Working Process with Postman

What is Postman?

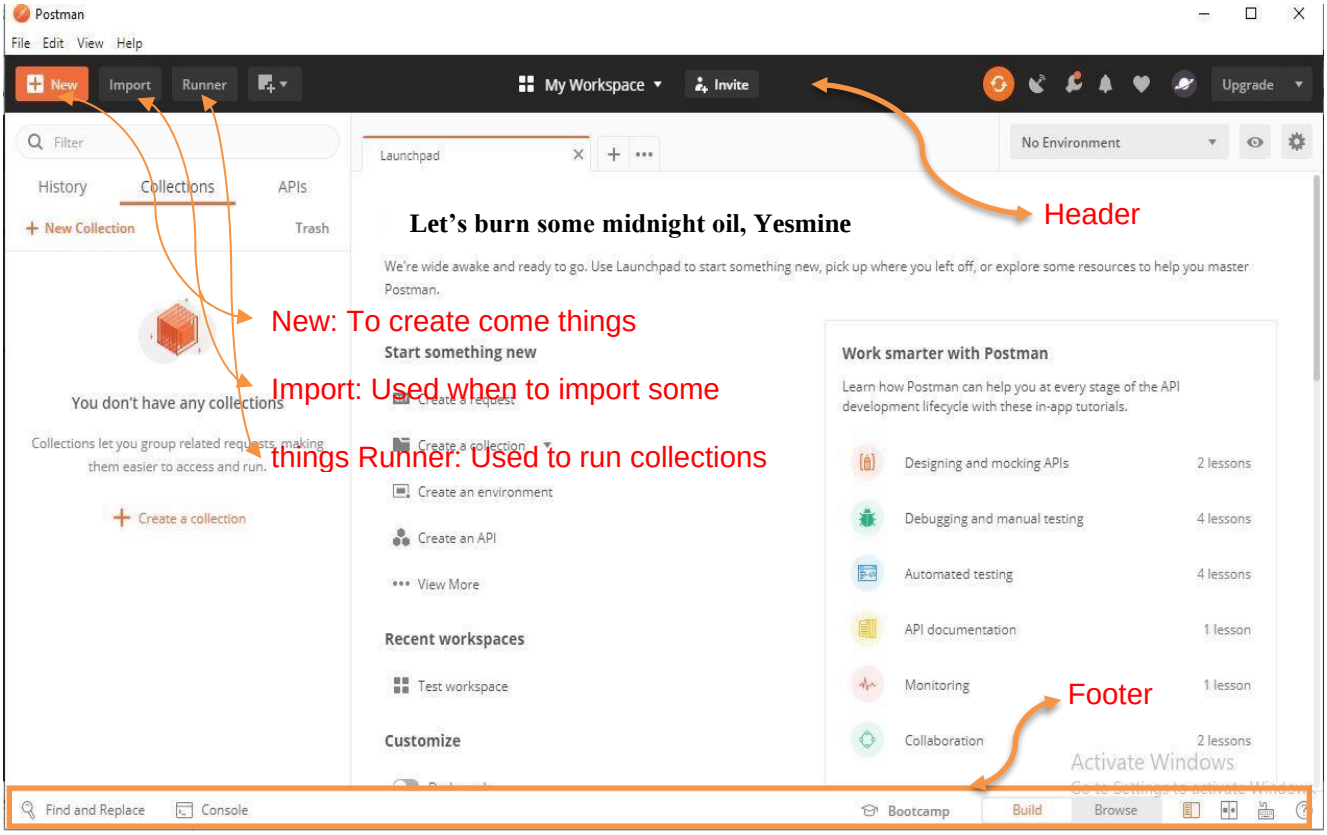
Postman is an API testing tool. We use it to check that if the API perfectly works or perfectly connect to server.

What is API?

Application **P**rogramming **I**nterface (**API**) works as middleman between server and website or application. An **API** is a set of programming code that enables data transmission between one software product and another. It also contains the terms of this data exchange.

API testing process with postman:

When we open postman, this window will come.

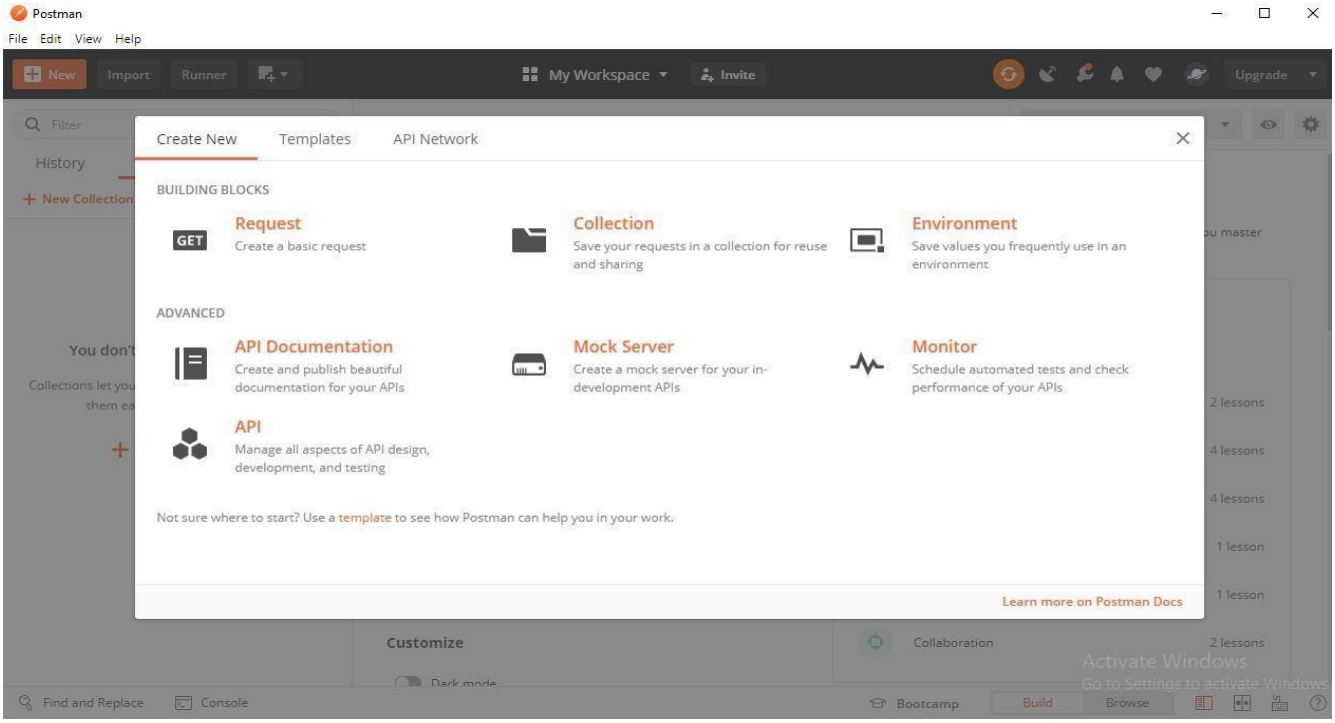


Picture 1: Postman homepage

Footer part has Build and Browse window. (Picture 1) is Browse window. Build window has activity history or summary part of Browse window.

