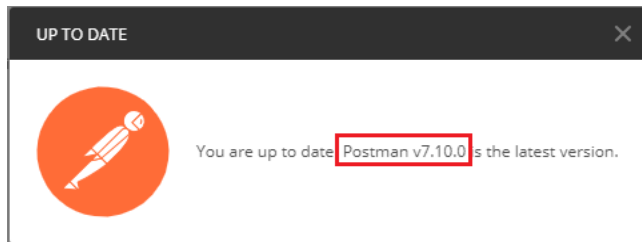


Table des matières

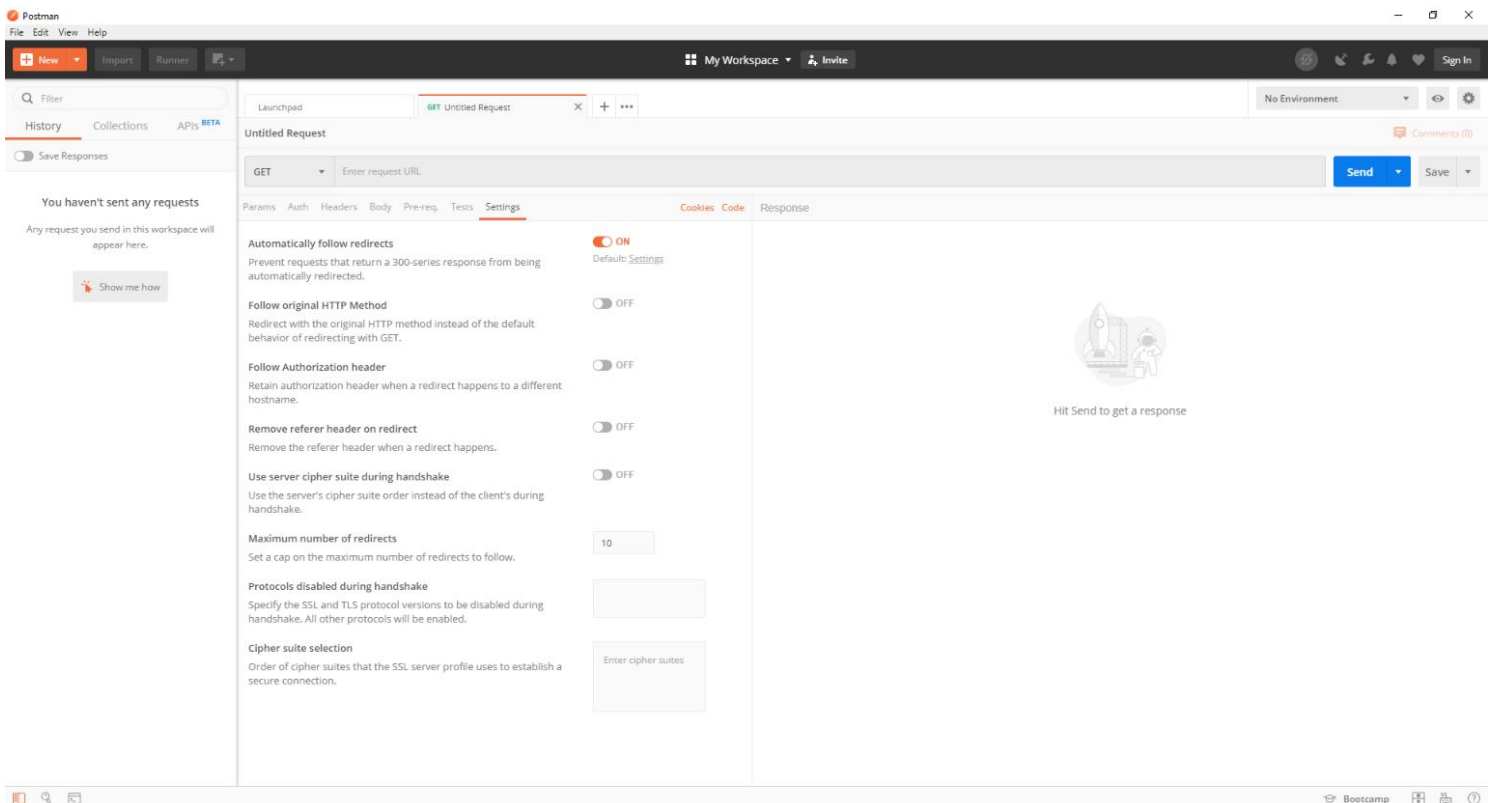
Settings	3
How to create First API Request	4
Collections :	5
Exécuter une collection	6
Les variables	8
Variable niveau collection	8
Variable niveau Globale	9
Afficher la console de postman	10
Utilisation des variables à partir des scripts en javascript	11
Les environnements	12
Création des scripts rapidement avec Postman :	16
Comment créer le premier test dans Postman	17
Workspace	18
Data file , comment récupération des données à partir d'un fichier CSV, JSON et faire les tests	20
Comment récupérer les données à partir d'un fichier csv	23
Comment récupérer les données à partir d'un fichier Json	24
Créer des tests	25
Exécuter Postman à partir de l'invite de commande (Newmane)	25
Exécuter une collection à distance par une Url	26
Exécuter avec Jenkins	28

Postman

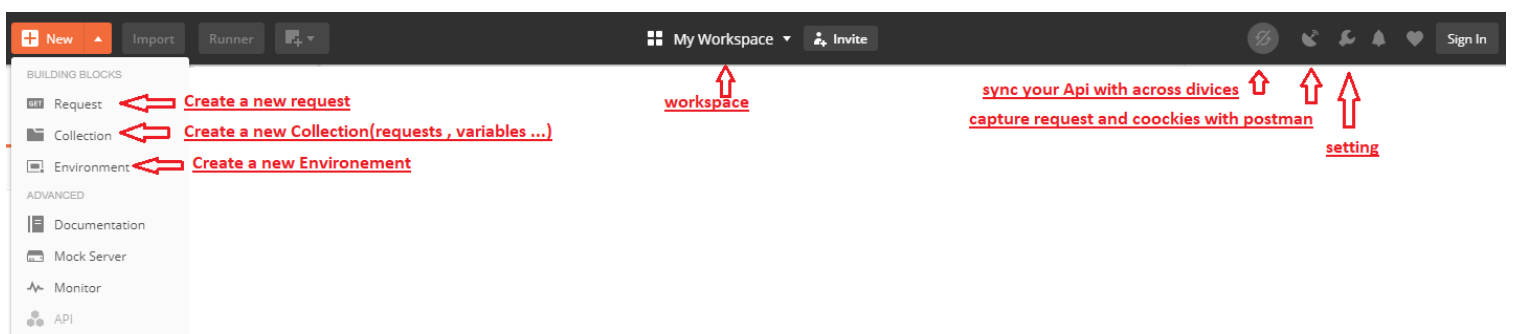
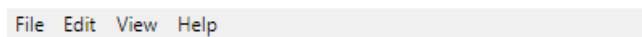
Dans cette documentation on va utiliser Postman version 7.0.0



Interface UI postman



Menu



Settings

NewImportRunner

My WorkspaceInvite

Settings

Release NotesDocumentationSecuritySupport@postmanclient

SETTINGS

GeneralThemesShortcutsDataAdd-onsCertificatesProxyUpdateAbout

REQUEST

Trim keys and values in request bodyOFF

New Code Generation ModeOFF

SSL certificate verificationON

Always open requests in new tabOFF

Always ask when closing unsaved tabsON

Language detectionAuto

Request timeout in ms (0 for infinity)0

Max response size in MB (0 to infinity)50

Automatically persist variable values enabling this will persist the current value of variables to the initial value at the end of every request execution.OFF

Learn more about variable values

WORKING DIRECTORY

Collaborate on files used in requests by sharing your working directory.

Learn how to set up your working directory

Location

C:\Users\yraghir\Postman\filesChoose

HEADERS

Send no-cache headerON

Send Postman Token headerON

Retain headers when clicking on linksOFF

Automatically follow redirectsON

Send anonymous usage data to PostmanON

USER INTERFACE

Editor Font Size (px)12

Two-pane viewOFF

Show icons with tab namesON

Variable autocompleteON

Open LaunchpadON

SETTINGS

GeneralThemesShortcutsDataAdd-onsCertificatesProxyUpdateAbout

Choose either the light or dark theme for the Postman app.

Choose your theme

SETTINGS

GeneralThemesShortcutsDataAdd-onsCertificatesProxyUpdateAbout

Shortcuts

ON

TABS

Open New TabCtrl + T

Close TabCtrl + W

Force Close TabCtrl + Alt + W

Switch To Next TabCtrl + Tab

Switch To Previous TabCtrl + Shift + Tab

Switch To Tab at PositionCtrl + 1 through Ctrl + 8

Switch To Last TabCtrl + 9

Reopen Last Closed TabCtrl + Shift + t

SETTINGS

GeneralThemesShortcutsDataAdd-onsCertificatesProxyUpdateAbout

Export data

Export all your collections, environments, globals and header presets to a single dump file.

Note: Exporting a data dump will save your globals from different workspaces as individual environments.

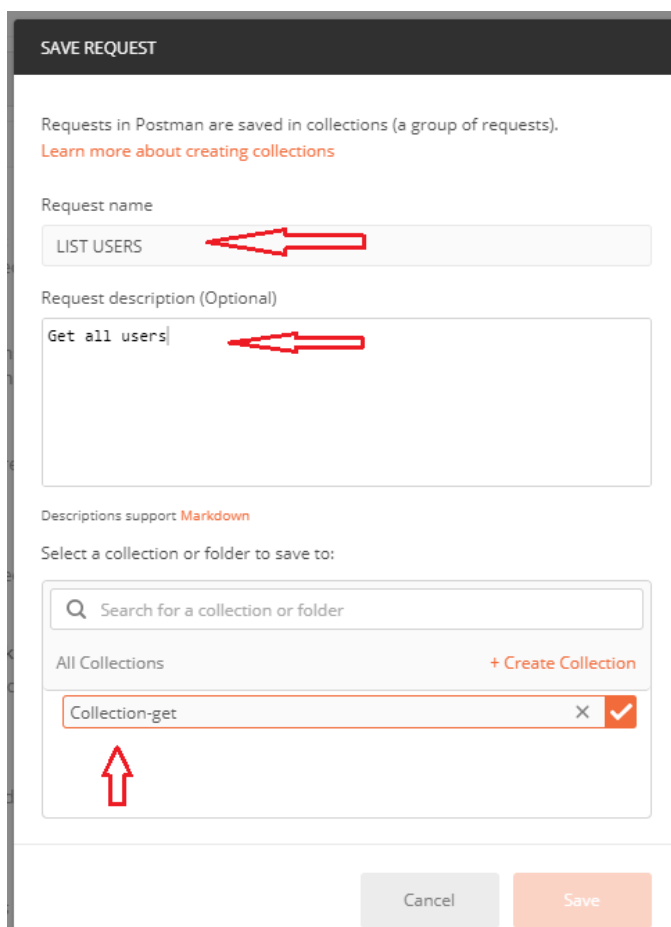
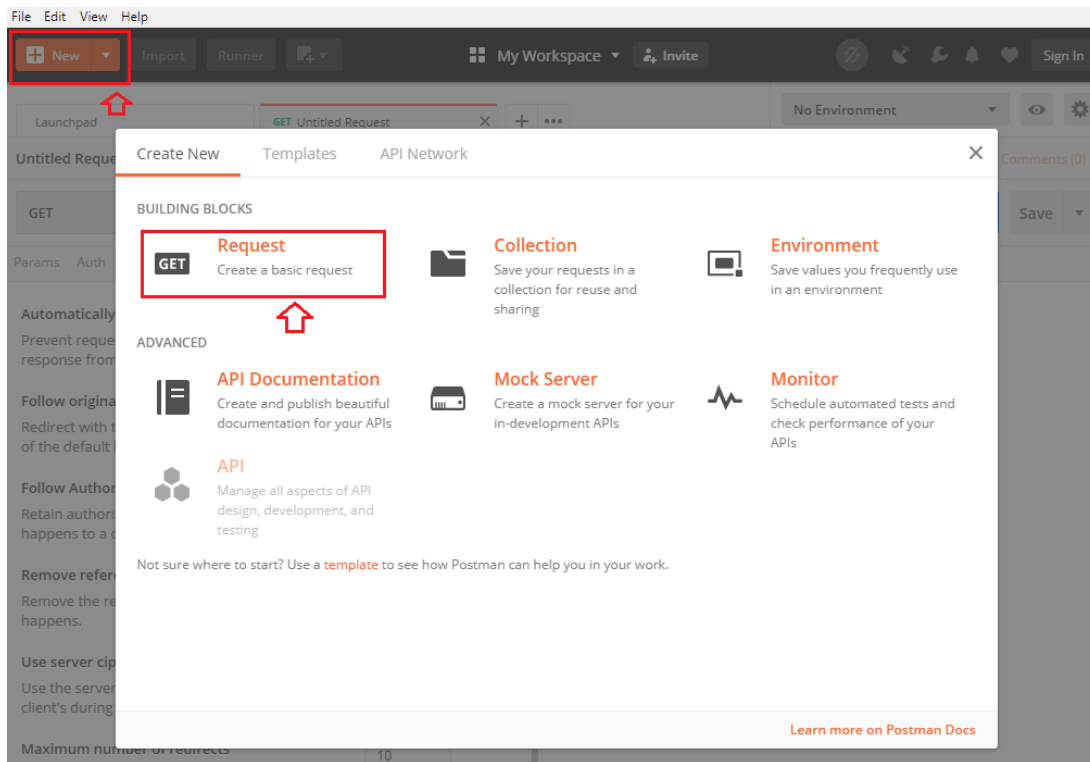
Download

Import data

Import a Postman data dump. This may overwrite existing data.

