

# Postman

Theory

Practice

100% completed, 0 problems solved ▾

## Theory

⌚ 8 minutes reading

Unskip this topic

Start practicing

If you already know something about REST services but haven't quite used them yet, it's time to try! REST requests are simple HTTP requests, but it's not convenient to make them through a browser as you'd usually do with most of the web content. You need a client to create requests, and [Postman](#) is a good one to try out.

We will use endpoints of the free API server [Reqres](#) to make our example requests. It has the implementation of all basic HTTP methods: *GET*, *POST*, *PUT*, *DELETE*.

### 2 required topics

- [JSON](#) ▾
- [REST](#) ▾

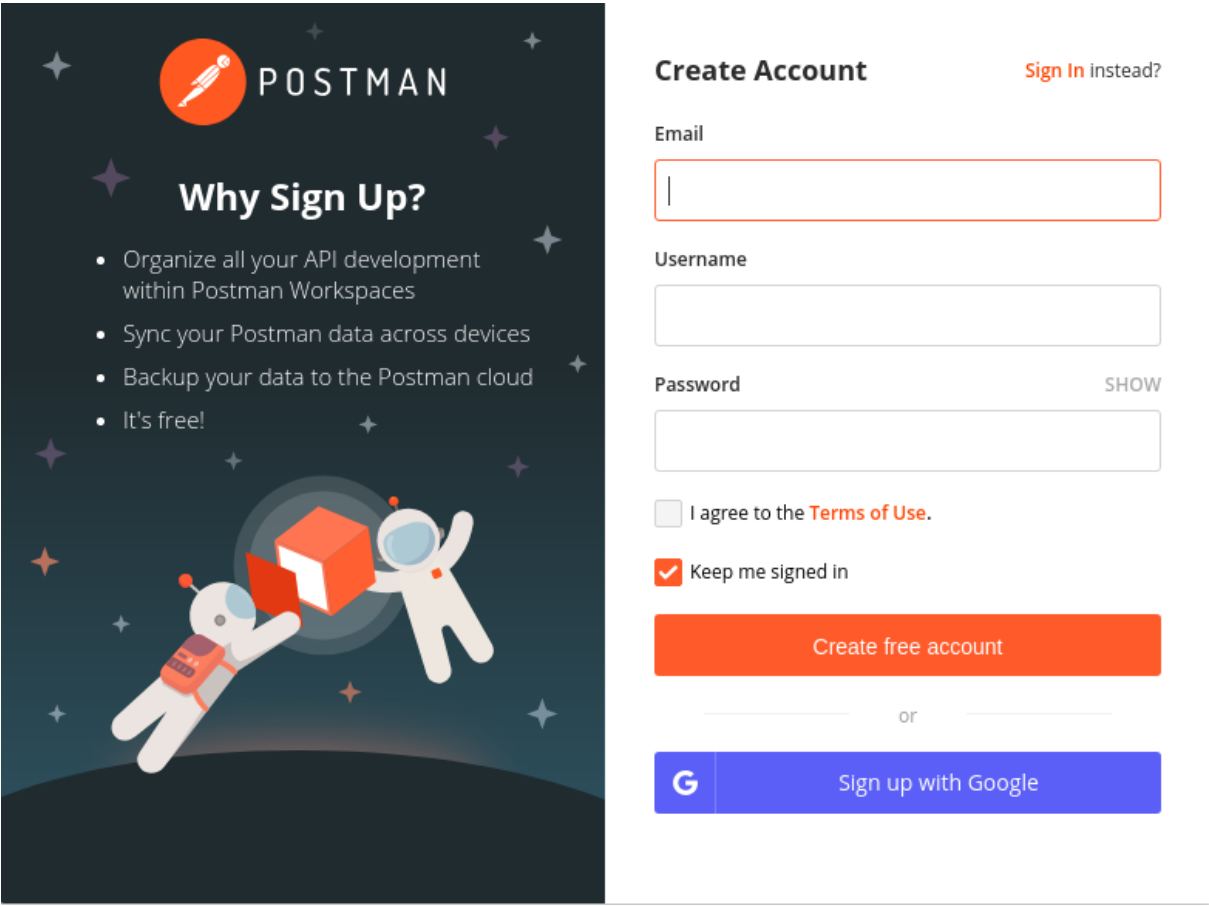
### 3 dependent topics

- [Getting data from REST](#) ▾
- [Requests handling](#) ▾
- [Exception handling](#) ▾

## §1. Installation

**Postman** is a collaboration platform for API development. It has more functions than a simple API client, but all we need now is its ability to create requests to API endpoints.