Create a workspace:

By clicking **NEW** we will find this window.



Picture 2: Create new workspace

Request:

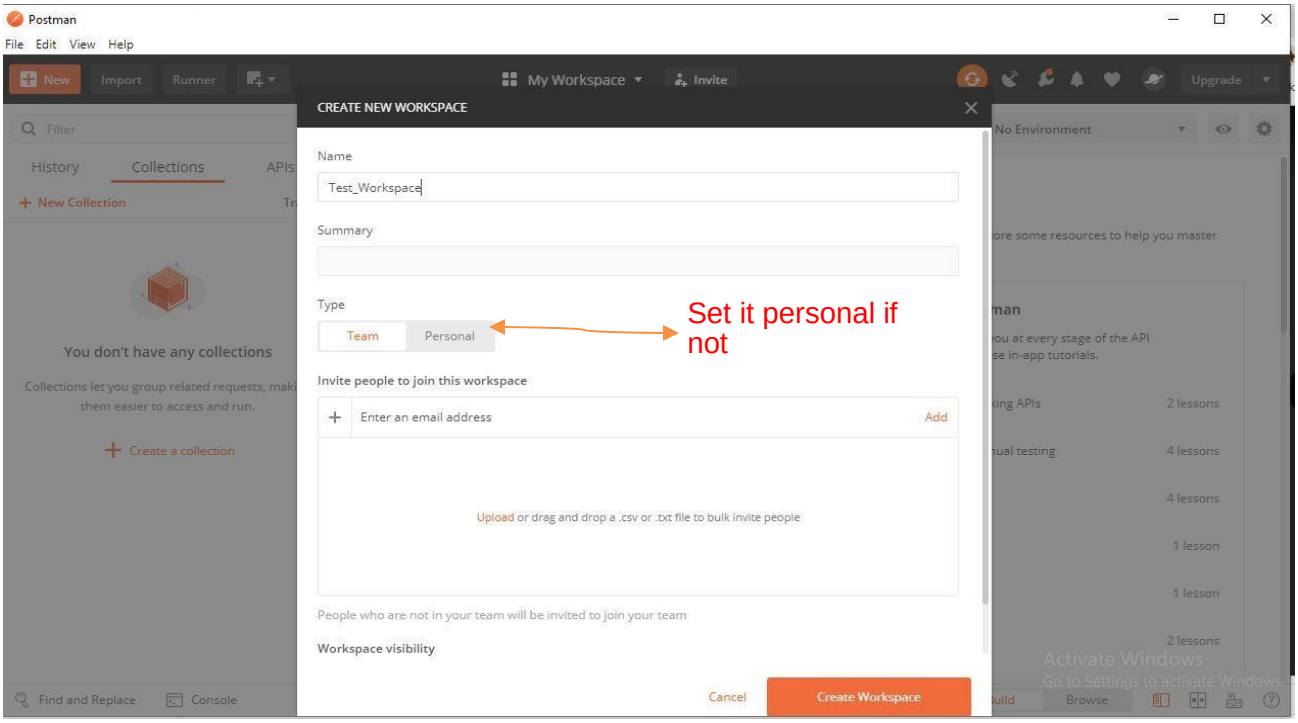
We can create request to run/check API.

Collection:

We need to save APIs in collection. Collection can contain many APIs. We have to maintain proper naming of the collections. This will help to manage the workflow properly and any other team member can get it easily.

Create a Workspace on Postman:

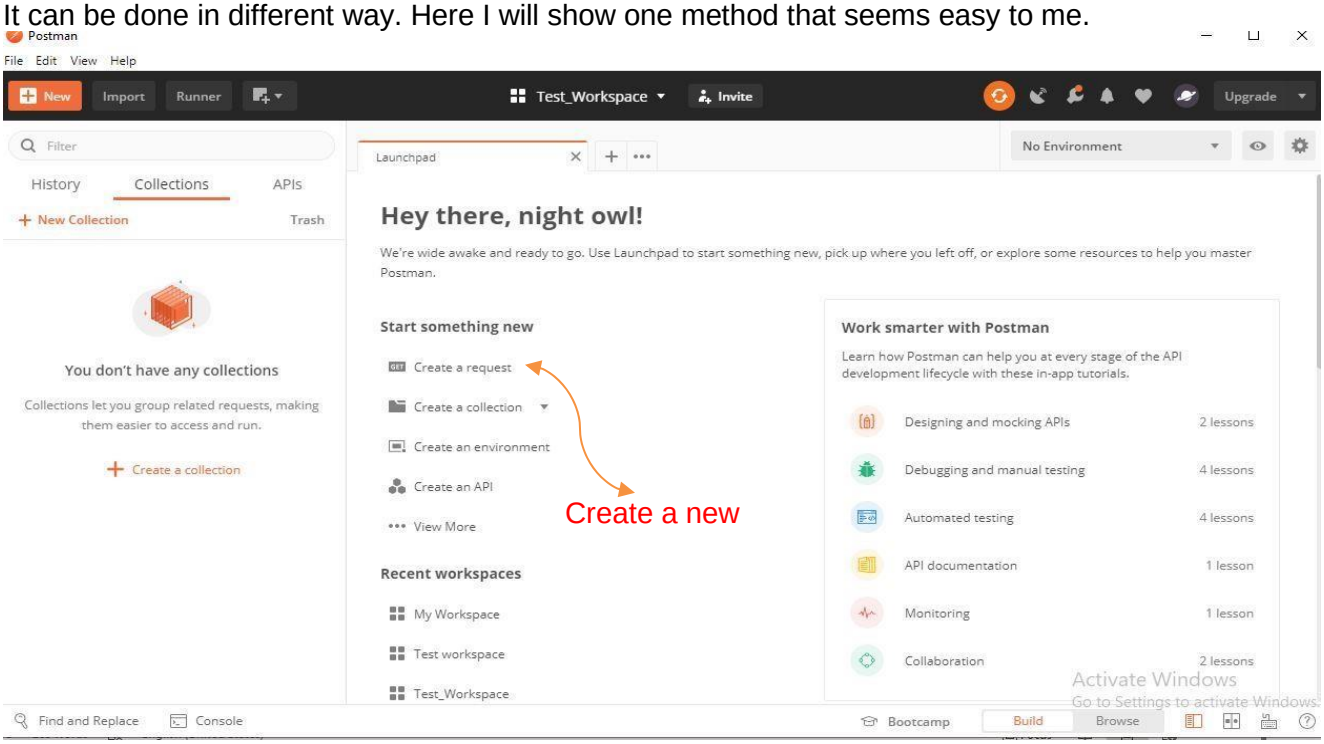
We have to create a workspace to test an API. So from header part we'll get **My Workspace.** Then create a new workspace and give a suitable name of it. (Picture 3)