Sélect. fichiersAucun fichier choisi

How to create First API Request



body response with pretty

execute

Save response into file

detail of response

copy response

search word in body response

Collections :

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

Créer une nouvelle collection

Create New

Collection
Save your requests in a collection for reuse and sharing

Request
Create a basic request

Environment
Save values you frequently use in an environment

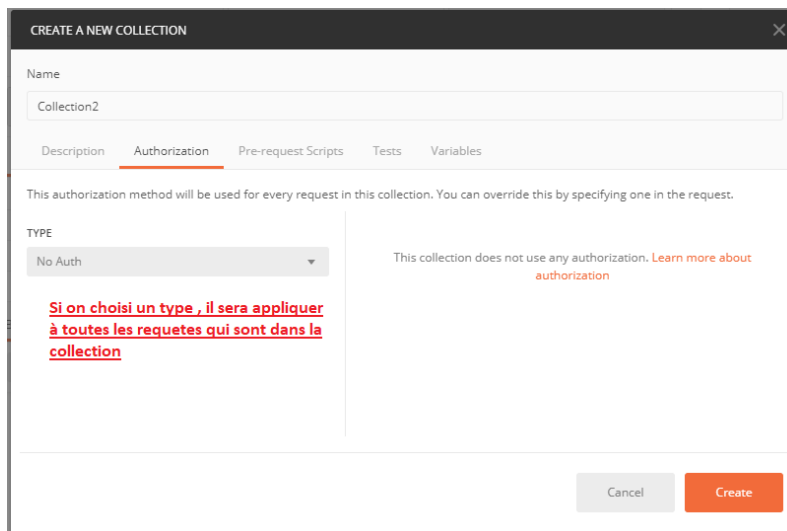
API Documentation
Create and publish beautiful documentation for your APIs

Mock Server
Create a mock server for your in-development APIs

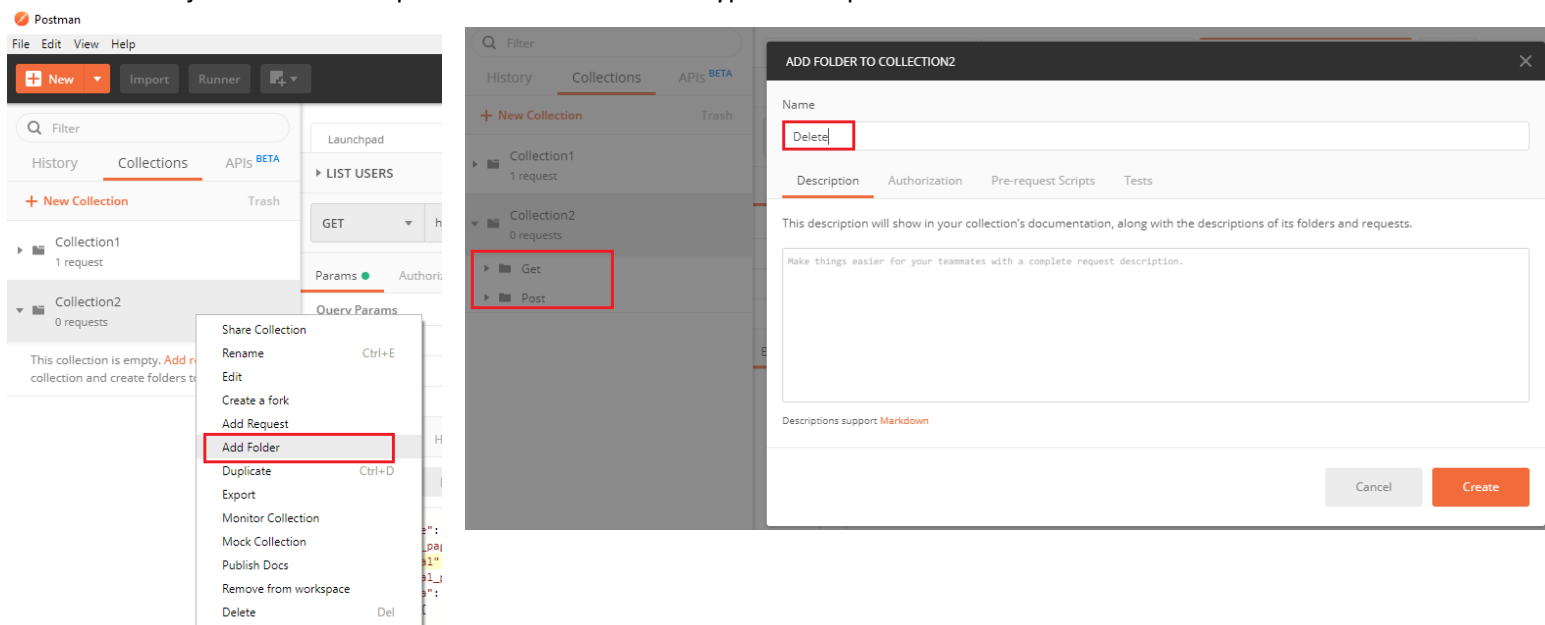
Monitor
Schedule automated tests and check performance of your APIs

Not sure where to start? Use a [template](#) to see how Postman can help you in your work.

[Learn more on Postman Docs](#)



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>

File Edit View Help

New Import Runner

My Workspace Invite

Filter

History Collections APIs BETA

+ New Collection Trash

Collection1
1 request

Collection2
3 requests

Get
GET LIST USERS
GET SINGLE USER
GET LIST <RESOURCE>

Post
This folder is empty. Add requests to this folder and create subfolders to organize them

Delete
This folder is empty. Add requests to this folder and create subfolders to organize them

Collection2

API You seem to be offline.

Share Run

Documentation Monitors Mocks Changelog

Learn how to document your requests

Add a description

Get
Post
Delete

Collection Runner

File Edit View Help

Collection Runner My Workspace Run In Command Line

Choose a collection or folder

Search for a collection or folder

Collection2

Get
Post
Delete

Choisir l'environnement

Environment No Environment

Iterations 1 nombre de fois pour executer les requetes dans la collection

Delay 0 ms temps entre chaque requete

Log Responses For all requests

Data Select File

Keep variable values

Run collection without using stored cookies

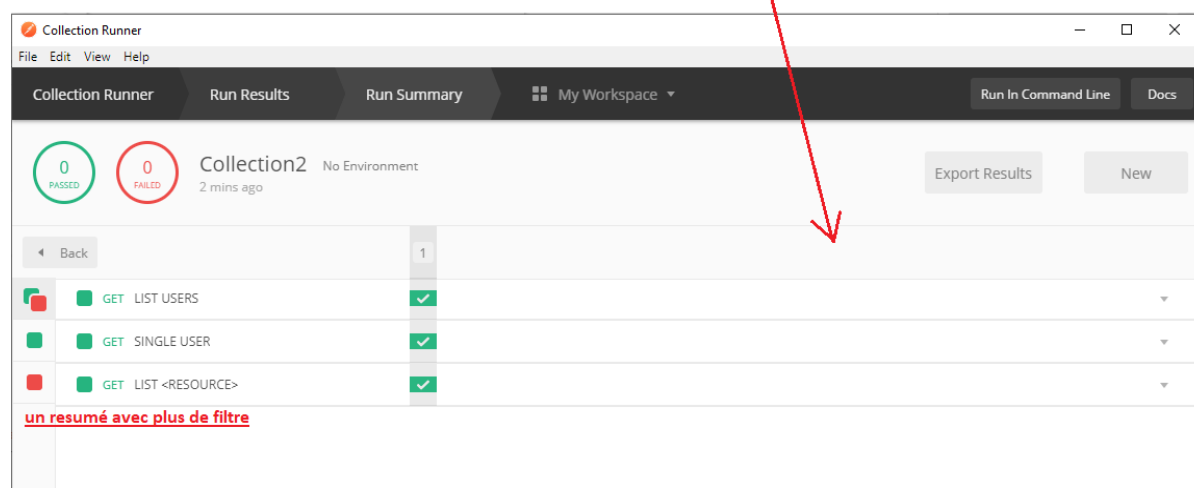
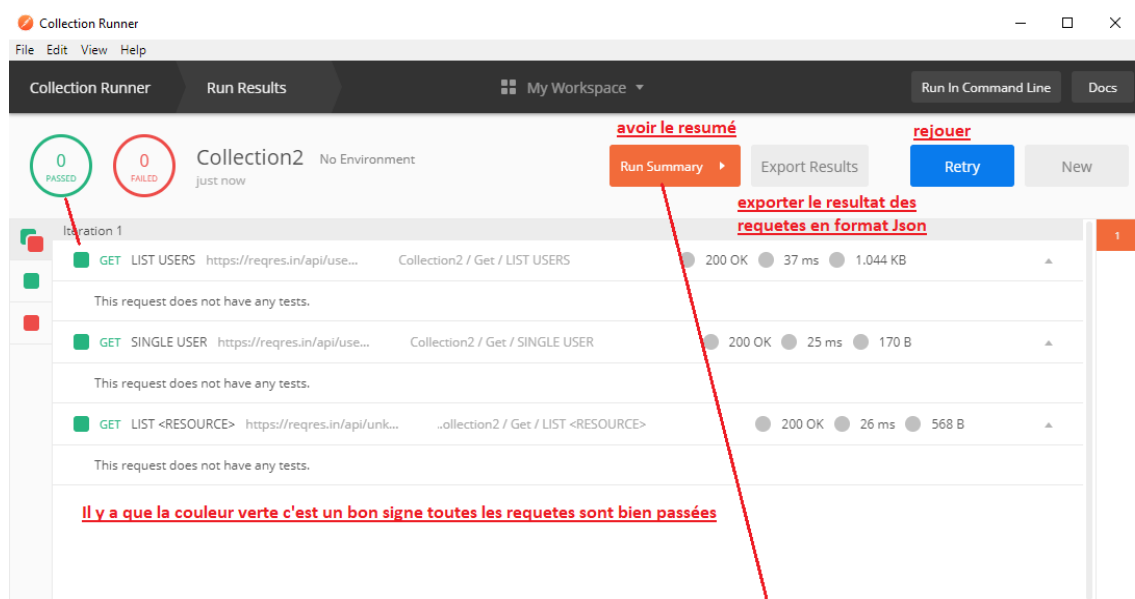
Save cookies after collection run

