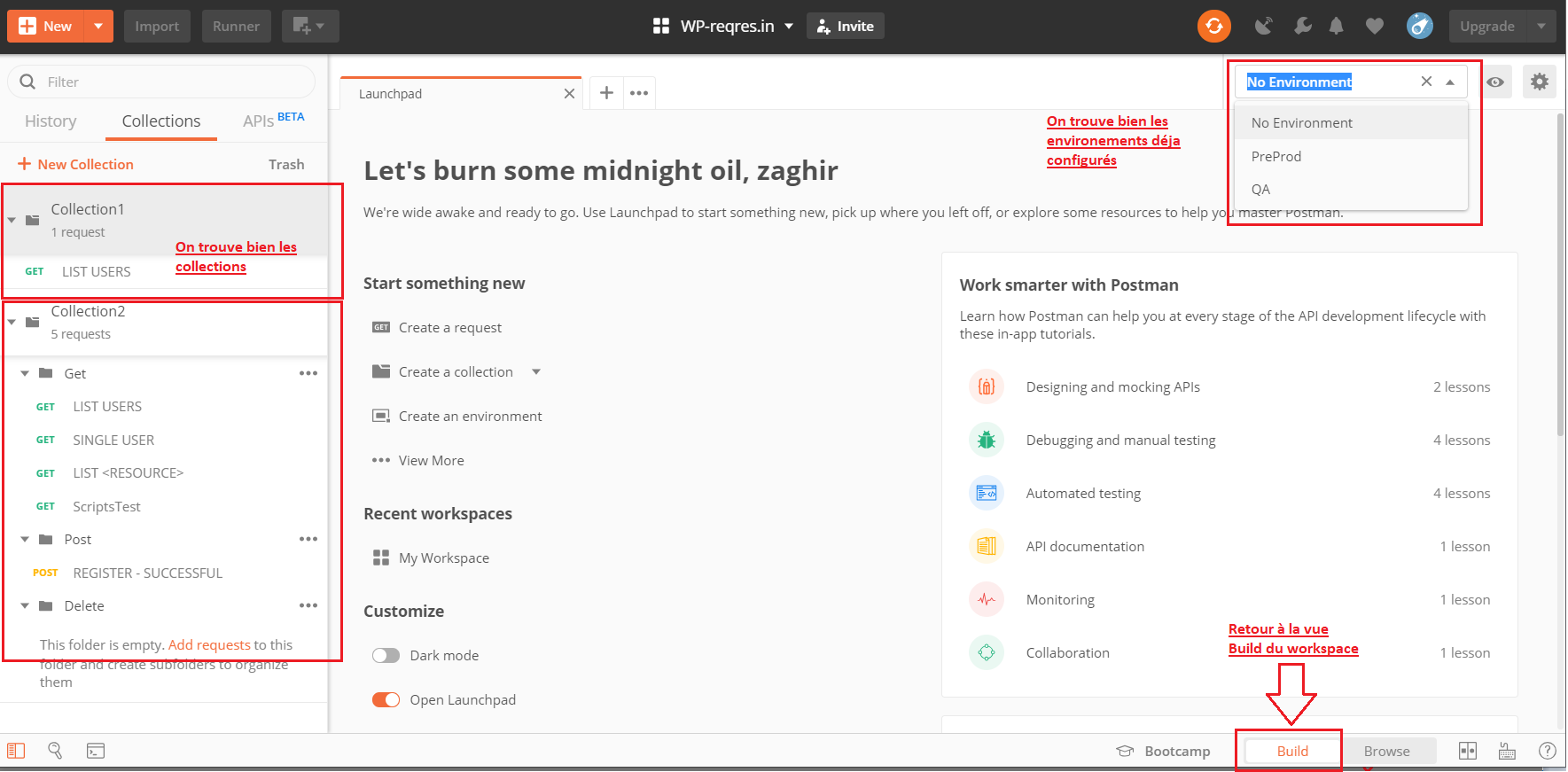












**Data file , comment récupération des données à partir d’un fichier CSV, JSON et faire les tests**

Avant de commencer on prépare une request de type post

Le site [https://reqres.in/](https://reqres.in/api/register) fourni une ressource de type post

URI = <https://reqres.in/api/register>

Descriptif =REGISTER SUCCESSFUL

Body param of Request =

{

"email": "eve.holt@reqres.in",

"password": "pistol"