Picture 3: Create workspace

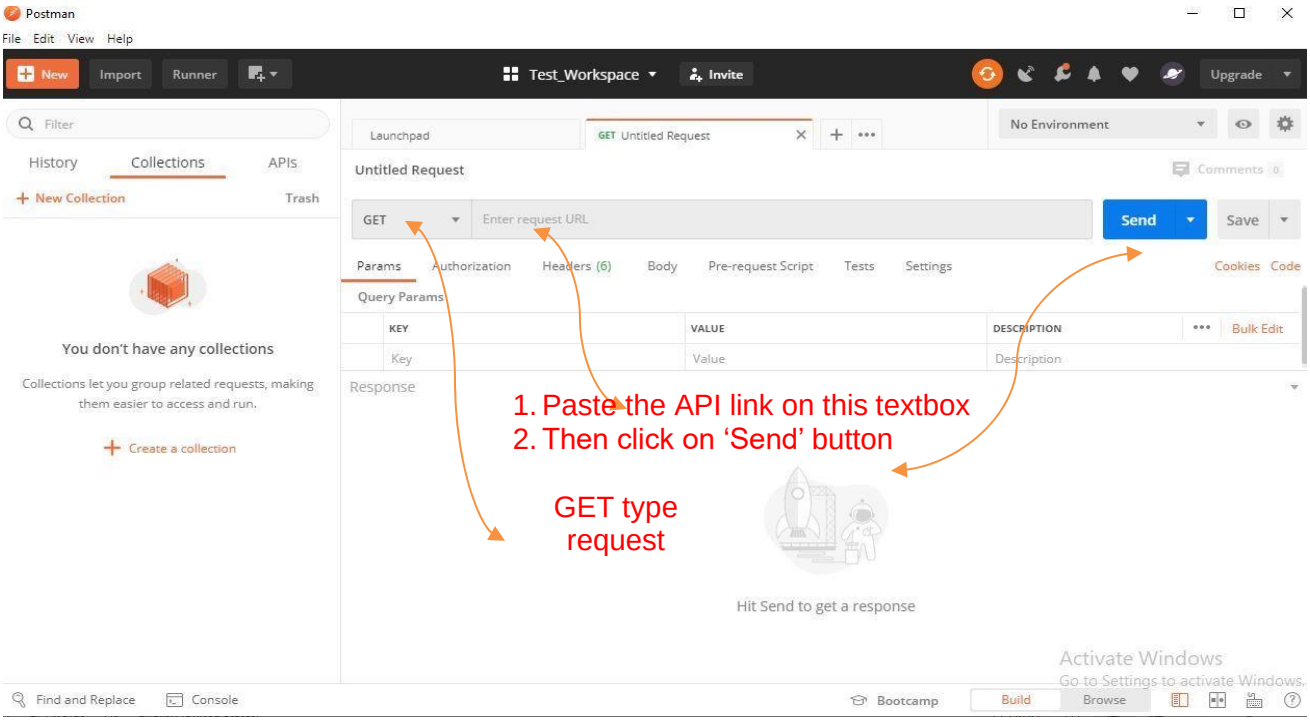
Create a new Request:

It can be done in different way. Here I will show one method that seems easy to me.



Picture 4: Create a request

Then we ‘ll find this window. Here we need to put our API link by the developer team.



Picture 5: Send Request

So, mainly developer team send the API link. But here we use some demo API from a website and check this whether it works or not.

Website link: https://reqres.in/

Here they provide many types of API likes GET, PUT, POST, DELETE, CREATE etc.

Picture 6: API link from website

Some most used APIs:

GET: This request is used to **get** a resource from a server.

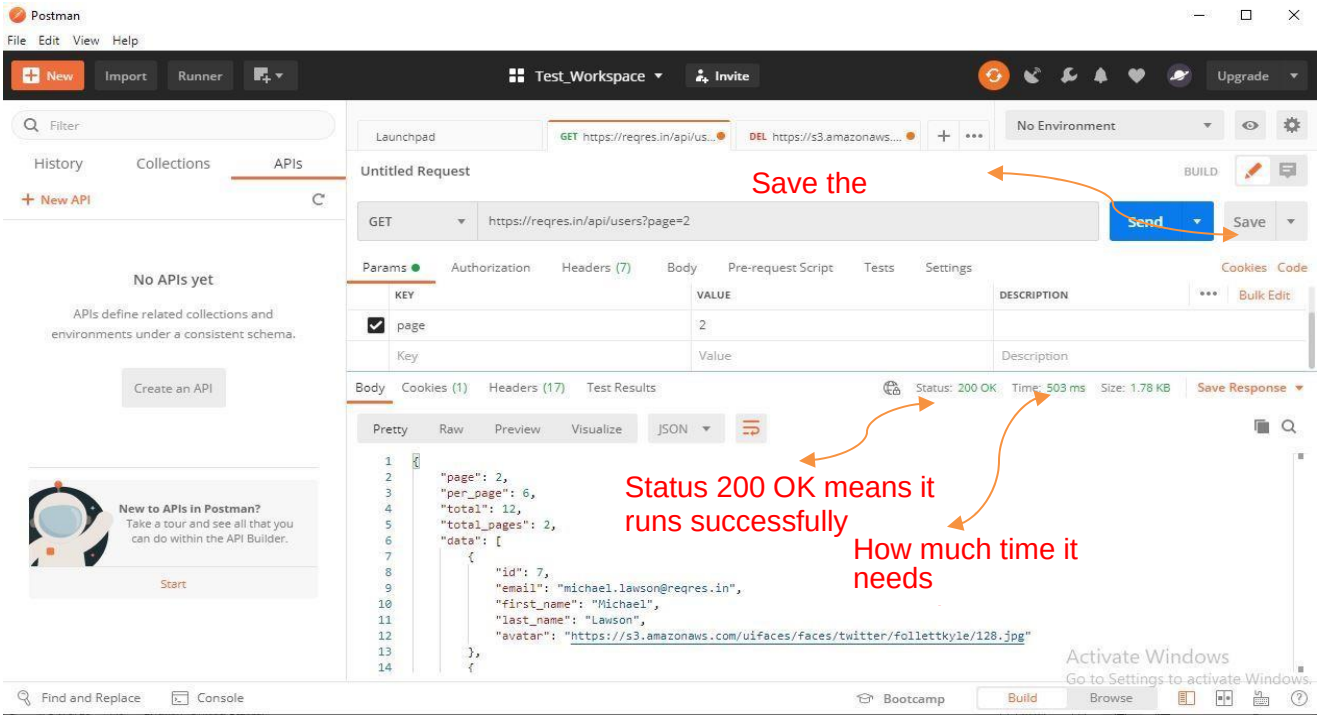
PUT: Use PUT when you want to modify a singular resource which is already a part of resources collection. PUT replaces the resource in its entirety.

POST: POST requests are used to send data to the **API** server to create or update a resource.

DELETE: The DELETE method is exactly as it sounds **delete the resource at the**

specified URL. **PATCH:** Used to modify the end point information.

After sending request we can see the result bellow.