Run Collection2 Executer les tests

RUN ORDER

Deselect All Select

GET LIST USERS
GET SINGLE USER
GET LIST <RESOURCE>



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

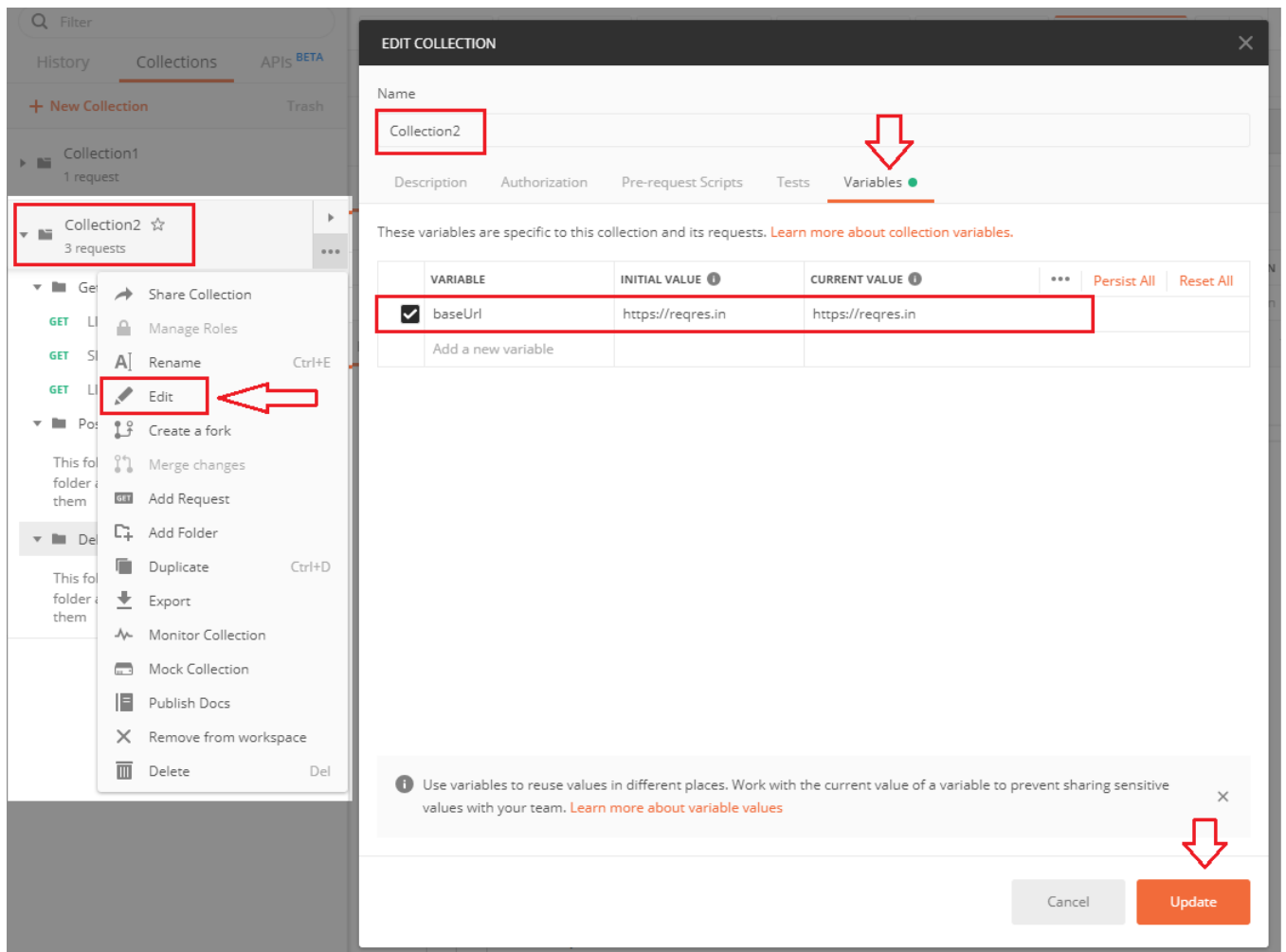
Interdire la répétition

Interdire de refaire la configuration des requêtes quand la valeur change

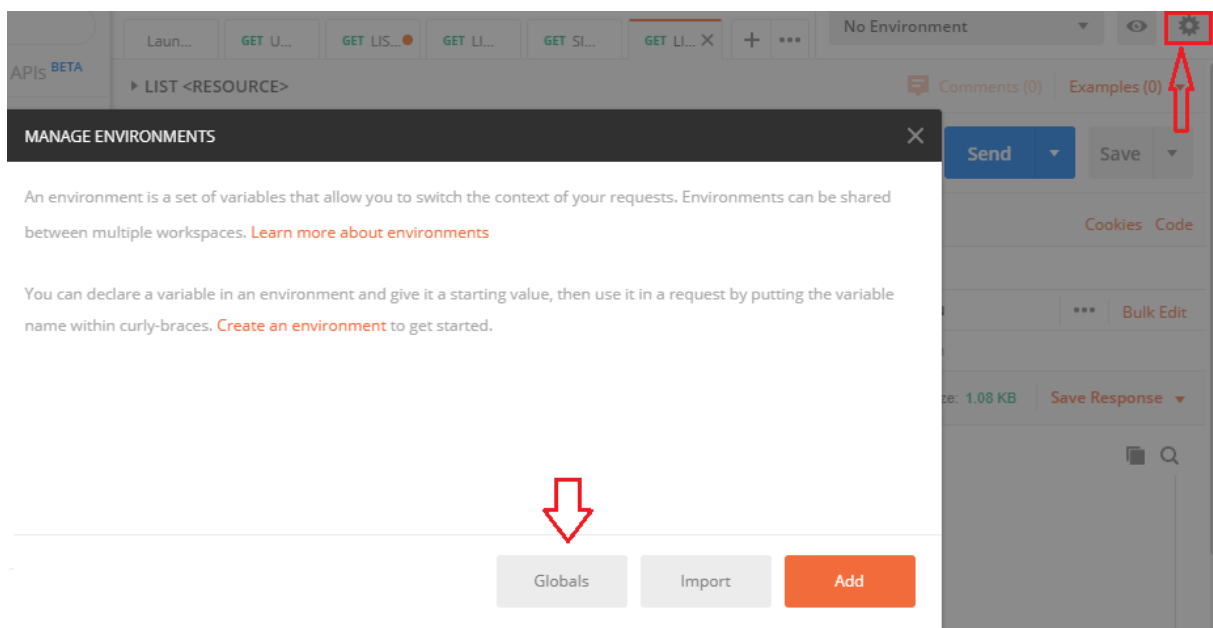
Il y a 3 scopes pour une variable. Elle peut être liée à la collection, à l'environnement, ou elle est reconnue d'une manière globale.

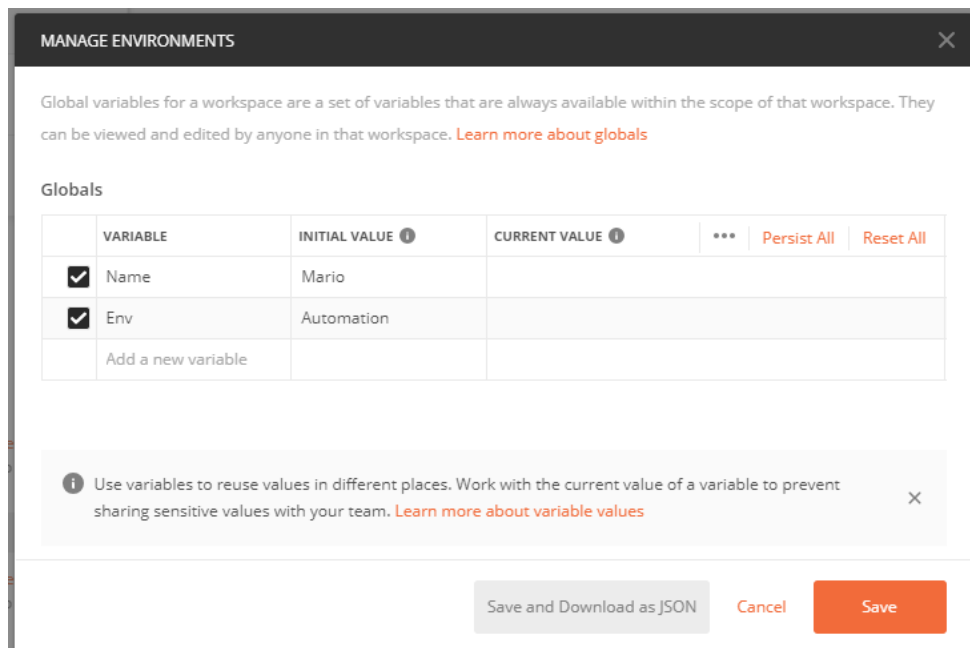
Dans notre exemple on va modifier nos uri pour que la partie static (url du site = <https://reqres.in>) soit variabiliser. Donc le format de notre request soit sous la forme `{{baseUrl}}/path`

Variable niveau collection

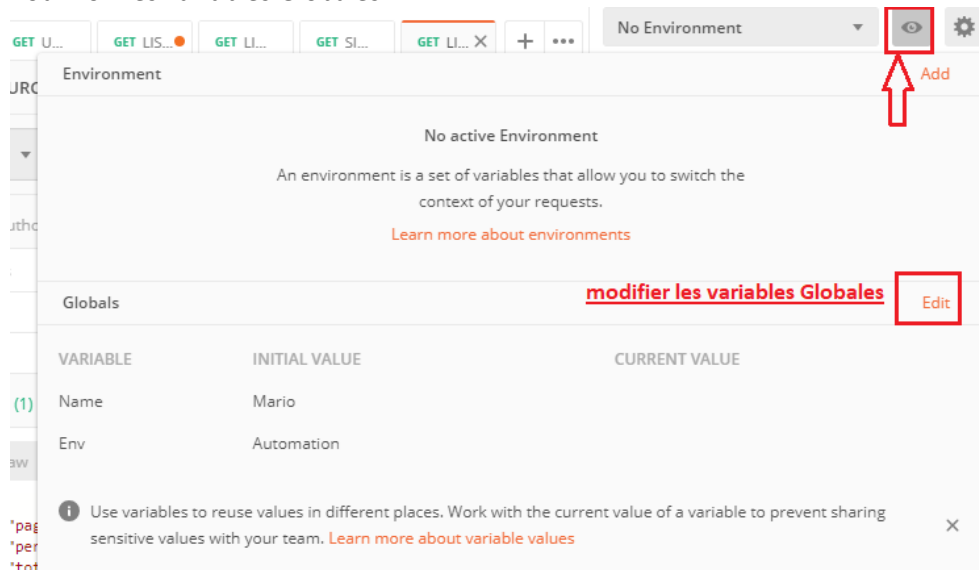


Variable niveau Globale

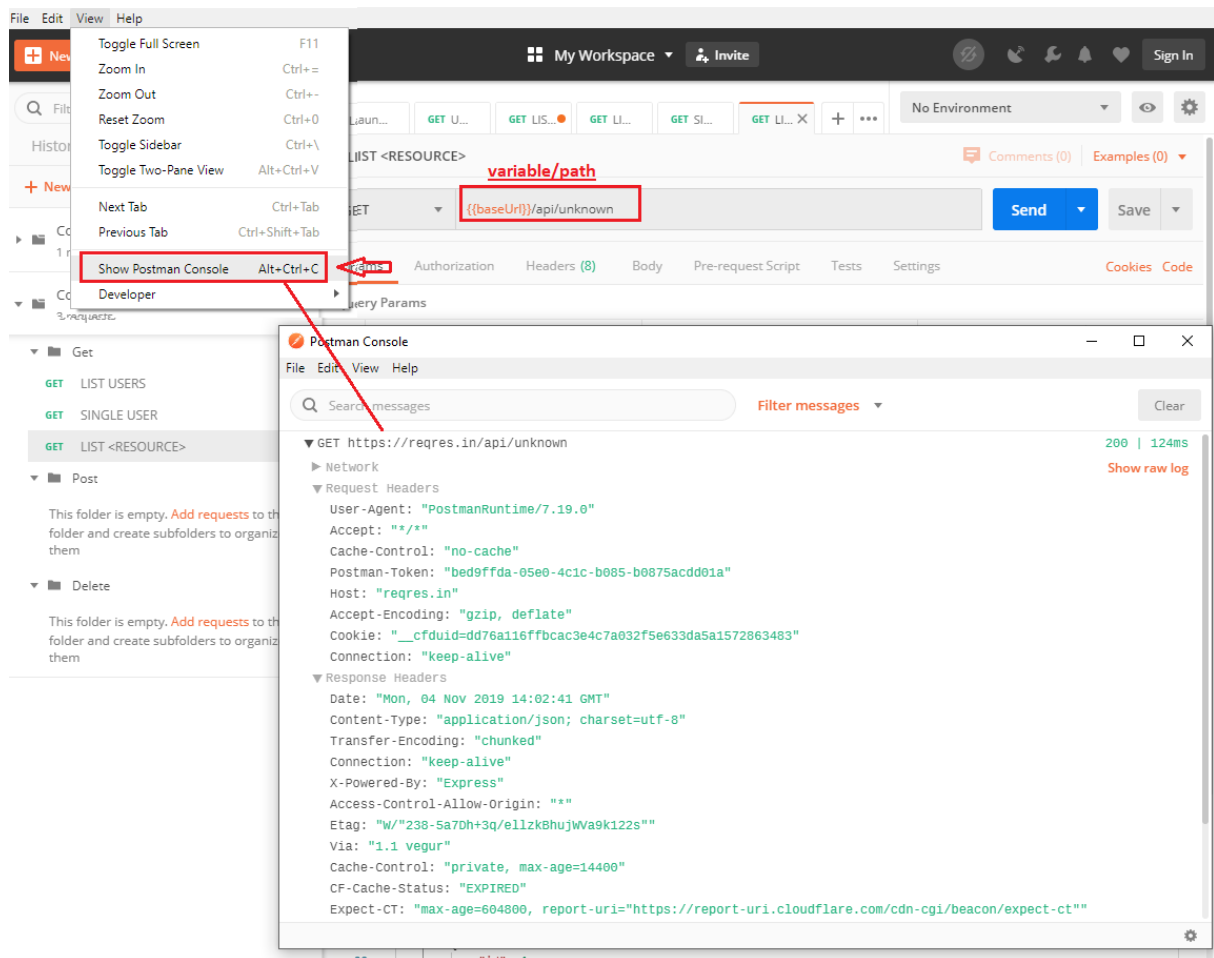




Pour voir les variables Globales



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

le test script est executé au niveau de la requete mais c'est possible de le faire au niveau de la collection

```
1 console.log("test Single User");
2
3 let baseUrl = pm.variables.get("baseUrl");
4 console.log("Value for url variable is :=> ", baseUrl);
5 // modifier la valeur d'une variable
6 pm.variables.set("baseUrl", "Postman");
7
8
9 let envVariable = pm.globals.get("Env");
10 console.log("Env avant :=> ", envVariable);
11 pm.globals.set("Env", "Pre Prod 1");
12 console.log("Env apres :=> ", envVariable);
13
14 //pm.environment.get();
15 //pm.environment.set()
```