}

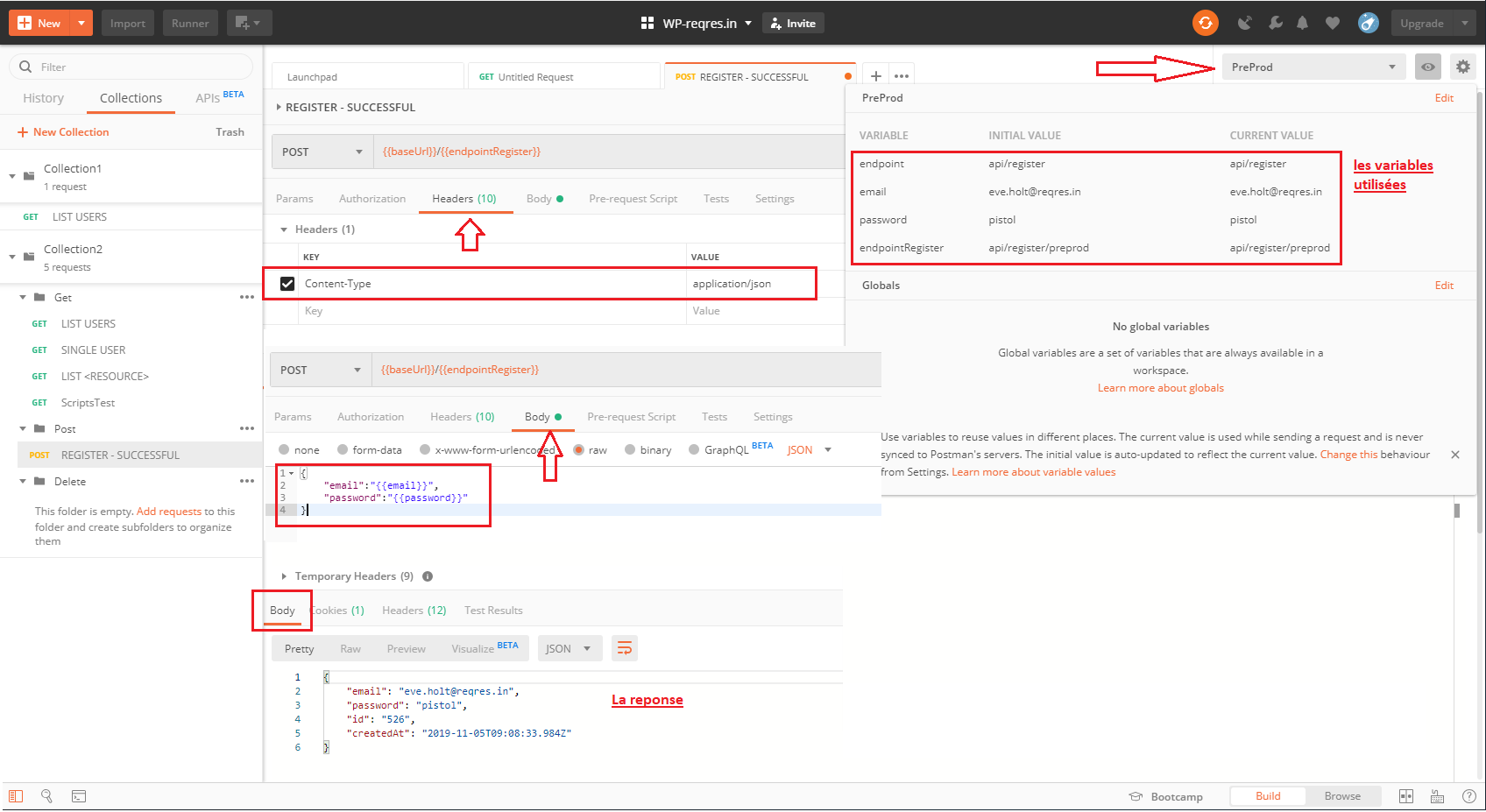
Response