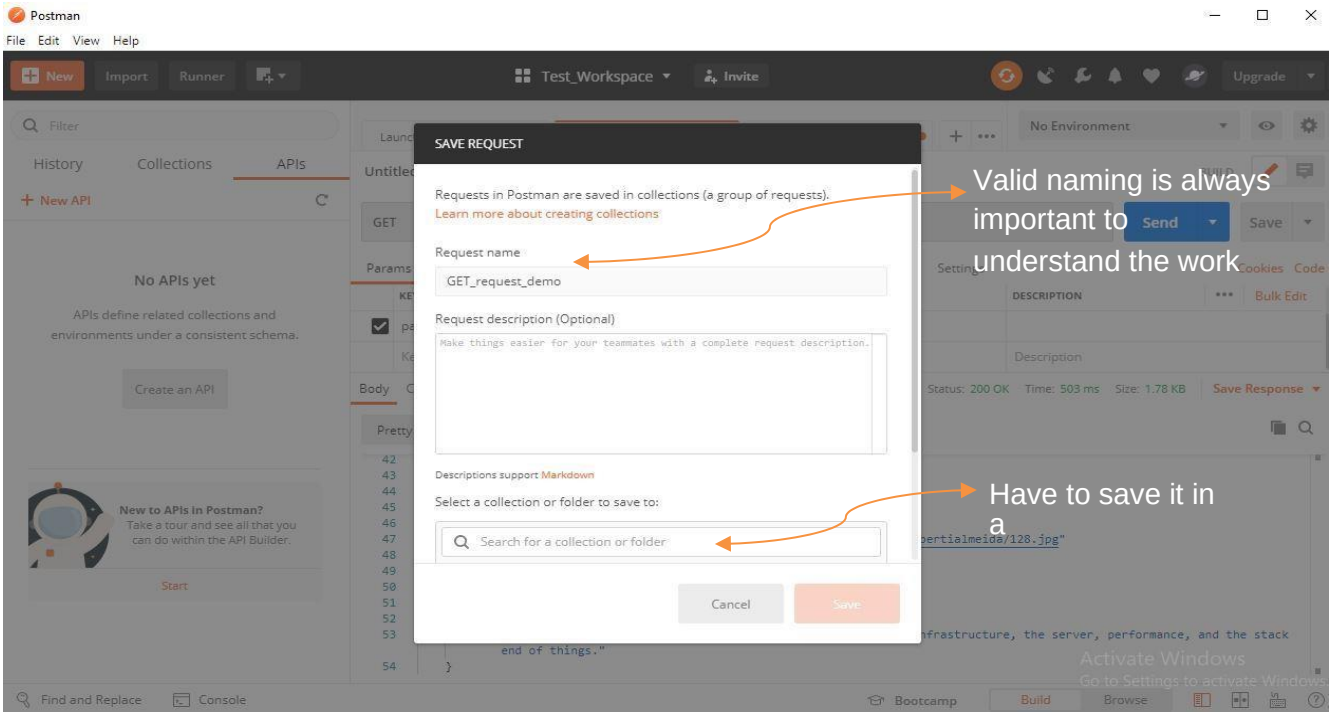


Picture 6: Server response with this API

So, we can save this response and see the result of the API in proper way to show as report.

We have to save this request in a **collection**. And we already discuss about the necessity of collection. Main reason is to **maintain same type APIs in proper way**.

This picture (Picture 7) will show you the window of save a request with collection. We can also create collection before saving the request.

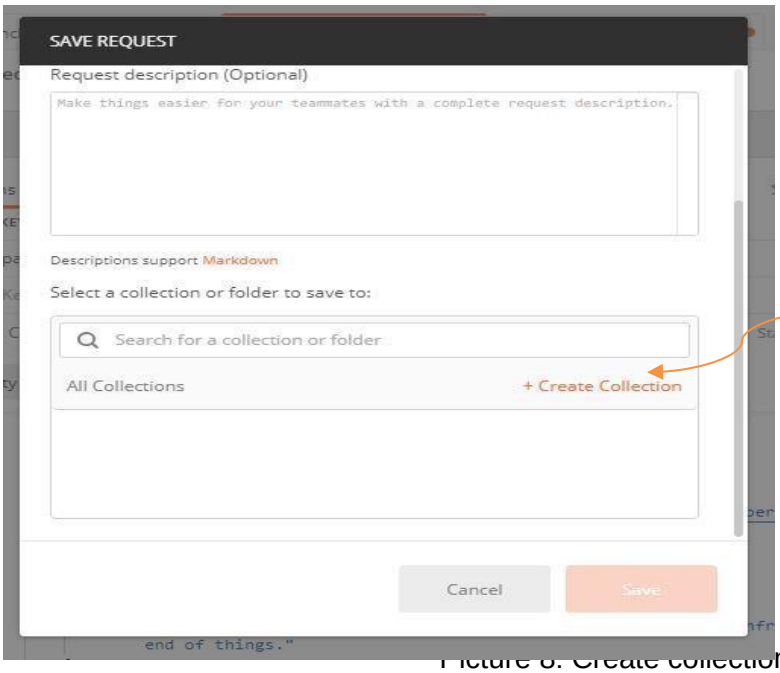


Picture 7: Save request in a collection

Collection Create:

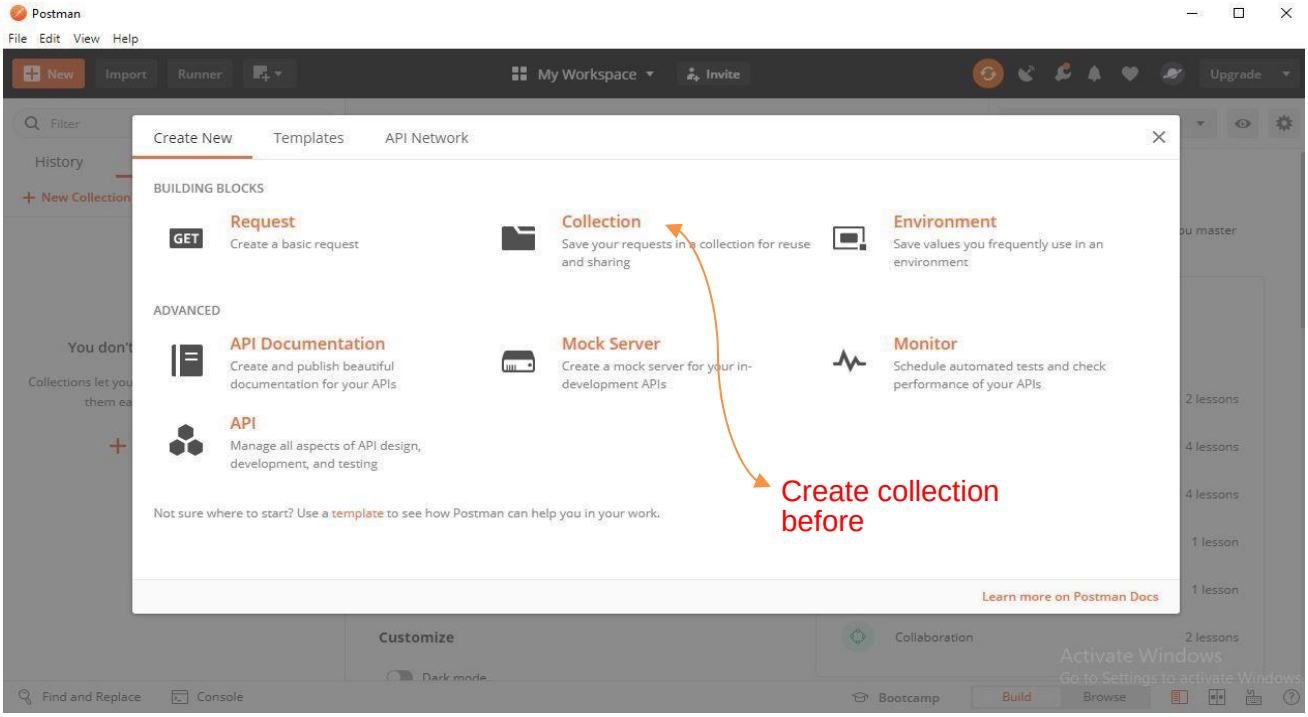
We have 3 different way to create a collection.