You can get the application on Postman's [official website](#). Once you download the executable file for Windows or an archive for Linux, unpack it if needed and run the app. You should see a page like this:



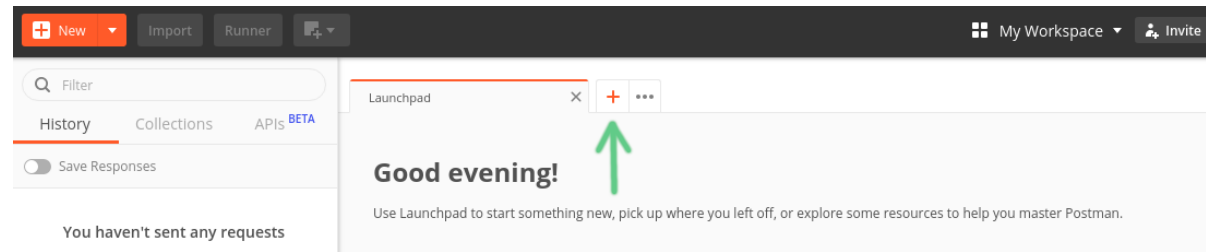
Using Postman is free, so you can use your email or Google account to sign in. This will help you share your workspace on several devices. The registration is not obligatory, so you can close this window for now and start using Postman anonymously.

We use Postman 7.15.0 for our illustrations here. The UI interface of your version may look slightly different, but the behavior of the application should be the same.

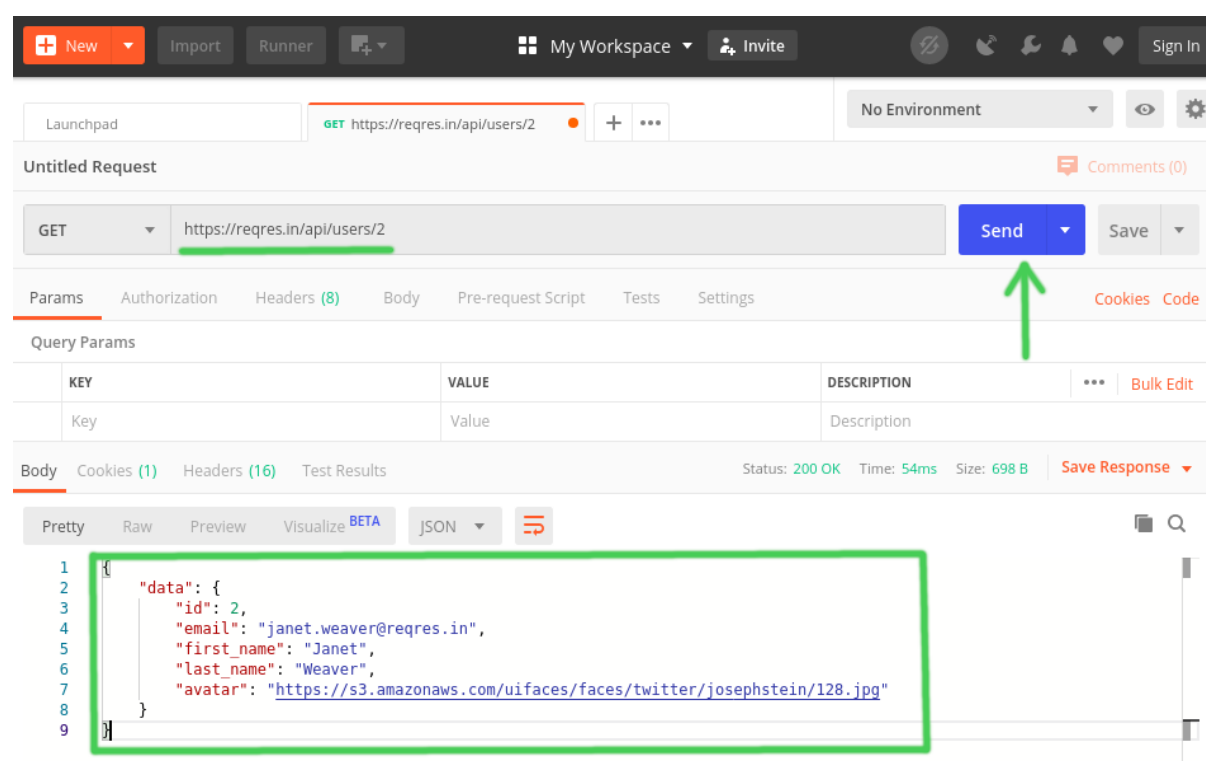
## §2. GET requests

We are almost ready to make the first request. For this example, we will use a free test API server. Reqres API has several endpoints; you can get their descriptions on the [official site](#).