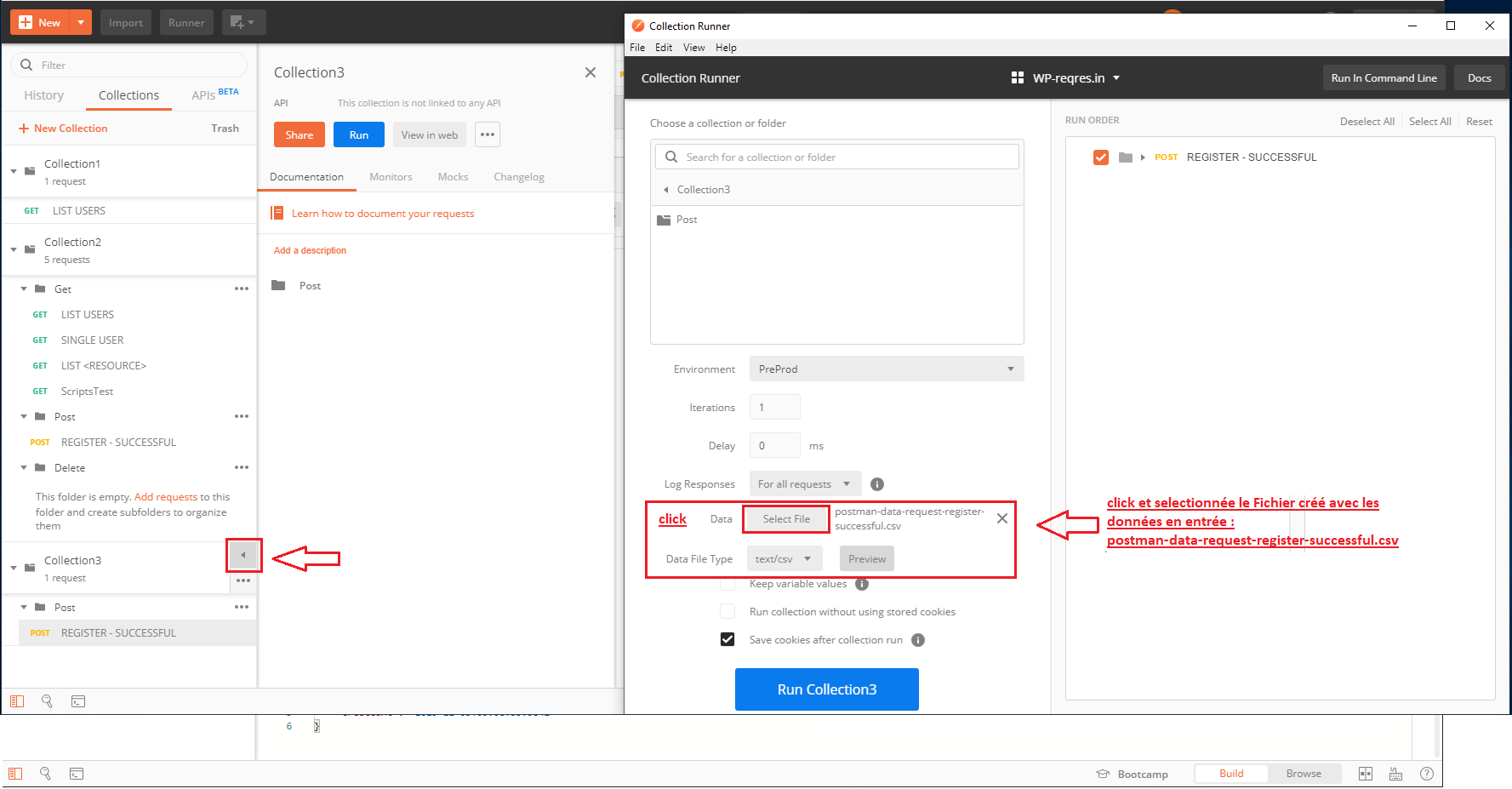
{

"id": 4,

"token": "QpwL5tke4Pnpja7X4"