1. Create it when save a request.



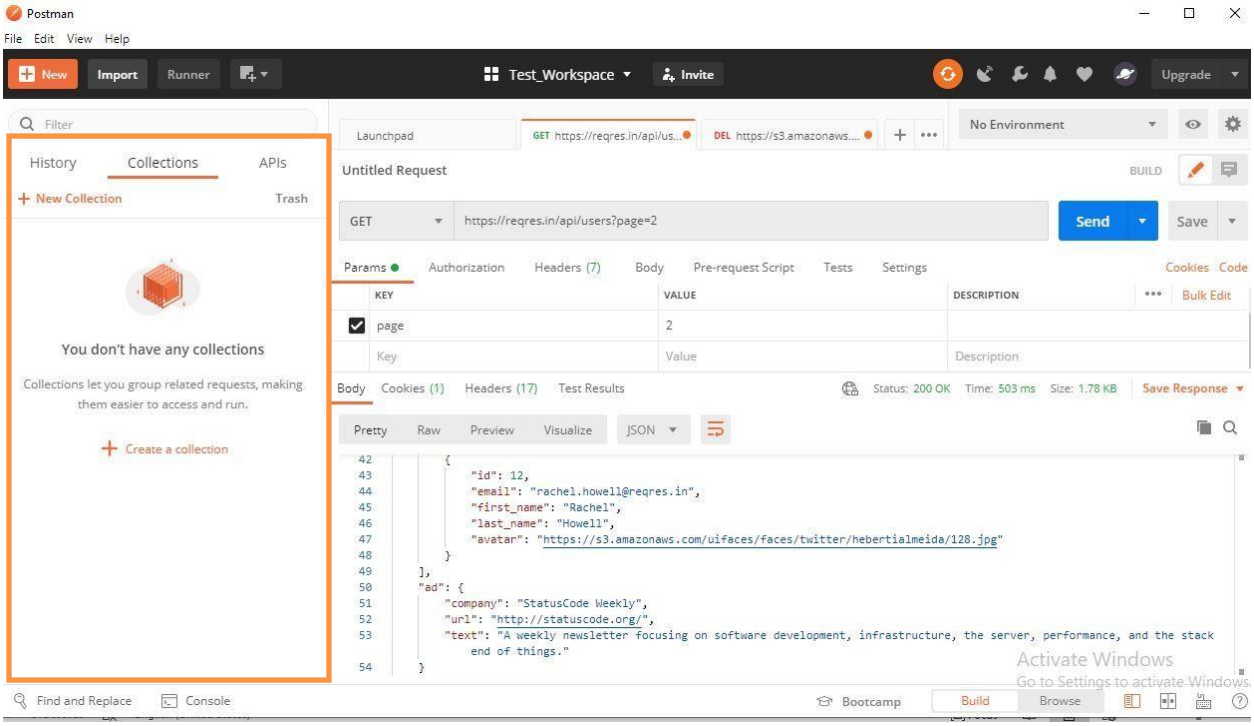
Picture 8: Create collection 1

2. Create collection from NEW button.



Picture 9: Create collection 2

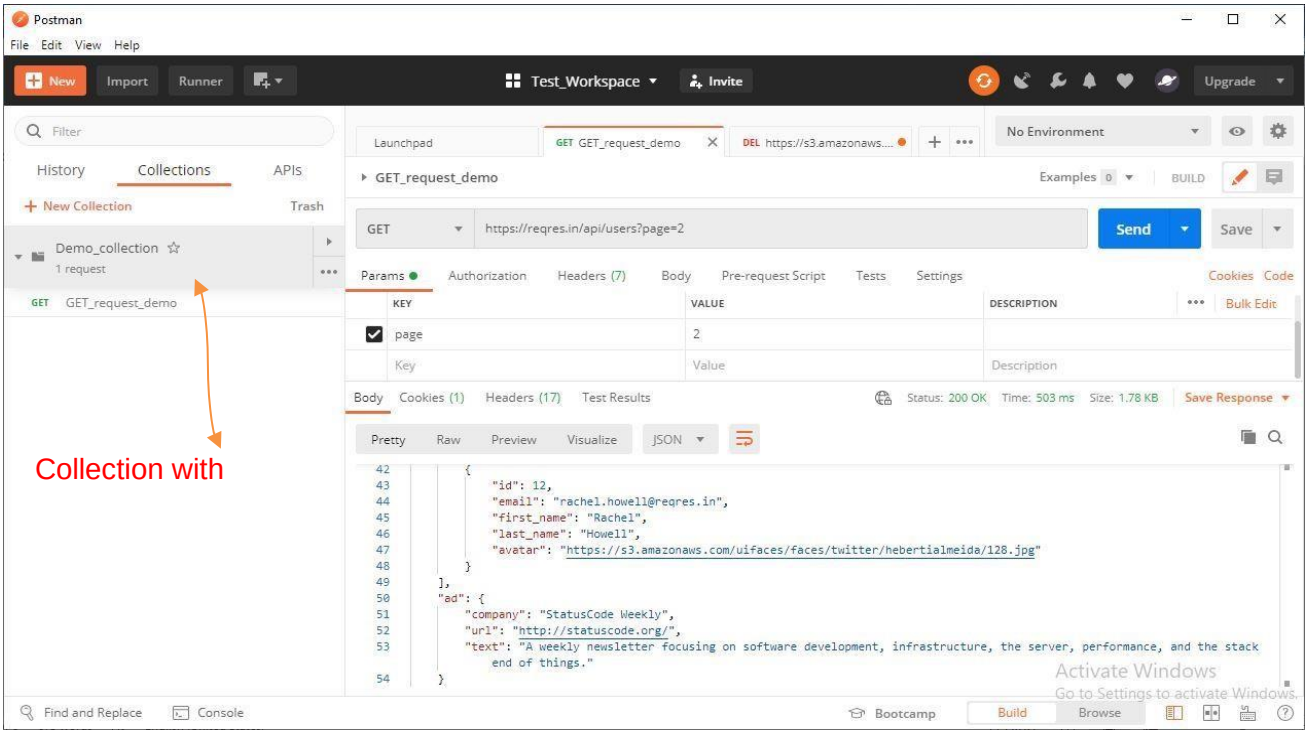
3. Create collection from collection tab.