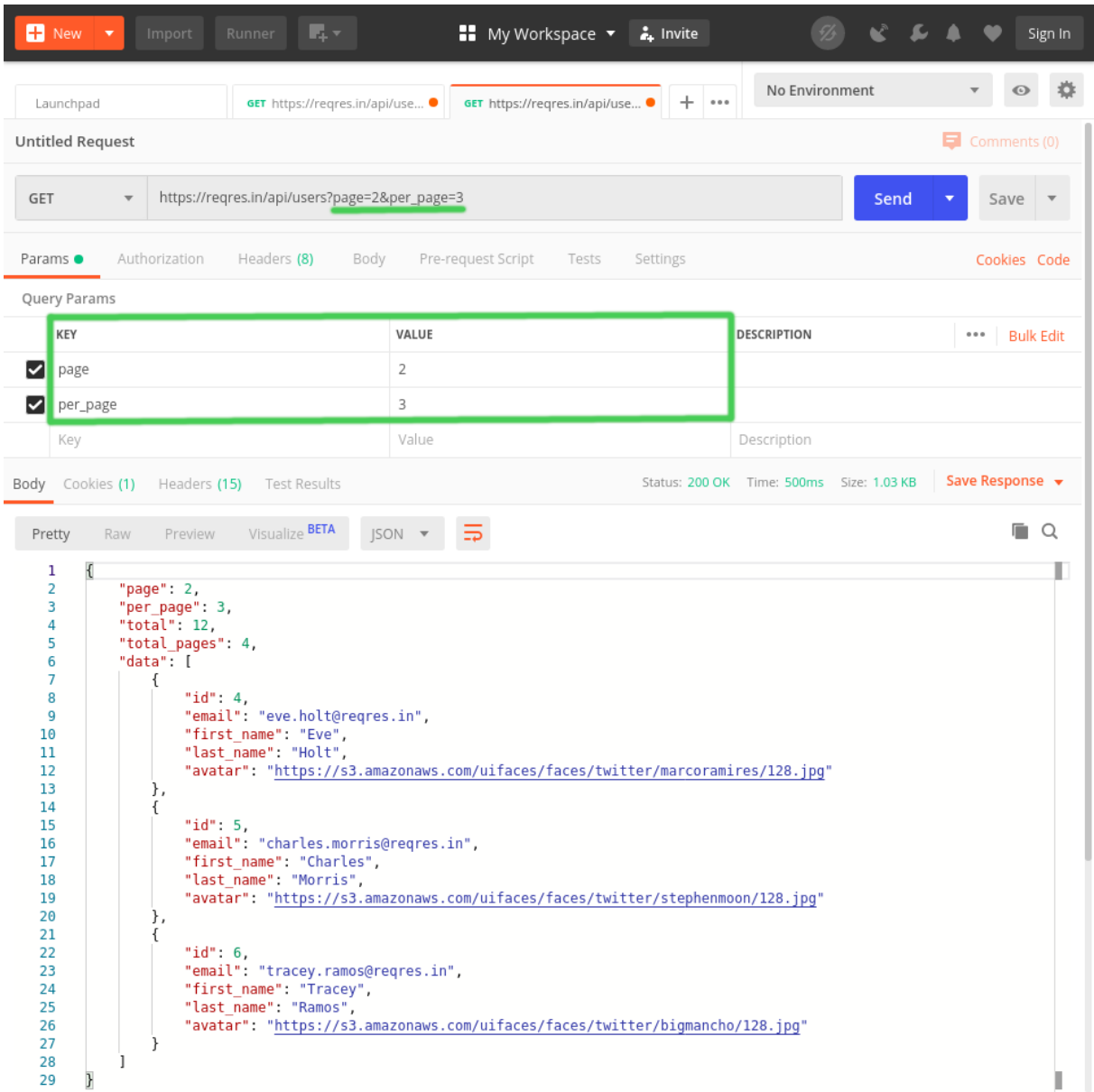
Postman is similar to a browser: you have tabs for your requests. Let's open the first one. Just hit the *plus* tab, and Postman will show the full panel.