}





**Comment récupérer les données à partir d’un fichier csv**

Créer un fichier .csv notre cas c’est

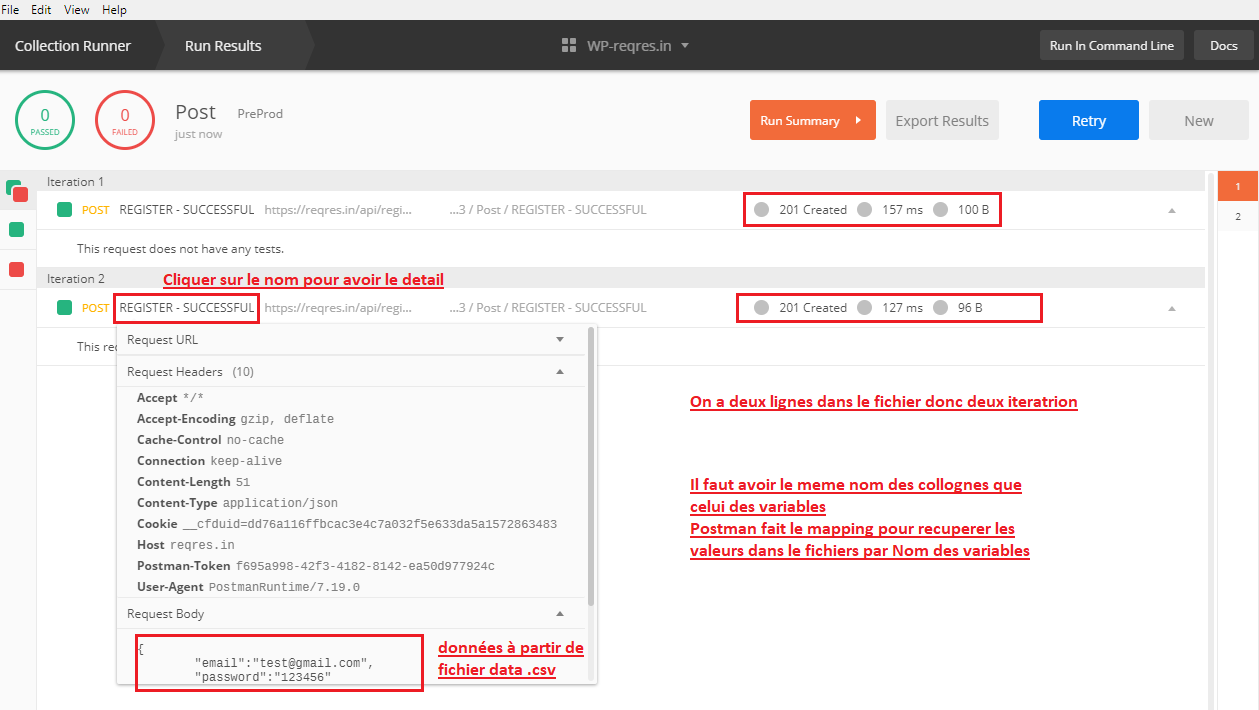
Nom de fichier : postman-data-request-register-successful.csv

Contenu du fichier :

endpointRegister ,email ,password

api/register/preprod ,eve.holt@reqres.in ,pistol

api/register/preprod ,test@gmail.com ,123456



**Comment récupérer les données à partir d’un fichier Json**

Préparer le fichier .json

Nom : postman-data-request-register-successful.json

Contenu :

[

    {

      "endpointRegister": "api/register/preprod",

      "email": "eve.holt@reqres.in",

      "password": "pistol"

    },

    {

      "endpointRegister": "api/register/preprod",

      "email": "test@gmail.com",

      "password": 123456

    }

  ]