Postman Console

```
test Single User
Value for url variable is :=> https://reqres.in
Env avant :=> Qualif
Env apres :=> Qualif
▶ GET https://reqres.in/api/users/2 200 | 30ms
test Single User
Value for url variable is :=> https://reqres.in
Env avant :=> Pre Prod
▶ GET https://reqres.in/api/users/2 200 | 28ms
test Single User
```

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 configuré dans Postman

My Workspace

Launchpad GET SINGLE USER POST REGISTER - SUCCESSFUL

Avant la configuration des environnements

POST {{baseUrl}}/api/register

Body

raw

JSON

type de media : Json

```

1 {
2   "email": "eve.holt@reqres.in",
3   "password": "pistol"
4 }

```

email et password sont fixé en dur ce qui ne facilite pas les tests dans les différents environnements

Status: 200 OK Time: 124ms Size: 472 B Save Response

JSON

```

1 {
2   "id": 4,
3   "token": "QpwL5tke4Pnpja7X4"
4 }

```

Response en format Json

Créer un nouvel environnement :

New

Create New Templates API Network

Deux options pour ajouter un nouveau environnement

Environment

Save values you frequently use in an environment

Request

Create a basic request

Collection

Save your requests in a collection for reuse and sharing

API Documentation

Create and publish beautiful documentation for your APIs

Mock Server

Create a mock server for your in-development APIs

Monitor

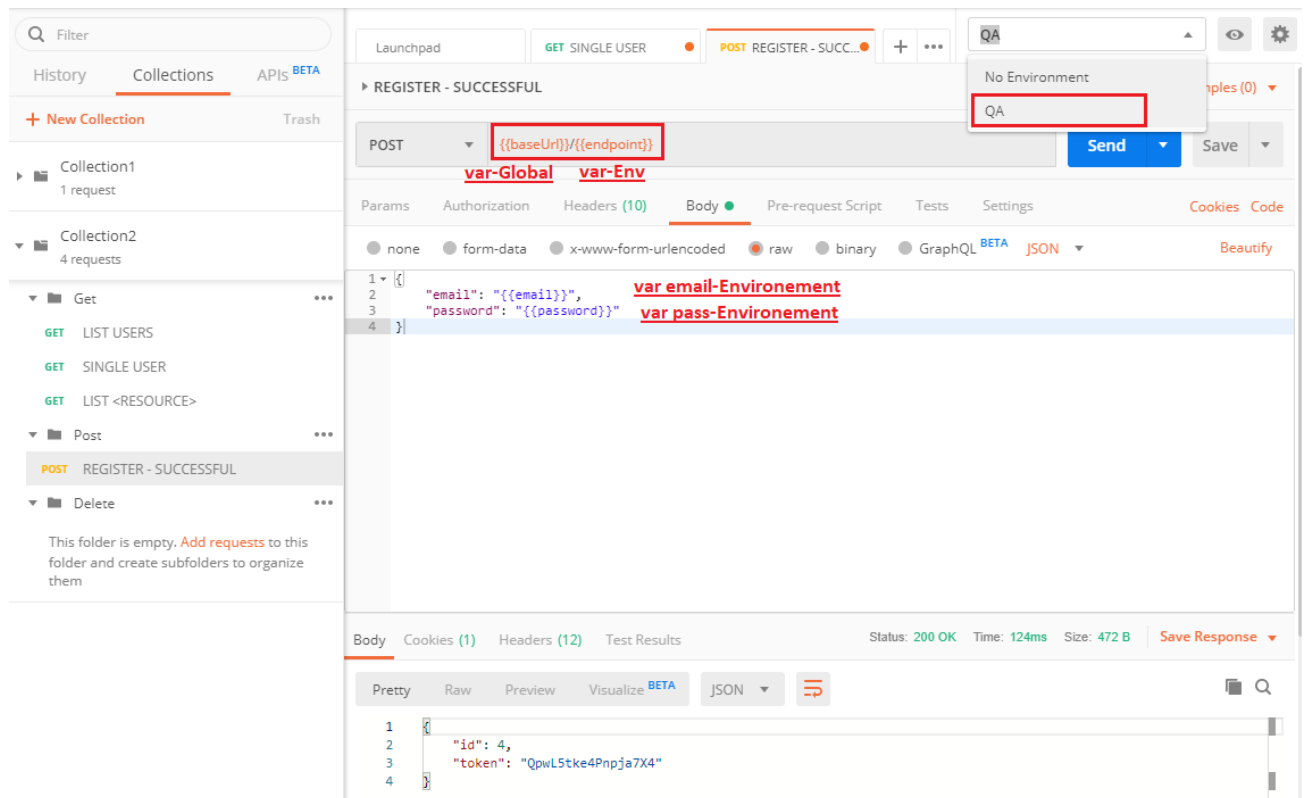
Schedule automated tests and check performance of your APIs

API

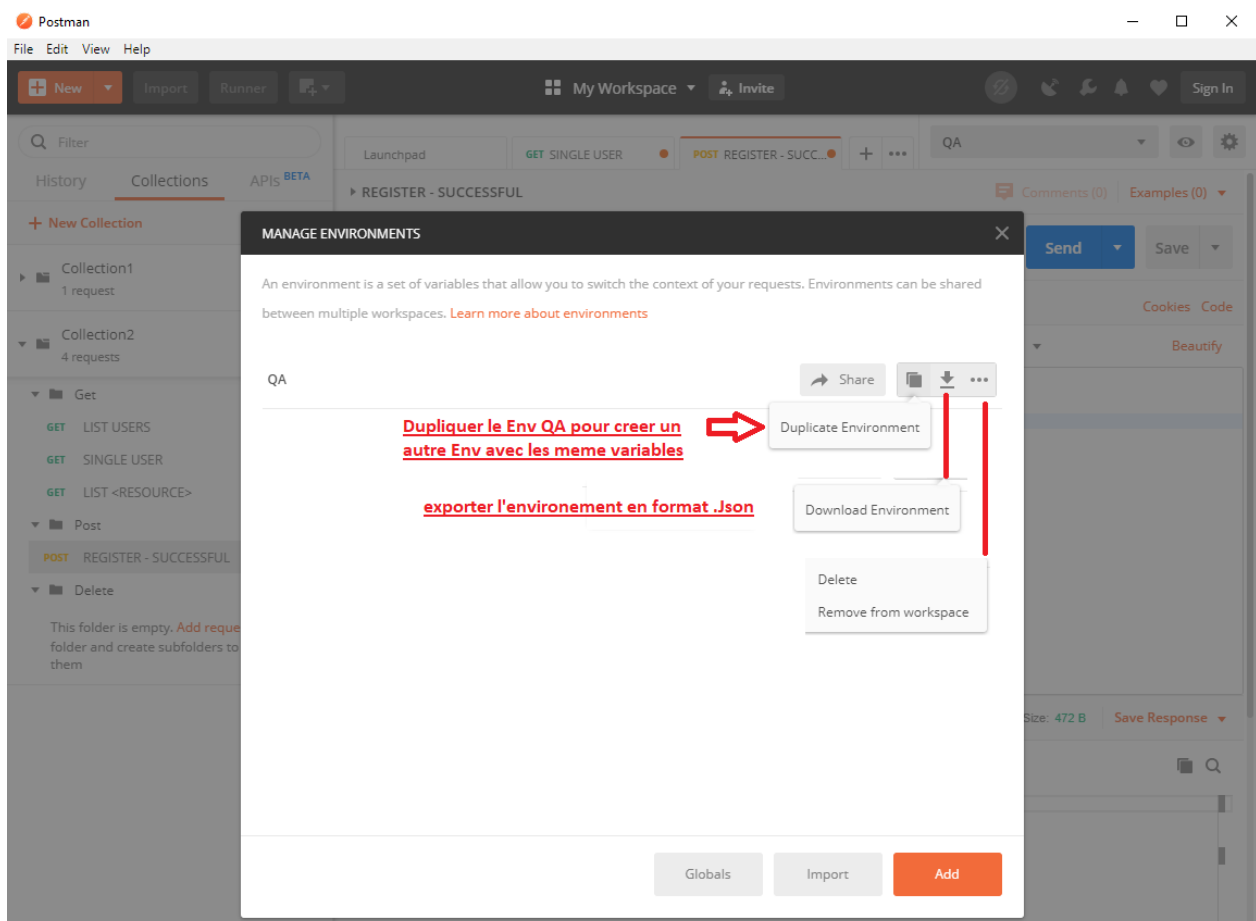
Manage all aspects of API design, development, and testing

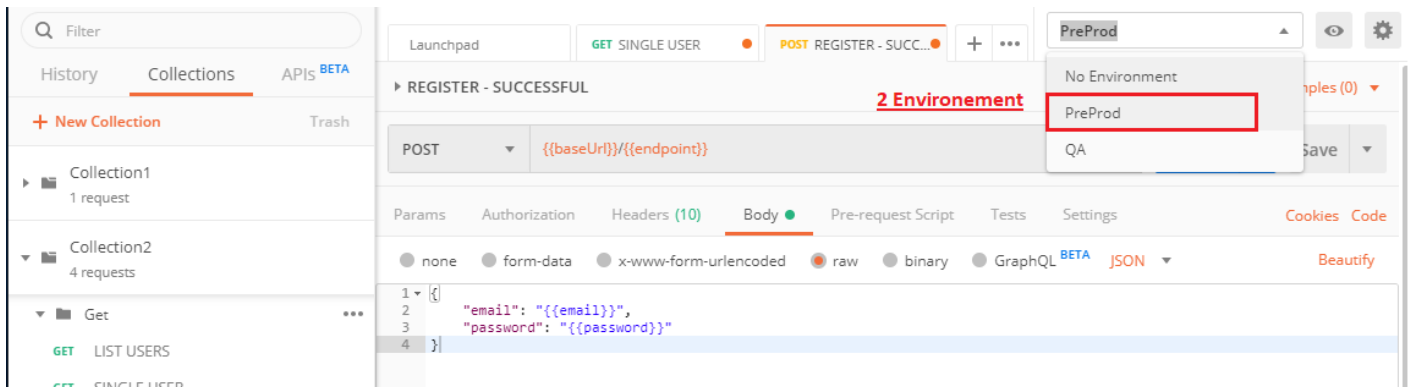
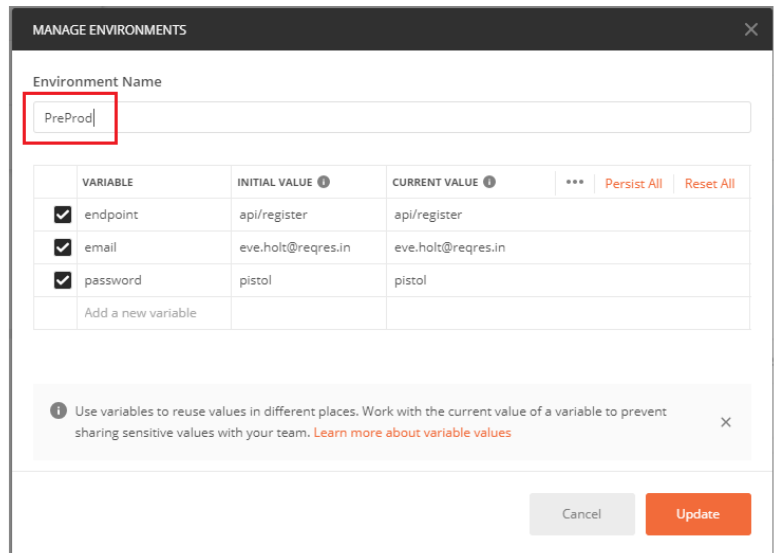
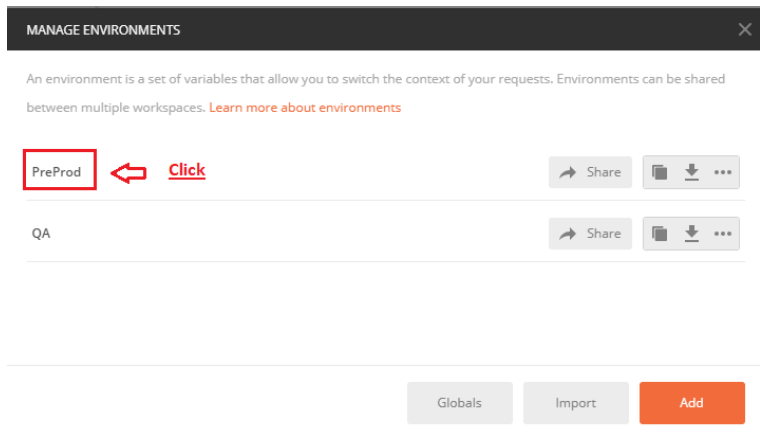
Not sure where to start? Use a [template](#) to see how Postman can help you in your work.