To create a simple *GET* request without query parameters, add the address of an endpoint of the service. In the image you can see that we use <https://reqres.in/api/users/2> for this purpose. Then press the *Send* button, and you will see a nicely formatted response.



If you want to use any query parameters, you can add them as key-value pairs. Notice that we used endpoint <https://reqres.in/api/users> this time.



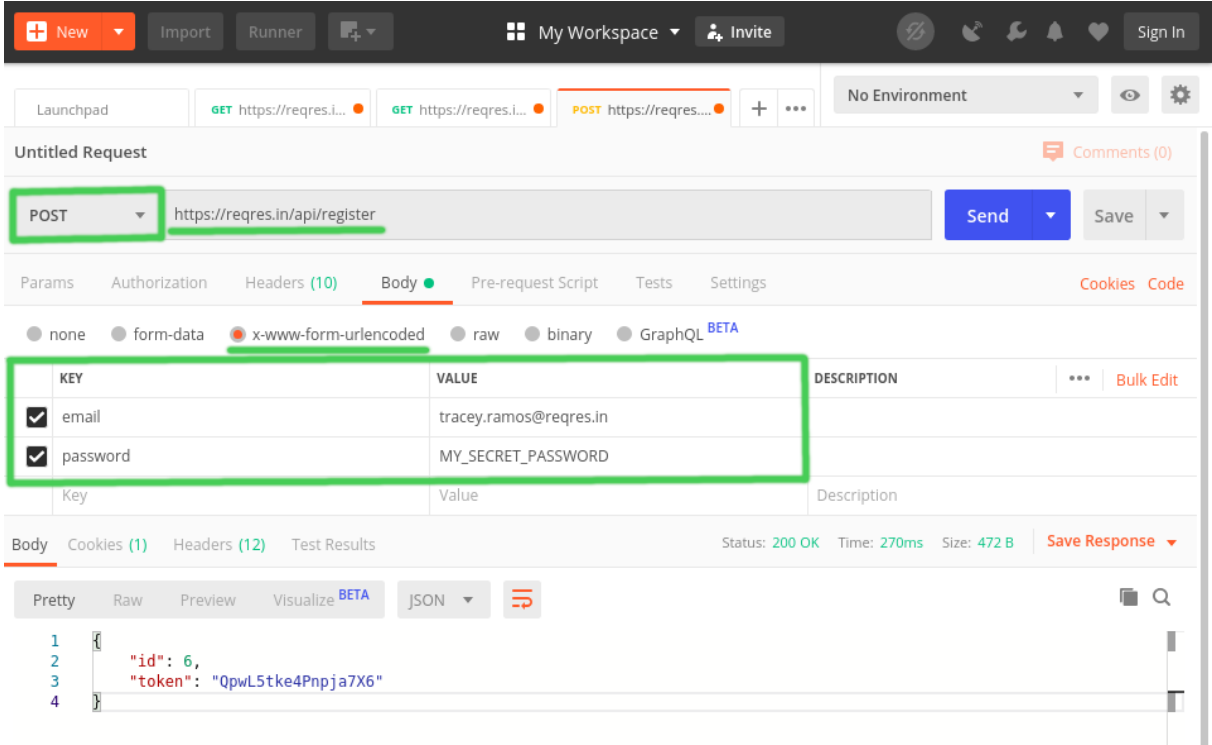
Query parameters and key-value pairs are interchangeable in the application, so you can use any method you want. Postman will fill the other one automatically.

We know how to get data from the server, but sometimes we need to send data with *POST* requests, so let's see how you can do it with Postman.

### §3. POST requests

A useful feature of *POST* requests is that we can fill the body of it with sensitive data. If we send our login and password through query parameters, it's easy to read them for someone who sees the traffic from your computer, for example, your internet service provider. We will try our best with registration and authentication to the server through API requests, but do not use real data for it though.