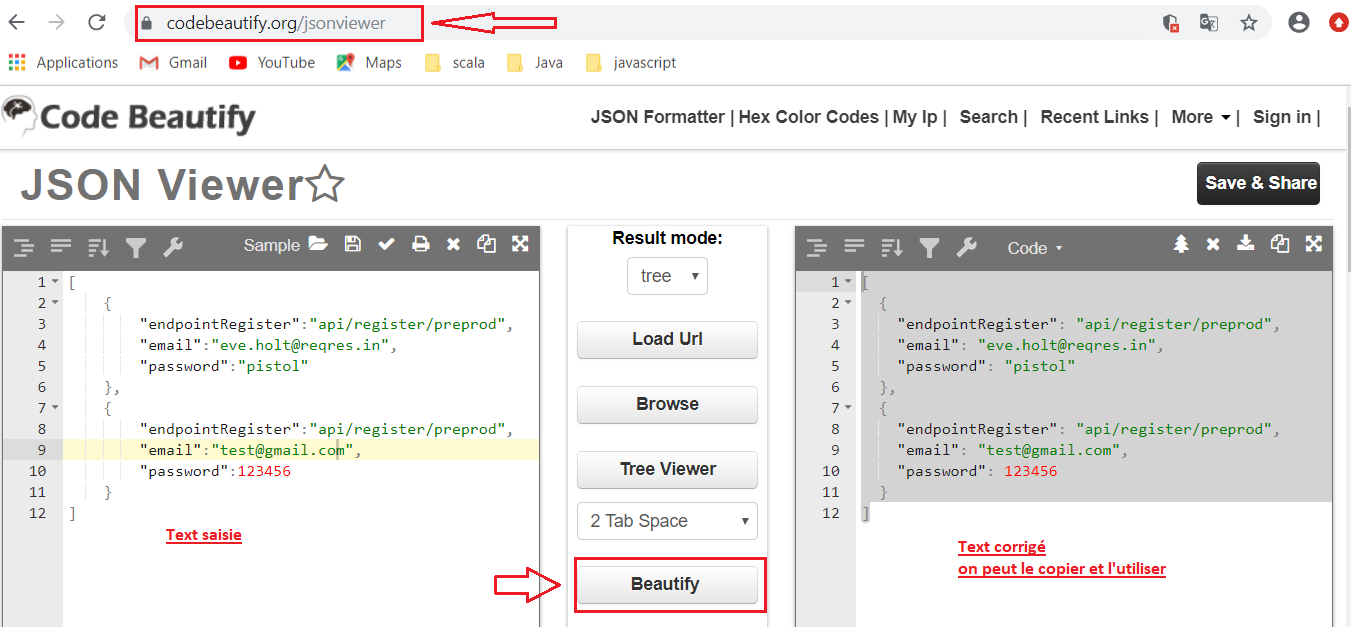
On définit un tableau d’objet par [ ]

Et un objet par {}

Dans notre fichier on a deux objets de test dans un tableau

Pour l’utilisation de fichier json c’est la même manipulation vue pour le fichier csv rien ne change

Dans le cas ou on veut vérifier la syntaxe saisie dans notre fichier json , on peut utilise un outil pour le vérifié exemple