Learn more on Postman Docs

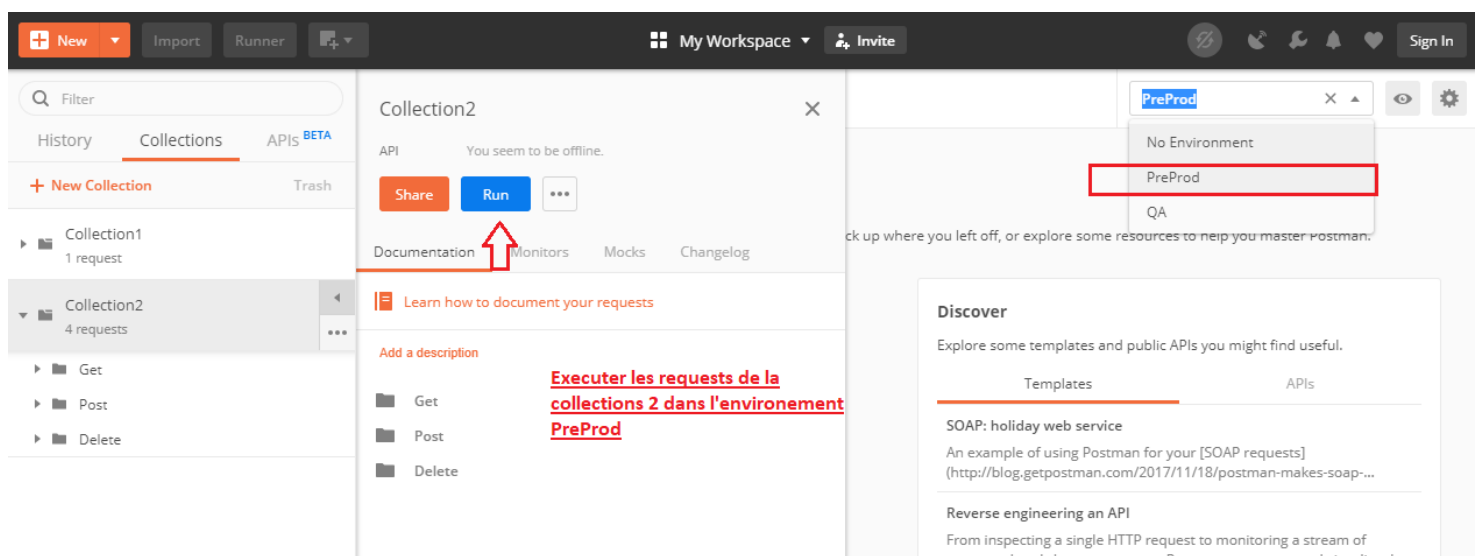


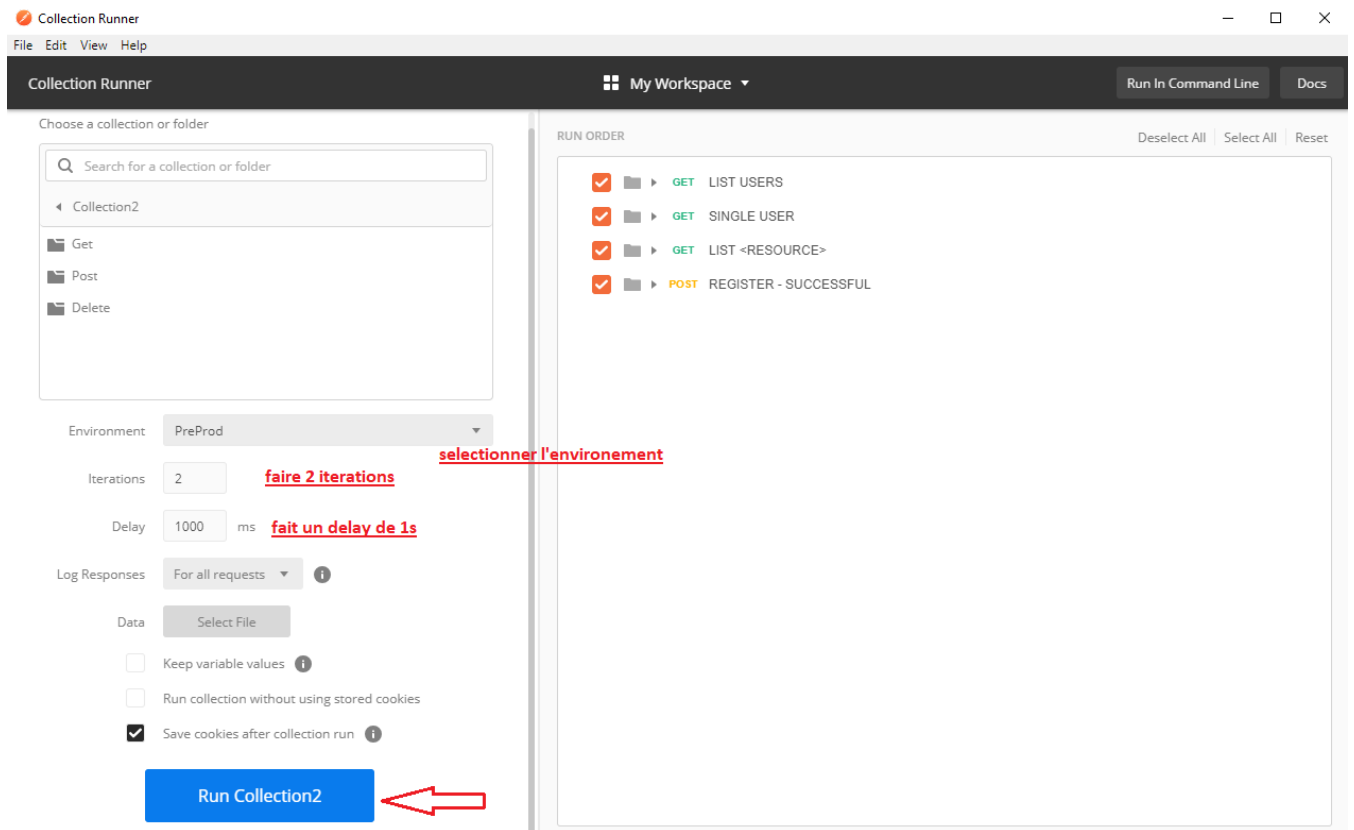
Créer un autre environnement PreProd





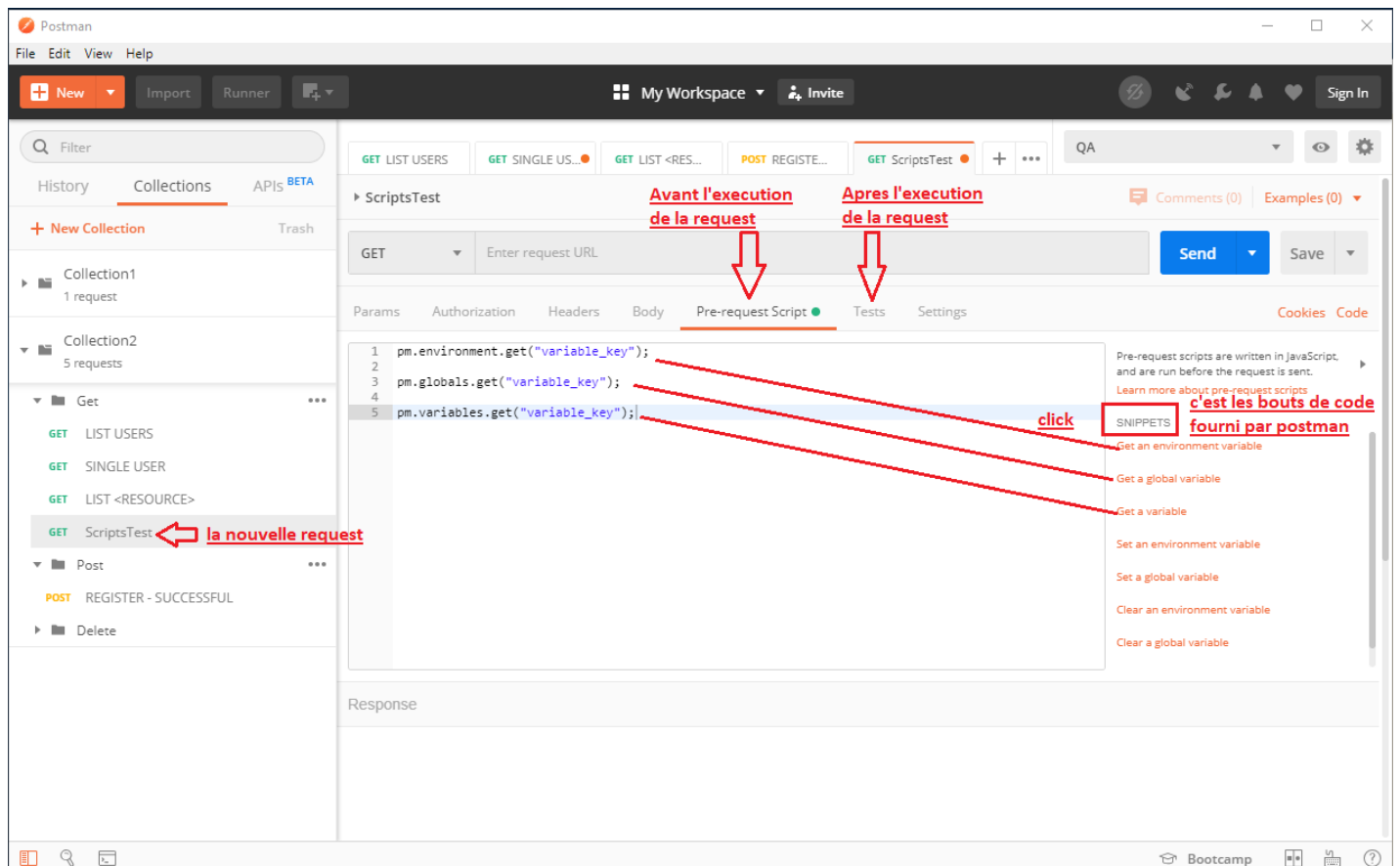
Exécuter les requests dans un environnement PreProd





Création des scripts rapidement avec Postman :