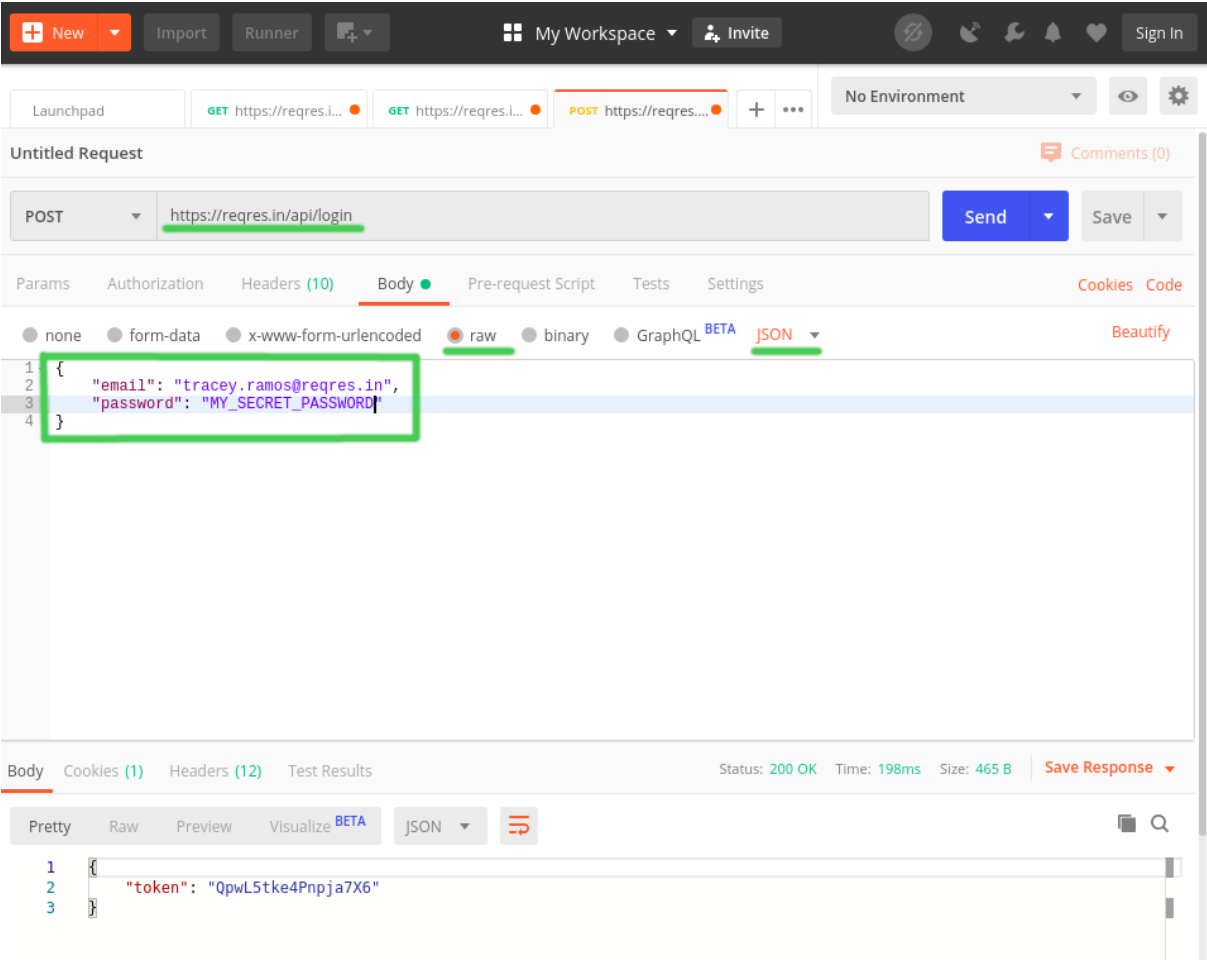
To create a *POST* request, open a new tab as you did earlier. Reqres allows you to use only defined emails that you receive in the previous example at the endpoint `https://reqres.in/api/register`. Let's choose "tracey.ramos@reqres.in" as the user's login for the registration. Change the request type to *POST*, add key-value pairs with email and password (any password you want) in the body of the request, and press *Send*:



We receive a response with a token. Tokens are identifiers that you can use for authorization. We almost repeated our previous requests, but this time choosing different parameters.