Url : <https://codebeautify.org/jsonviewer>

**Créer des tests**

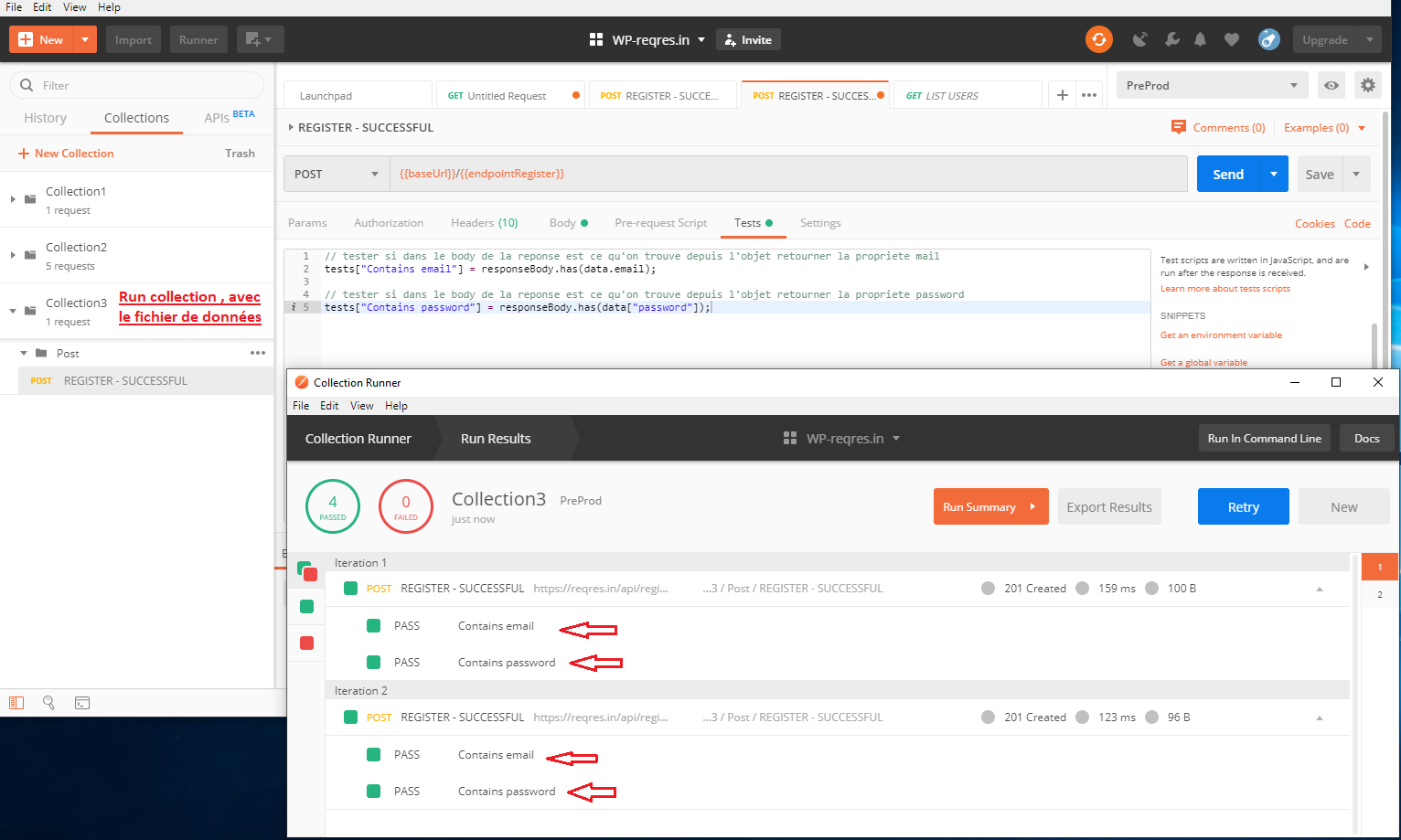
Les tests en javascript

  // tester si dans le body de la reponse est ce qu'on trouve depuis l'objet retourner la propriete mail

tests["Contains email"] = responseBody.has(data.email);

// tester si dans le body de la reponse est ce qu'on trouve depuis l'objet retourner la propriete password

tests["Contains password"] = responseBody.has(data["password"]);



**Exécuter Postman à partir de l’invite de commande (Newmane)**

Comment executer Postman les requests/tests à partir de l’invite de commande ?

1. Installer nodeJS

<https://nodejs.org/en/>

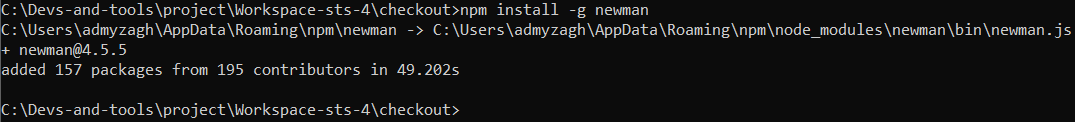
Vérifier l’installation de nodeJs avec : node -v