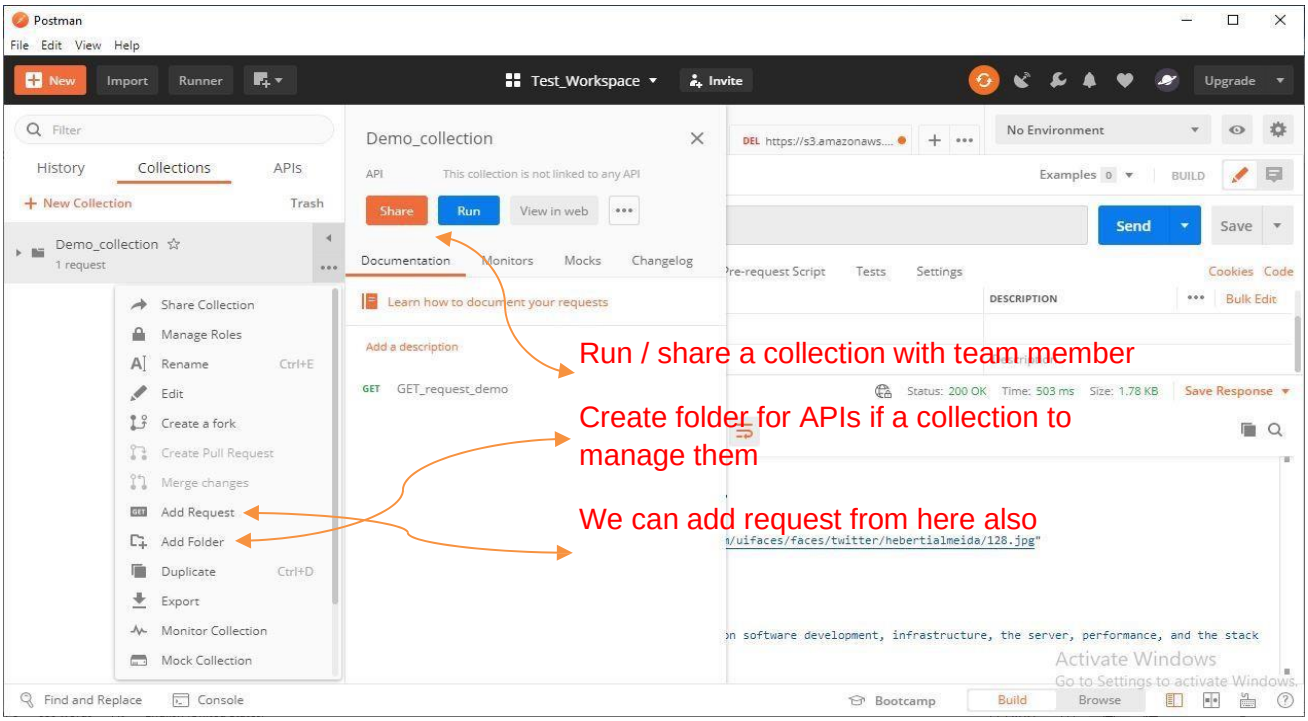
Picture 10: Create collection

Here I create it by 1st way and set name as Demo_collection.

Our request name is GET_request_demo and our collection name are Demo_collection.