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



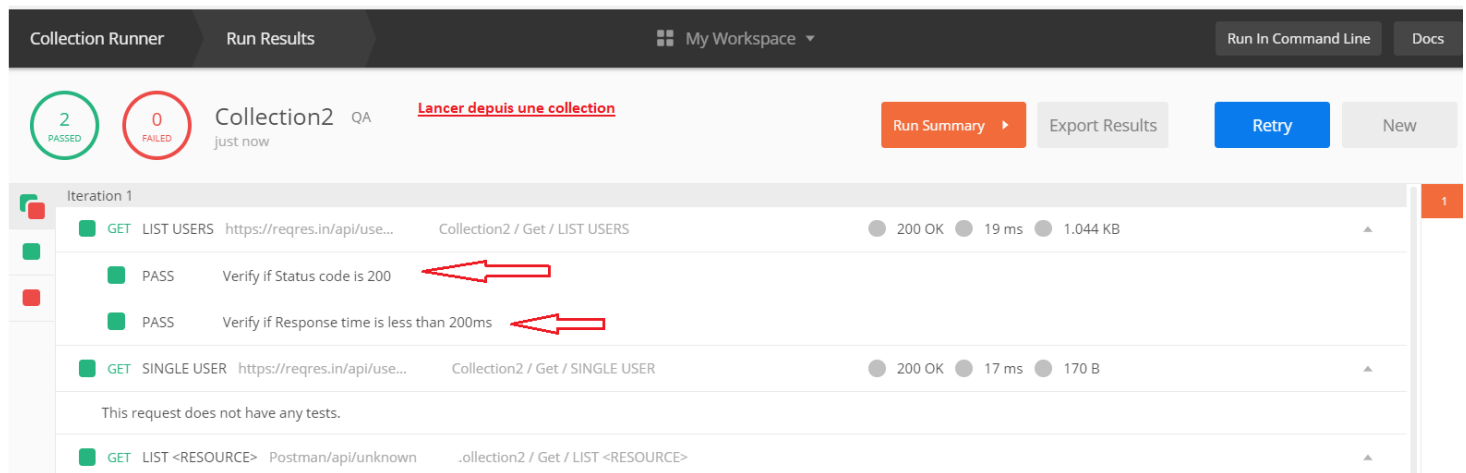
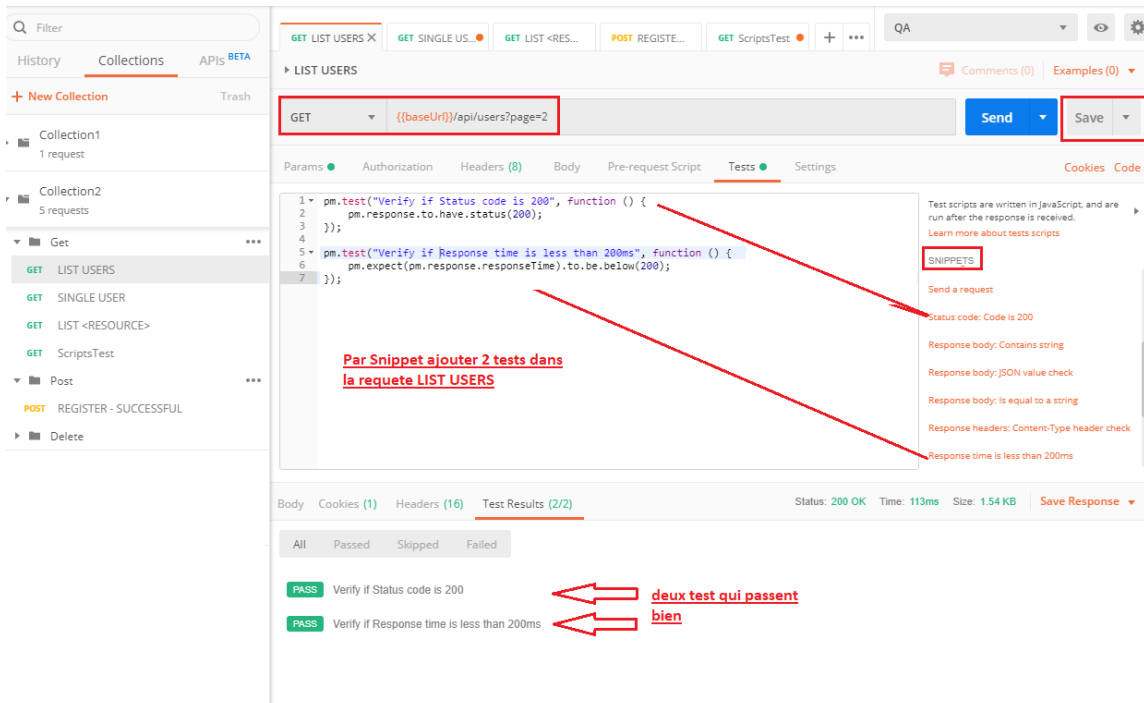
Comment créer le premier test dans Postman

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 exécutés au niveau d'une collection, d'un dossier, d'une requête

Ils peuvent être joués 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);  
});
```

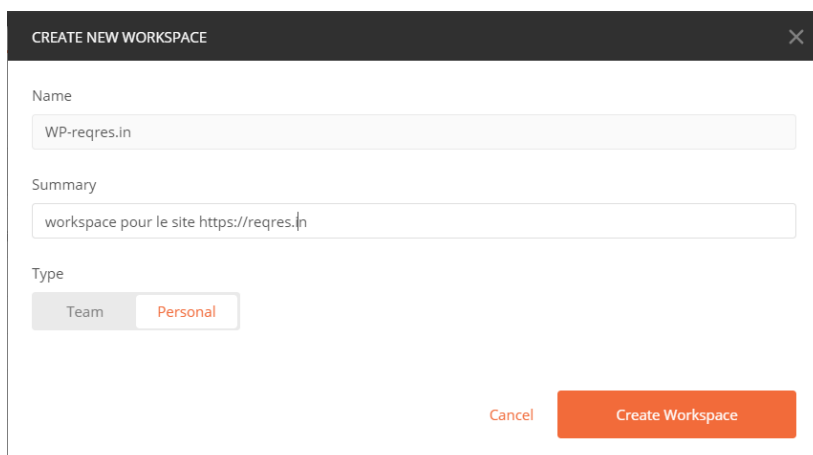
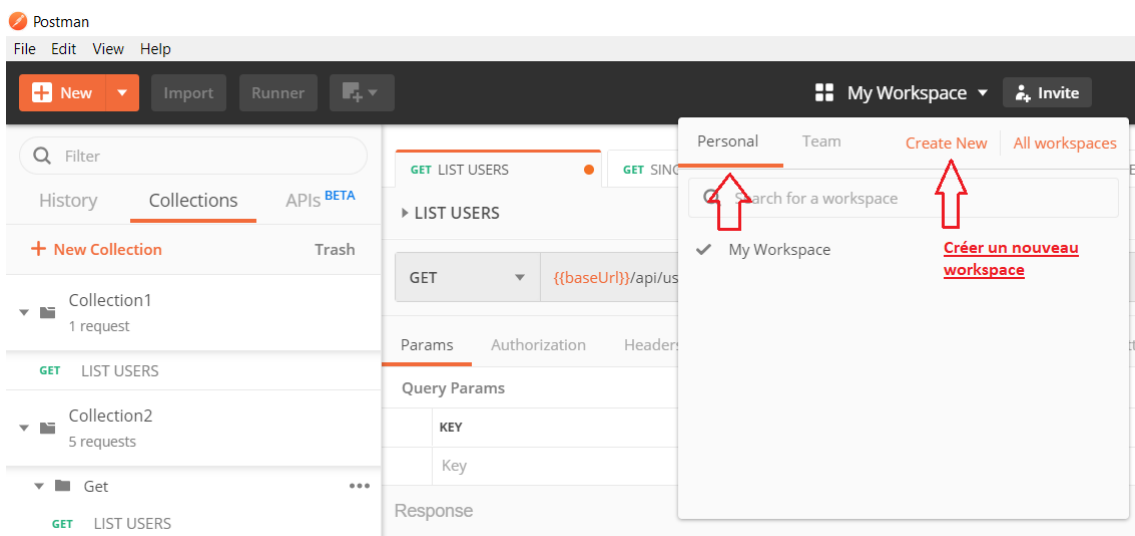
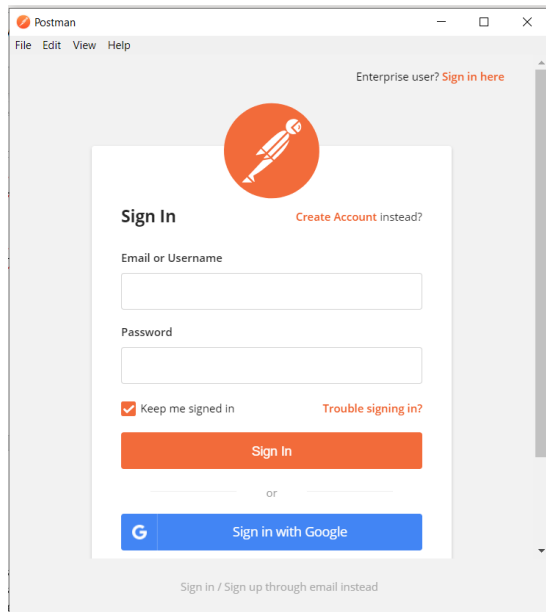
```
pm.test("Verify if Response time is less than 200ms", function () {  
  pm.expect(pm.response.responseTime).to.be.below(200);  
});
```

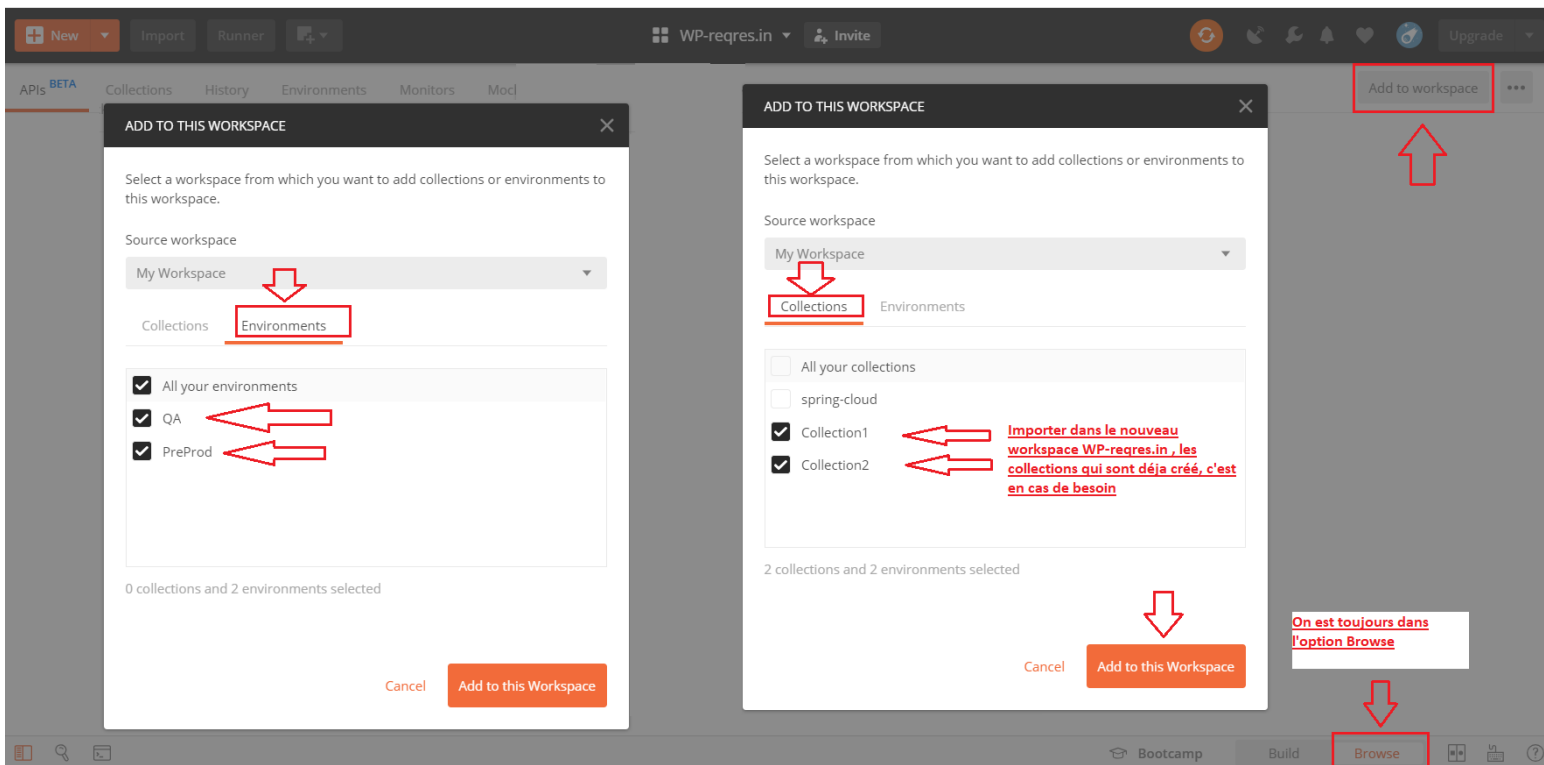
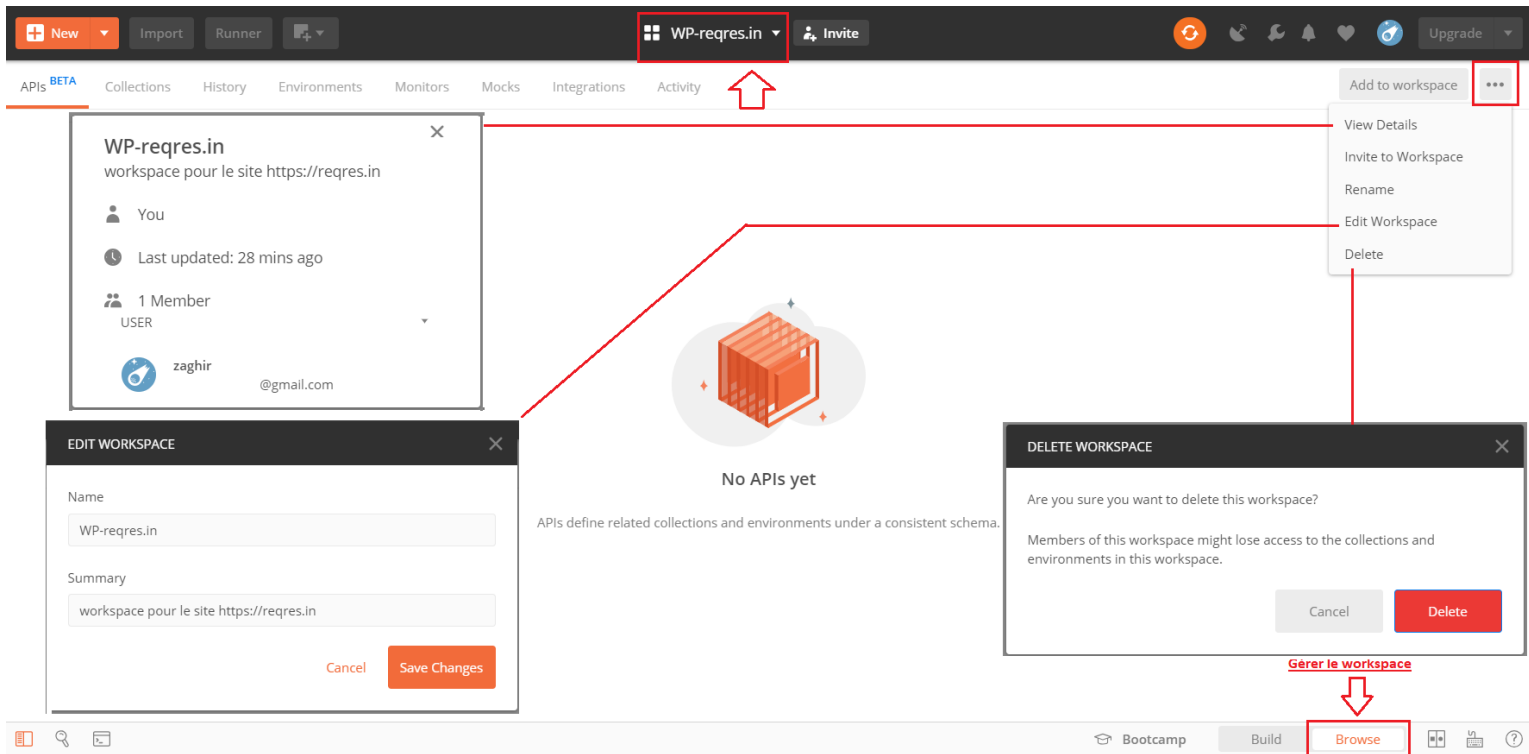
Workspace

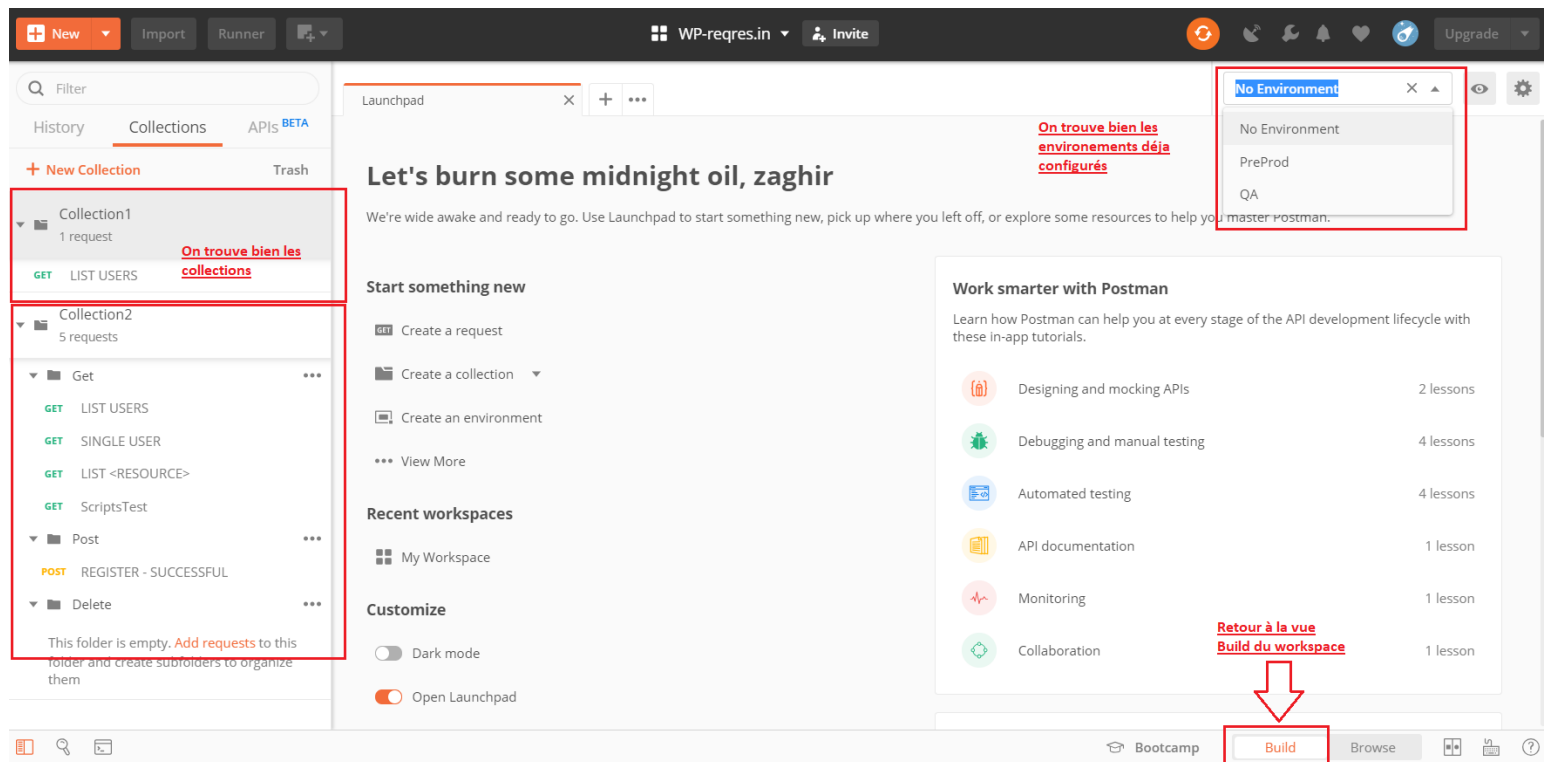
Workspace c'est quoi ?

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

Pour commencer il faut se 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/> 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"
}
```