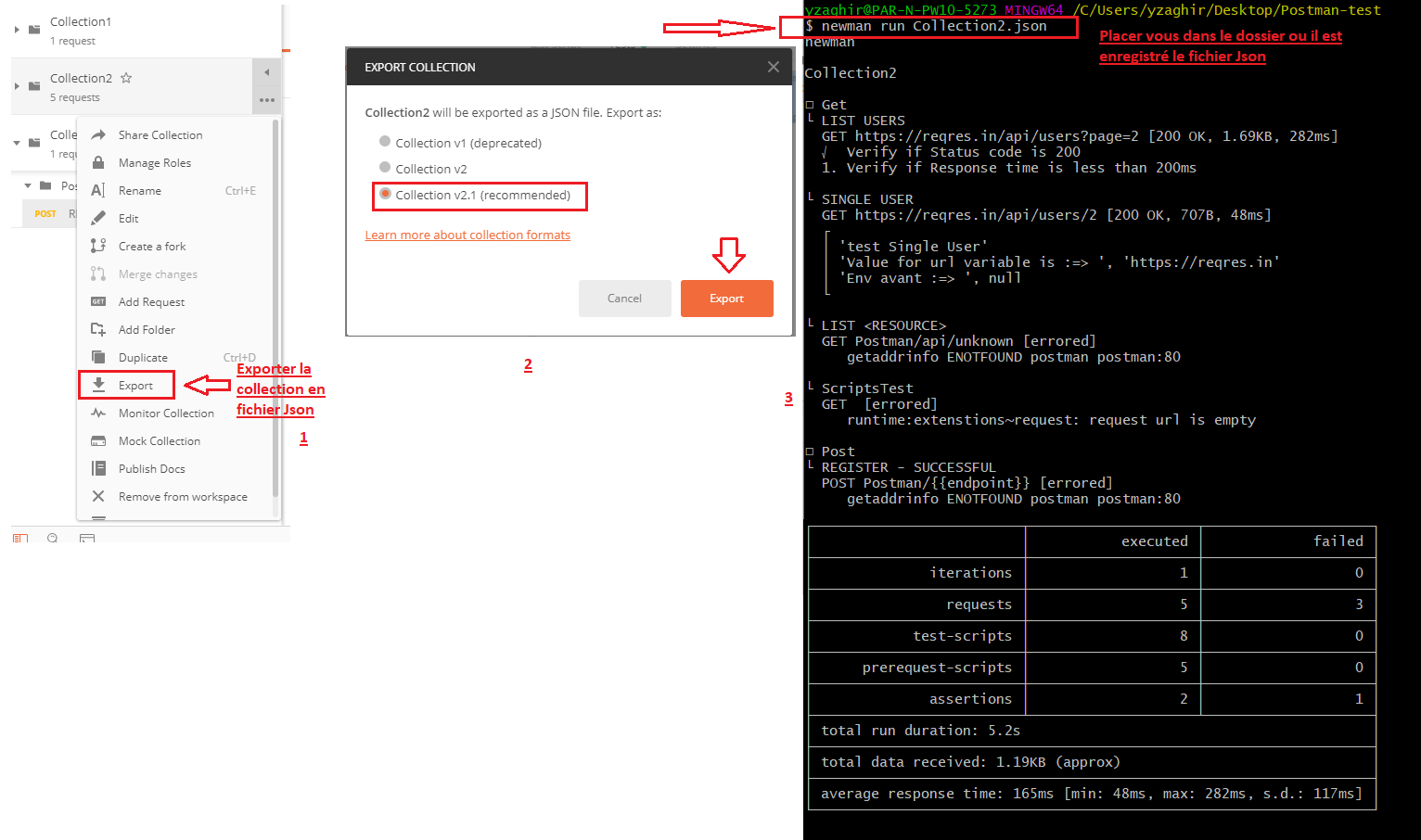
Vérifier l’installation de npm avec : npm -v

1. Installer Newman

Newman est un exécuteur des ligne des commandes des collections pour Postman

Installer Newman avec : npm install -g newman



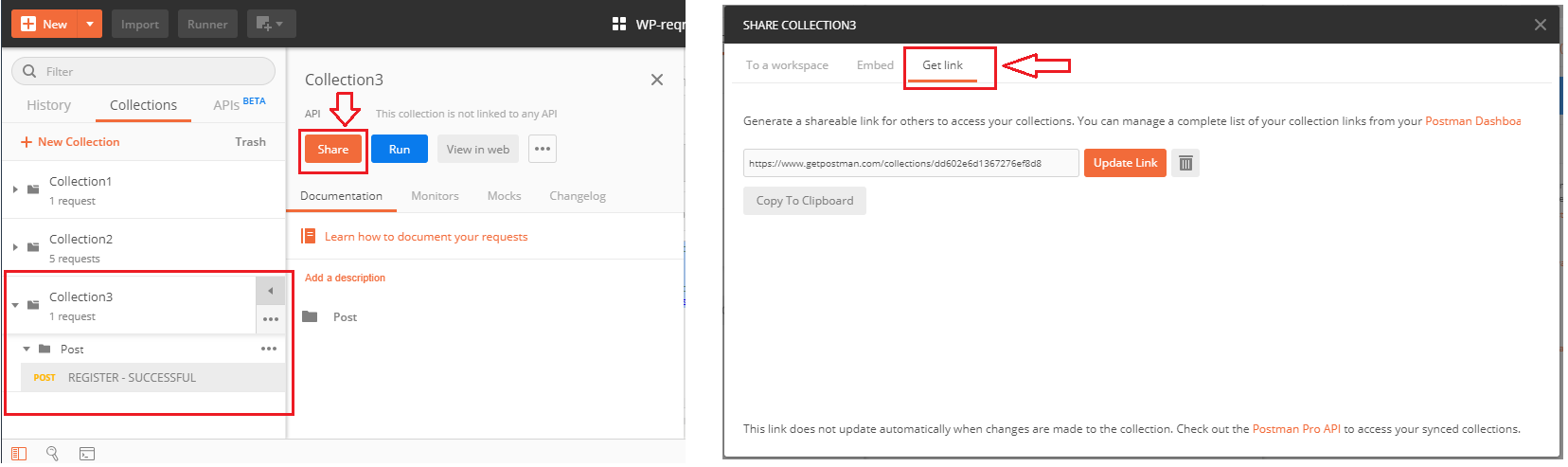
1. Export la collection comme un fichier json
2. Exécuter la commande : newman run nomFichierCollection.json