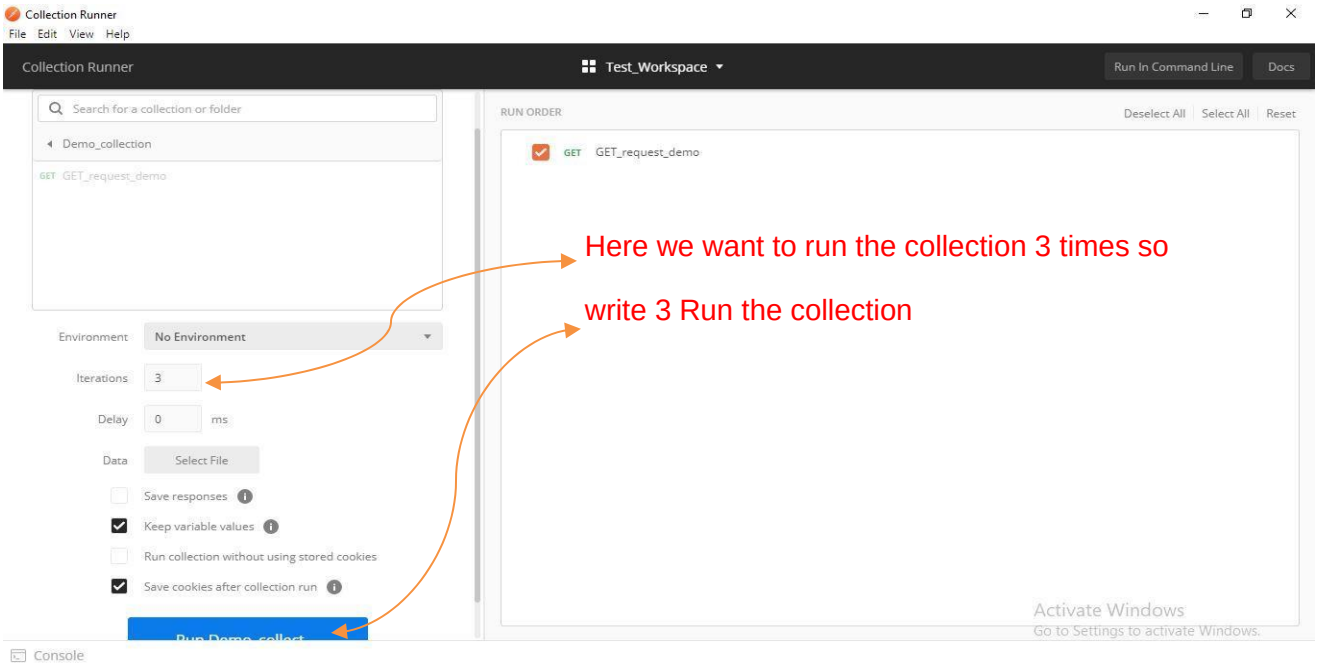


Picture 11: Save collection



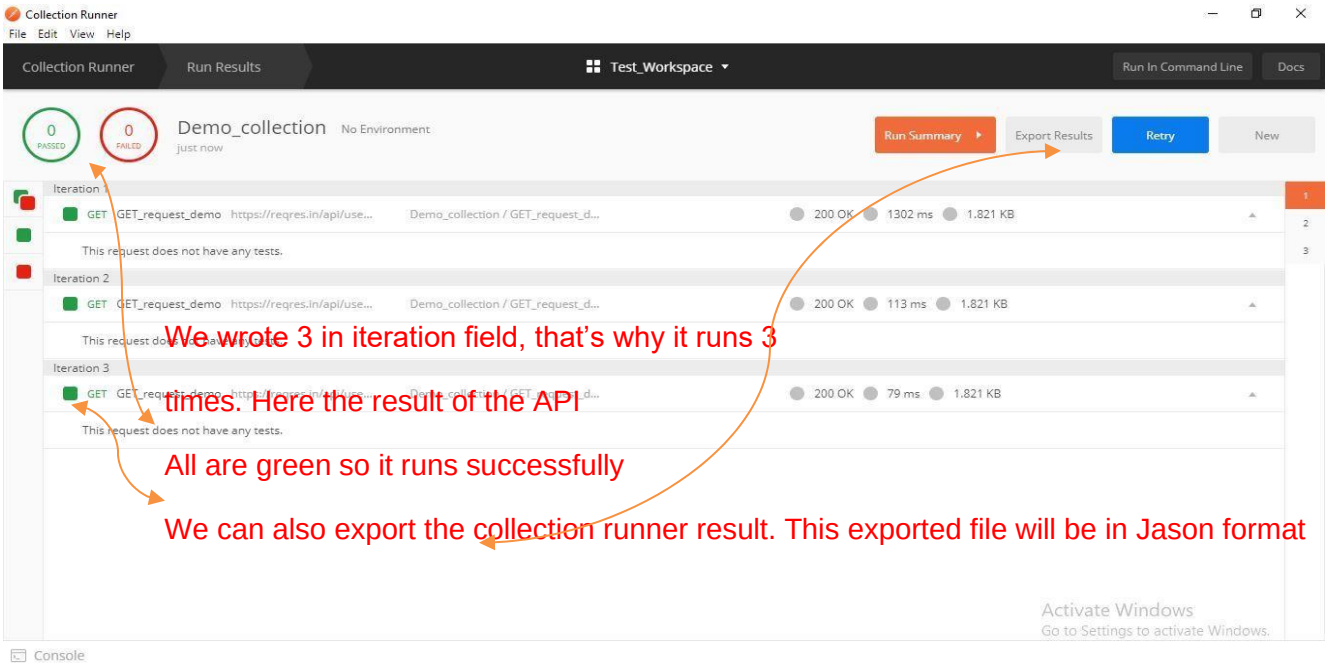
Picture 12: Some basic options with collection

When we click on **Run** then a new window will open and show some options to run a collection.



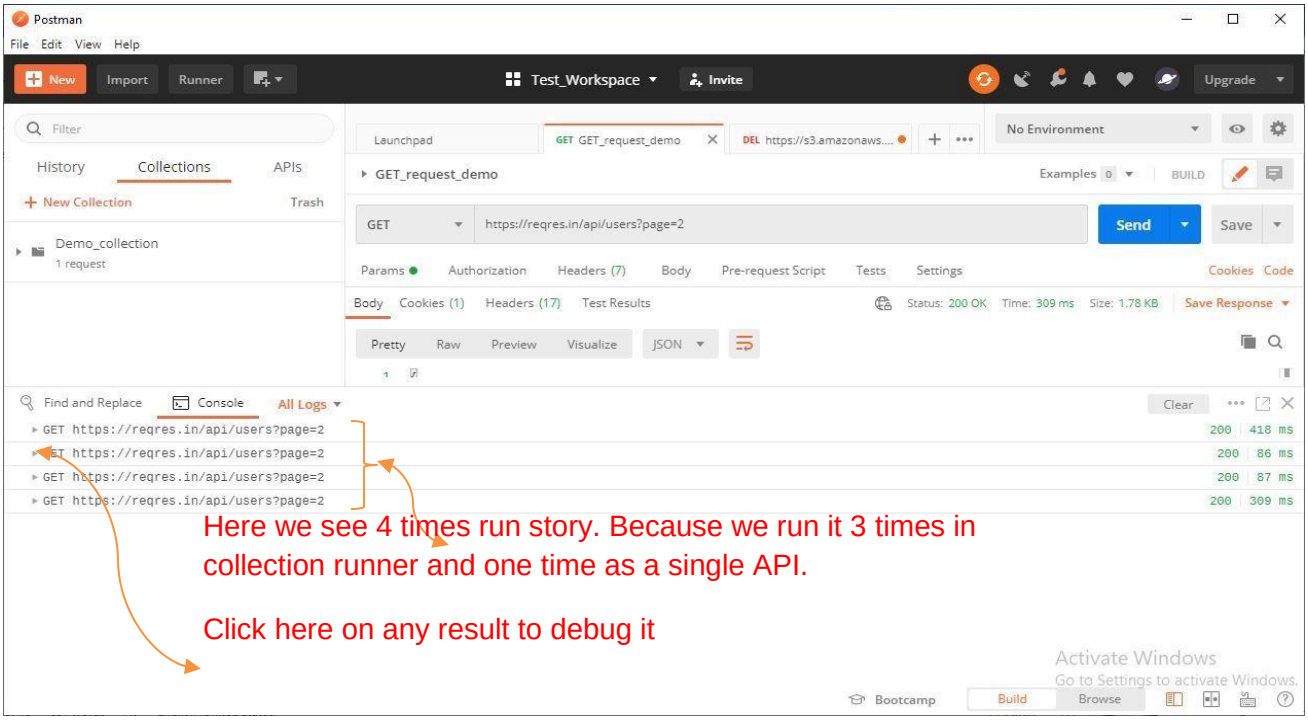
Picture 13: Collection Runner

Results that we can see below.