NewImportRunner

WP-reqres.inInvite

Filter

HistoryCollectionsAPIsBETA

+ New CollectionTrash

Collection1
1 request

GET LIST USERS

Collection2
5 requests

GET LIST USERS

GET SINGLE USER

GET LIST <RESOURCE>

GET ScriptsTest

Post

POST REGISTER - SUCCESSFUL

Delete

This folder is empty. Add requests to this folder and create subfolders to organize them

LaunchpadGET Untitled RequestPOST REGISTER - SUCCESSFUL

POST{{baseUrl}}/{{endpointRegister}}

ParamsAuthorizationHeaders (10)BodyPre-request ScriptTestsSettings

Headers (1)

KEY	VALUE
<input checked="" type="checkbox"/> Content-Type	application/json
Key	Value

POST{{baseUrl}}/{{endpointRegister}}

ParamsAuthorizationHeaders (10)BodyPre-request ScriptTestsSettings

noneform-datax-www-form-urlencodedrawbinaryGraphQLBETAJSON

```
1 {  
2   "email": "{{email}}",  
3   "password": "{{password}}"  
4 }
```

Temporary Headers (9)

BodyCookies (1)Headers (12)Test Results

PrettyRawPreviewVisualizeBETAJSON

```
1 {  
2   "email": "eve.holt@reqres.in",  
3   "password": "pistol",  
4   "id": "526",  
5   "createdAt": "2019-11-05T09:08:33.984Z"  
6 }
```

PreProd

VARIABLE	INITIAL VALUE	CURRENT VALUE
endpoint	api/register	api/register
email	eve.holt@reqres.in	eve.holt@reqres.in
password	pistol	pistol
endpointRegister	api/register/preprod	api/register/preprod

Globals

No global variables

Global variables are a set of variables that are always available in a workspace.

Learn more about globals

Use variables to reuse values in different places. The current value is used while sending a request and is never synced to Postman's servers. The initial value is auto-updated to reflect the current value. Change this behaviour from Settings. Learn more about variable values

La reponse

The image shows the Postman application interface. On the left, the 'Collections' tab is active, displaying a list of collections. 'Collection3' is selected, showing a single request: 'POST REGISTER - SUCCESSFUL'. A red box highlights the 'Collection3' entry in the list, with a red arrow pointing to it.

The main panel displays the details for 'Collection3', which is not linked to any API. It includes tabs for 'Documentation', 'Monitors', 'Mocks', and 'Changelog'. The 'Post' folder is expanded, showing the 'REGISTER - SUCCESSFUL' request.

The 'Collection Runner' window is open on the right. It shows the 'Collection3' collection selected. The 'Environment' is set to 'PreProd', 'Iterations' is 1, and 'Delay' is 0 ms. The 'Log Responses' dropdown is set to 'For all requests'. A red box highlights the 'Data' section, which includes a 'Select File' button and a file named 'postman-data-request-register-successful.csv'. A red arrow points from the text 'click et sélectionnée le Fichier créé avec les données en entrée : postman-data-request-register-successful.csv' to the 'Select File' button.

The 'RUN ORDER' panel on the right shows the 'POST REGISTER - SUCCESSFUL' request as the first item in the run order.

At the bottom of the screen, the 'Run Collection3' button is visible.