Pour plus d’information sur newman exécuté la commande : newman run -h

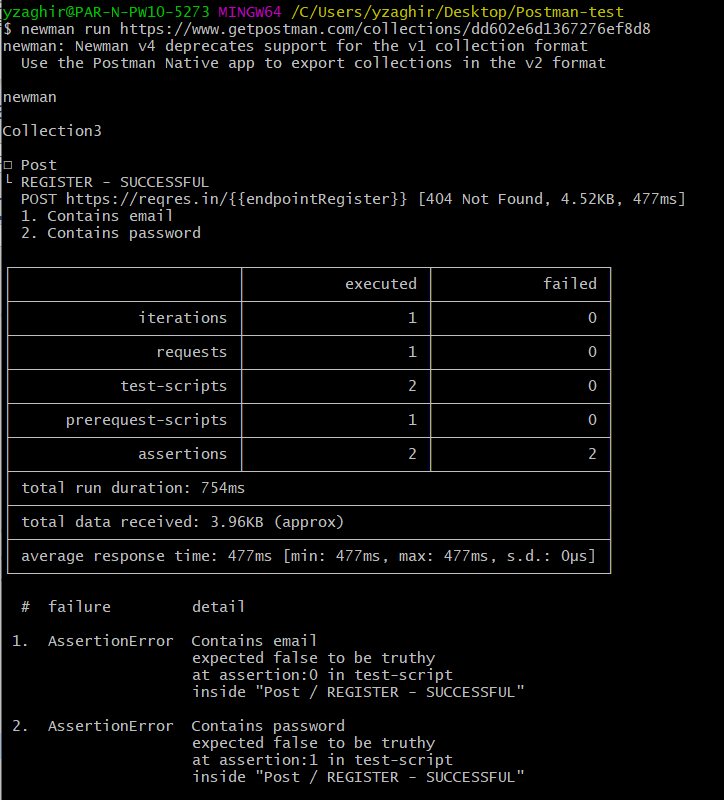
**Exécuter une collection à distance par une Url**

Récupérer l’url de la collection , exemple :

https://www.getpostman.com/collections/dd602e6d1367276ef8d8

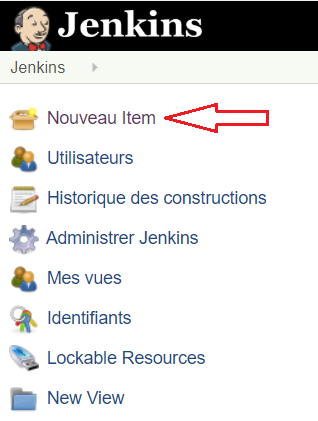


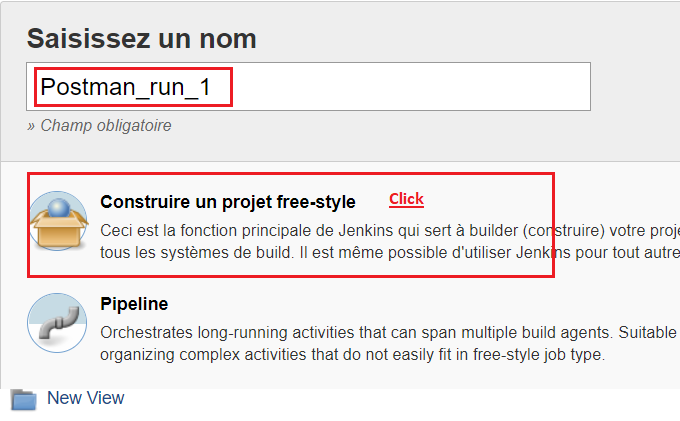
Dans une autre machine on peut lancer la collection, mais à savoir qu’il faut installer newman dans la machine distante



**Exécuter avec Jenkins**

Créer un nouveau Job





Sélectionner Exécute windows batch command dans l’option build puis ajouter le petit script