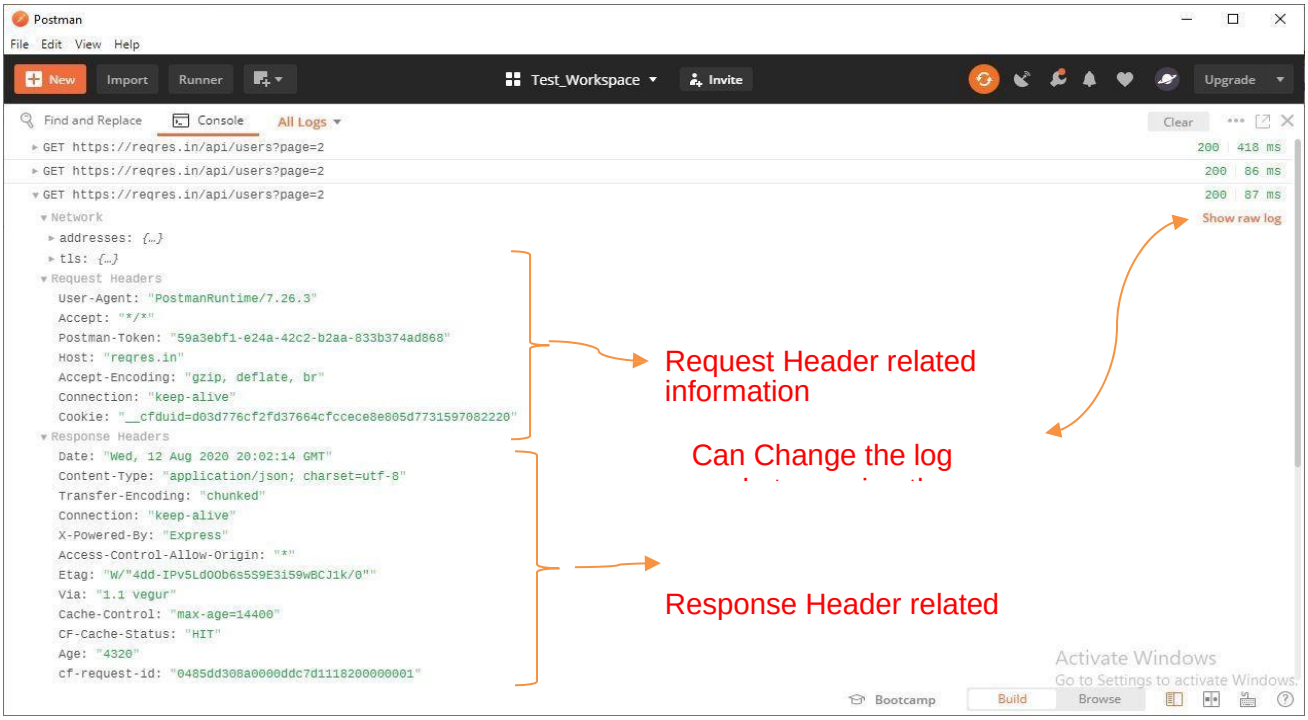


Picture 14: Collection Runner result

Next part is debugging. Frist of all we have to take a look on footer panel of postman. There we will find
Console. So we click on the console.



Picture 15: Console Panel

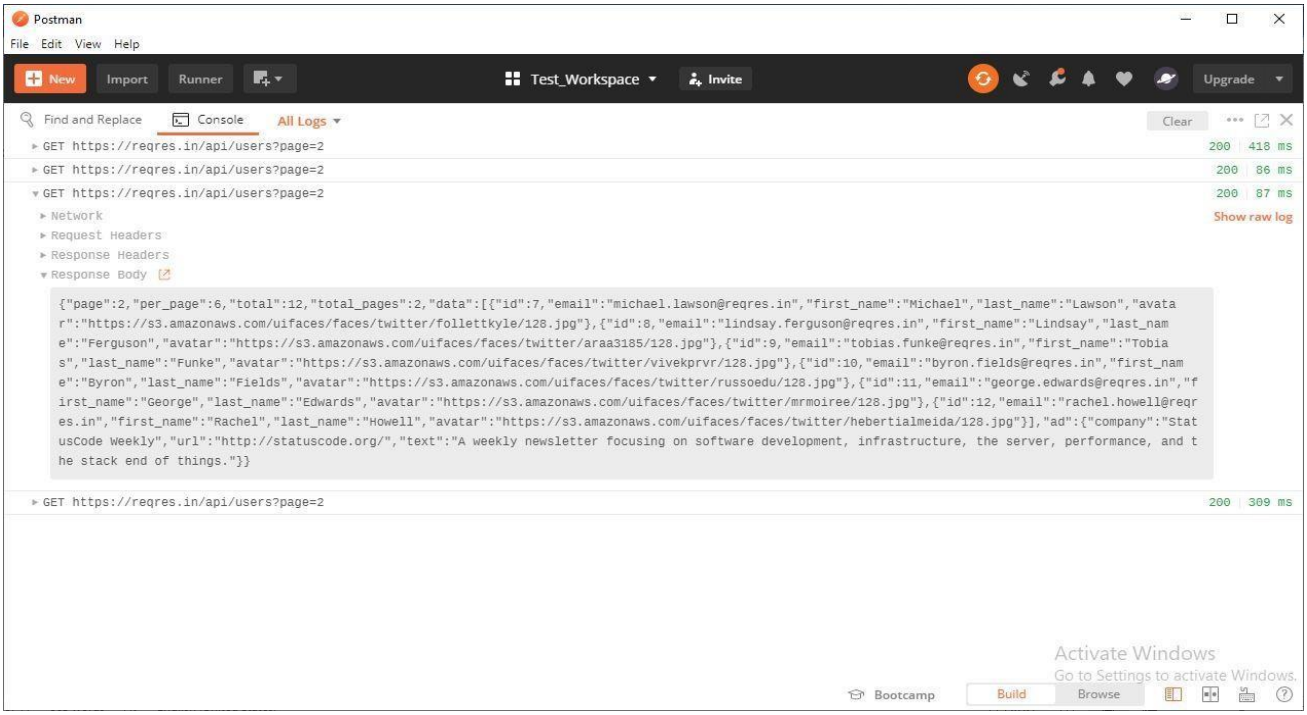


Picture 16: Request and response header

Debugging:

We need to justify that, does our response is like our

request. So, lets check the response body.

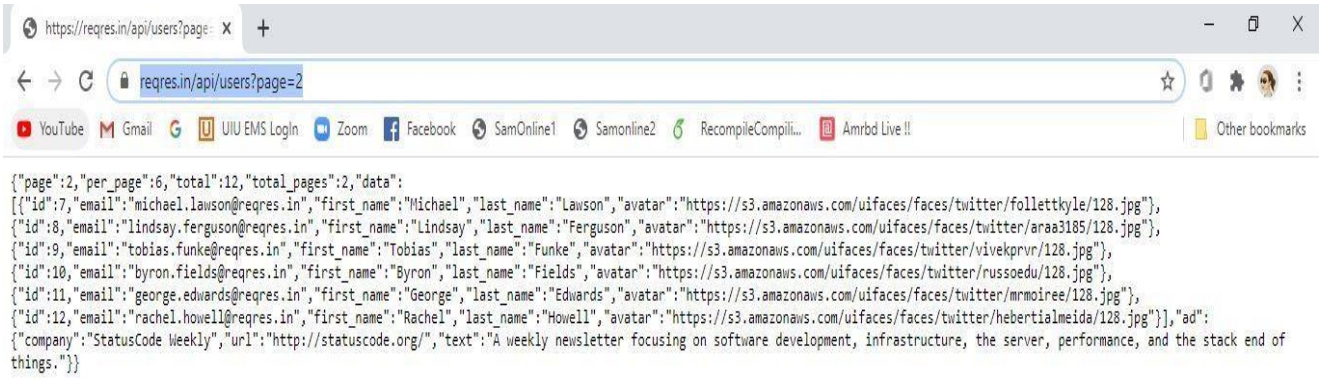


Picture 17: Response

Body Now we need to copy our API and paste on any

browser.

Our API is: https://reqres.in/api/users?page=2



Picture 18: API debugging

Both are same. So, we can say the API works successfully.

Some important information of API running status:

Here we get **200 Ok.** We need to know other status also. Following is some other status.

- **1xx: Informational** – Communicates transfer protocol-level information.
- **2xx: Success** – Indicates that the client's request was accepted successfully.
- **3xx: Redirection** – Indicates that the client must take some additional action to complete their request.
- **4xx: Client Error** – This category of error status codes points the finger at clients.
- **5xx: Server Error** – The server takes responsibility for these error status codes.

Finally, we can say this is very basic level concept on Postman. We will learn deeply in future with proper providing API link by developer team.