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
```

PREVIEW DATA			
Iteration	endpointRegister	email	password
1	"api/register/preprod"	"eve.holt@reqres.in"	"pistol"
2	"api/register/preprod"	"test@gmail.com"	123456

Si on clique sur preview

File Edit View Help

Collection Runner Run Results WP-reqres.in Run In Command Line Docs

0 PASSED 0 FAILED Post PreProd just now Run Summary Export Results Retry New

Iteration 1
POST REGISTER - SUCCESSFUL https://reqres.in/api/regi... 201 Created 157 ms 100 B
This request does not have any tests.

Iteration 2
POST REGISTER - SUCCESSFUL https://reqres.in/api/regi... 201 Created 127 ms 96 B
Cliquer sur le nom pour avoir le detail

This request
Request URL
Request Headers (10)
Accept */*
Accept-Encoding gzip, deflate
Cache-Control no-cache
Connection keep-alive
Content-Length 51
Content-Type application/json
Cookie __cfduid=dd76a116ffb3ac3e4c7a032f5e633da5a1572863483
Host reqres.in
Postman-Token f695a998-42f3-4182-8142-ea50d977924c
User-Agent PostmanRuntime/7.19.0
Request Body
{
 "email": "test@gmail.com",
 "password": "123456"
}

On a deux lignes dans le fichier donc deux iteration

Il faut avoir le meme nom des collognes que celui des variables
Postman fait le mapping pour recuperer les valeurs dans le fichiers par Nom des variables

données à partir de fichier data .csv

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 utiliser un outil pour le vérifier exemple

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

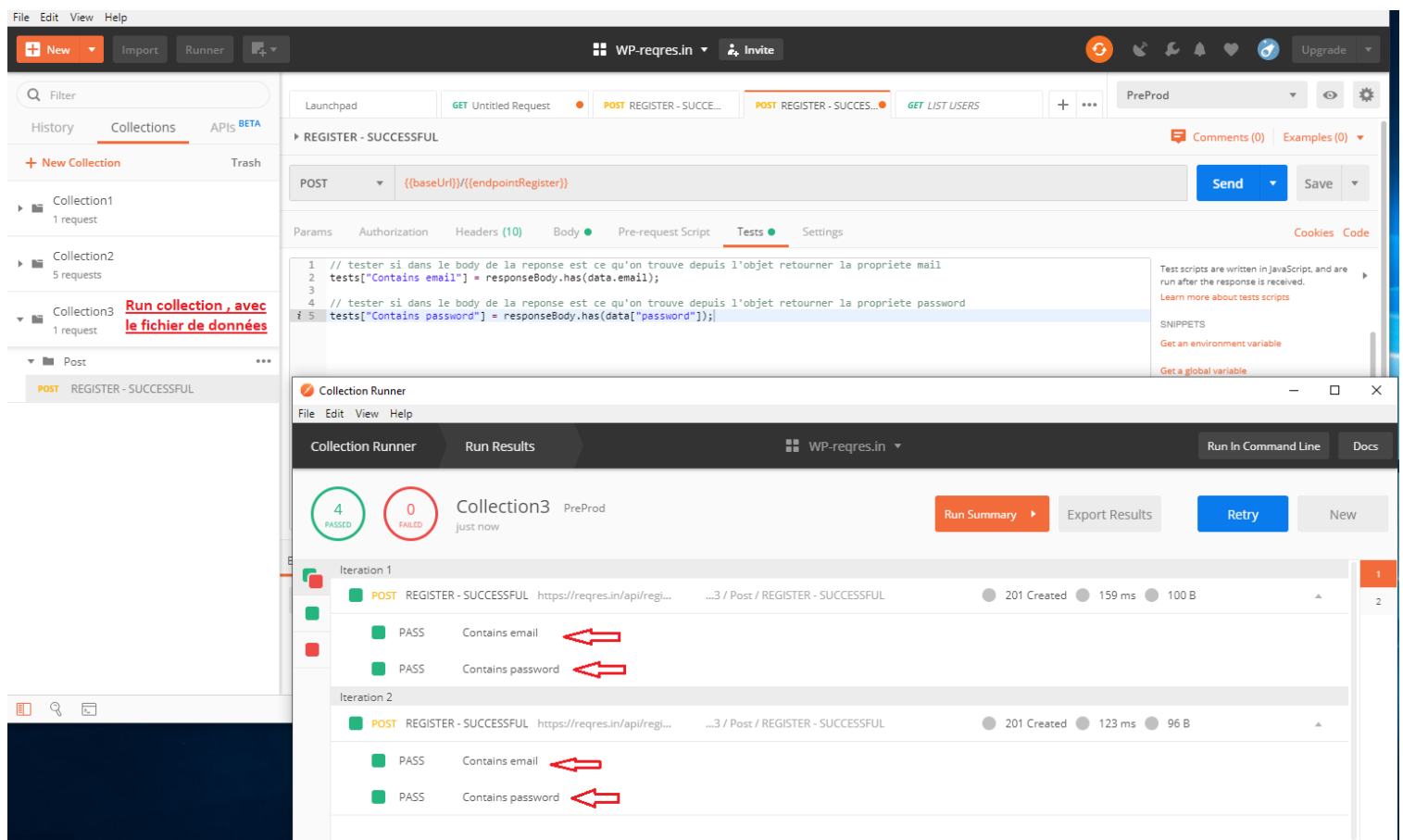
The screenshot shows the Code Beautify JSON Viewer interface. The browser address bar highlights the URL `codebeautify.org/jsonviewer`. The page header includes the Code Beautify logo and navigation links like "JSON Formatter", "Hex Color Codes", "My Ip", "Search", "Recent Links", "More", and "Sign in". The main section is titled "JSON Viewer" with a "Save & Share" button. The interface is divided into three panels. The left panel, labeled "Sample", contains the raw JSON data from the file, with a red arrow pointing to it and the text "Text saisie". The middle panel, labeled "Result mode:", contains buttons for "tree", "Load Url", "Browse", "Tree Viewer", "2 Tab Space", and "Beautify", with a red arrow pointing to the "Beautify" button. The right panel, labeled "Code", shows the JSON data after being formatted, with a red arrow pointing to it and the text "Text corrigé on peut le copier et l'utiliser".

Créer des tests

Les tests en javascript

```
// tester si dans le body de la reponse est ce qu'on trouve depuis l'objet r
etourner 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 r
etourner la propriete password
tests["Contains password"] = responseBody.has(data["password"]);
```



Exécuter Postman à partir de l'invite de commande (Newman)

Comment exécuter 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`

- 2) Installer Newman

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

Installer Newman avec : `npm install -g newman`

```
C:\Devs-and-tools\project\Workspace-sts-4\checkout>npm install -g newman
C:\Users\admyzagh\AppData\Roaming\npm\newman -> C:\Users\admyzagh\AppData\Roaming\npm\node_modules\newman\bin\newman.js
+ newman@4.5.5
added 157 packages from 195 contributors in 49.202s

C:\Devs-and-tools\project\Workspace-sts-4\checkout>
```

3) Export la collection comme un fichier json

The screenshot illustrates the steps to export a Postman collection and execute it using Newman. On the left, the Postman interface shows 'Collection2' with 5 requests. A red box highlights the 'Export' option in the menu, with a red arrow pointing to it and the text 'Exporter la collection en fichier Json'. In the center, the 'EXPORT COLLECTION' dialog box is shown, with 'Collection v2.1 (recommended)' selected and a red arrow pointing to the 'Export' button. On the right, a terminal window shows the command 'newman run Collection2.json' being executed, with a red box around it and the text 'Placer vous dans le dossier ou il est enregistré le fichier Json'. Below the terminal output, a table summarizes the execution results.

Collection2

```
Collection2
  GET https://reqres.in/api/users?page=2 [200 OK, 1.69KB, 282ms]
    / Verify if Status code is 200
    1. Verify if Response time is less than 200ms
  SINGLE USER
  GET https://reqres.in/api/users/2 [200 OK, 707B, 48ms]
    'test Single User'
    'Value for url variable is :=> ', 'https://reqres.in'
    'Env avant :=> ', null
  LIST <RESOURCE>
  GET Postman/api/unknown [errored]
  getaddrinfo ENOTFOUND postman postman:80
  ScriptsTest
  GET [errored]
    runtime:extensions~request: request url is empty
  POST
  REGISTER - SUCCESSFUL
  POST Postman/{{endpoint}} [errored]
  getaddrinfo ENOTFOUND postman postman:80
```

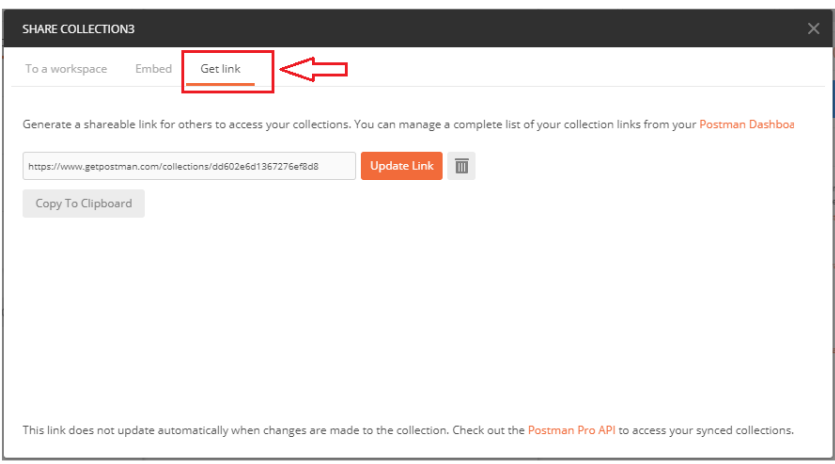
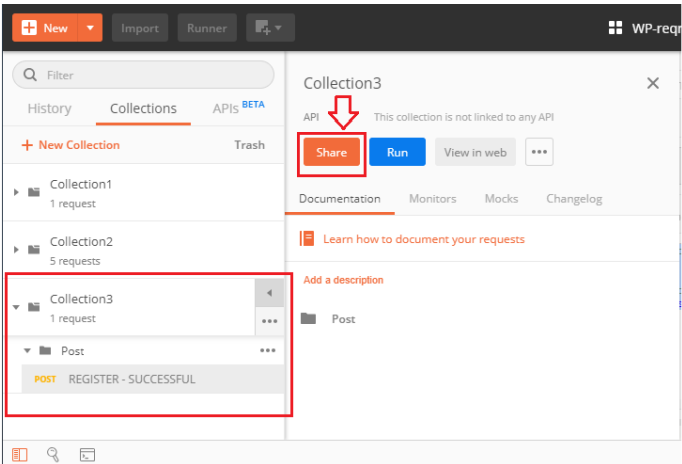
	executed	failed
iterations	1	0
requests	5	3
test-scripts	8	0
prerequisite-scripts	5	0
assertions	2	1
total run duration: 5.2s		
total data received: 1.19KB (approx)		
average response time: 165ms [min: 48ms, max: 282ms, s.d.: 117ms]		

- 4) 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

```
yzaghir@PAR-N-PW10-5273 MINGW64 /C/Users/yzaghir/Desktop/Postman-test
$ newman run https://www.getpostman.com/collections/dd602e6d1367276ef8d8
newman: Newman v4 deprecates support for the v1 collection format
      Use the Postman Native app to export collections in the v2 format

newman
Collection3

□ Post
└─ REGISTER - SUCCESSFUL
   POST https://regres.in/{{endpointRegister}} [404 Not Found, 4.52KB, 477ms]
   1. Contains email
   2. Contains password
```

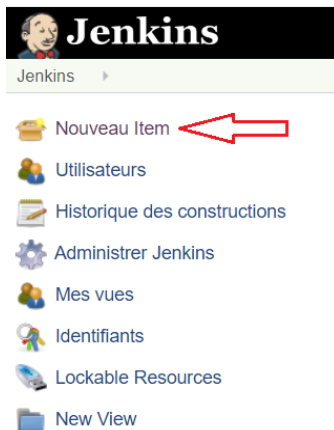
	executed	failed
iterations	1	0
requests	1	0
test-scripts	2	0
prerequest-scripts	1	0
assertions	2	2
total run duration: 754ms		
total data received: 3.96KB (approx)		
average response time: 477ms [min: 477ms, max: 477ms, s.d.: 0µs]		

```
# failure      detail
1. AssertionError Contains email
   expected false to be truthy
   at assertion:0 in test-script
   inside "Post / REGISTER - SUCCESSFUL"

2. AssertionError Contains password
   expected false to be truthy
   at assertion:1 in test-script
   inside "Post / REGISTER - SUCCESSFUL"
```

Exécuter avec Jenkins

Créer un nouveau Job



Saisissez un nom

Postman_run_1

» Champ obligatoire

Construire un projet free-style [Click](#)

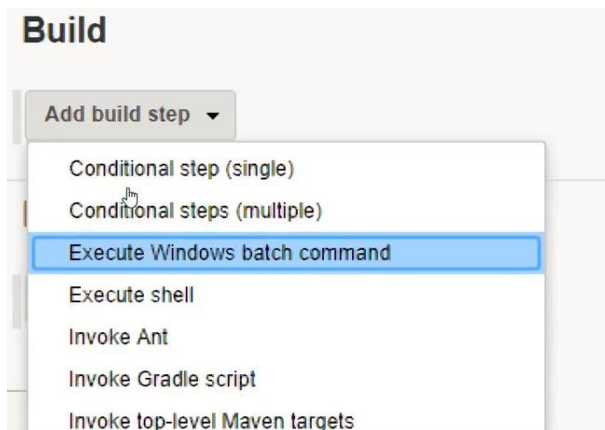
Ceci est la fonction principale de Jenkins qui sert à builder (construire) votre projet tous les systèmes de build. Il est même possible d'utiliser Jenkins pour tout autre

Pipeline

Orchestrates long-running activities that can span multiple build agents. Suitable organizing complex activities that do not easily fit in free-style job type.

[New View](#)

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



Build

Exécuter une ligne de commande batch Windows


Commande

```
cd C:\Users\yzaghir\Desktop\Postman-test
newman run Collection2.json
```

Voir [la liste des variables d'environnement disponibles](#)

[Avancé...](#)


Ajouter une étape au build ▼

 [Retour au tableau de bord](#)


 [État](#)


 [Modifications](#)

 [Répertoire de travail](#)

 [Lancer un build](#)

 [Supprimer Projet](#)

 [Configurer](#)

 [Rename](#)

 **Historique des builds** [tendance](#)

X

	#3	5 nov. 2019 16:03
	#2	5 nov. 2019 16:01
	#1	5 nov. 2019 16:01

 [RSS des builds](#)  [RSS des échecs](#)