Sometimes it's preferable to use JSON to send data to the server, for example, when we have nested fields. Though we don't need to make a nested structure to send a login request, we try to make a JSON to show out how it works.

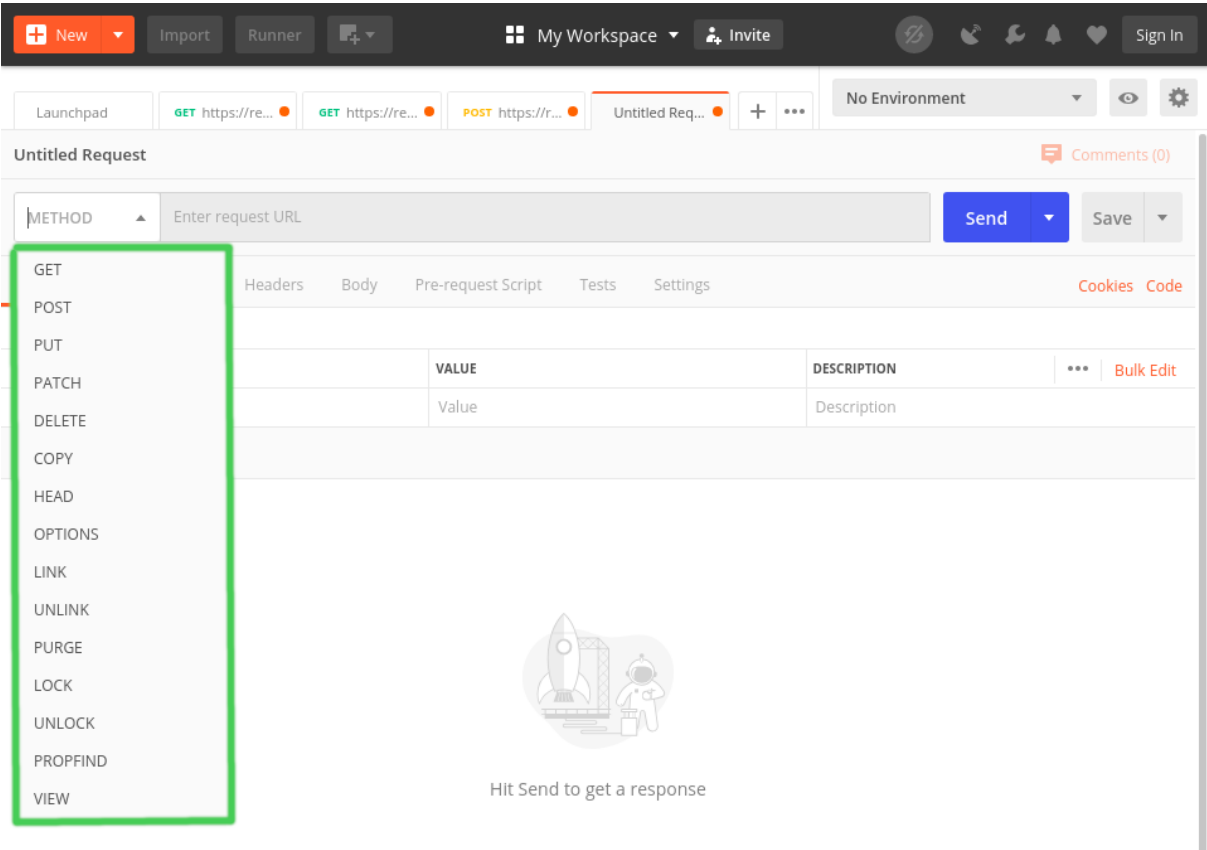
Choose a raw format for a body and JSON type for the data type. Then paste JSON with email and password to the editor and send the request to the server:



And again we succeed and get a token.

## §4. Other requests

Postman allows you to create HTTP requests with other methods: *DELETE*, *PUT*, *PATCH*... You should select the one you want as the request type:



You know enough to fill the fields by yourself and practice *DELETE*, *PUT*, *PATCH* with the [Reqres API](#) server.

Before using any API, do not forget to read the documentation first, or you can make some inappropriate changes on the server.

### Table of contents:

- [1 Postman](#)
- [§1. Installation](#)
- [§2. GET requests](#)
- [§3. POST requests](#)
- [§4. Other requests](#)
- [§5. Conclusion](#)
- [Discussion](#)

Now you are ready to send a request to any server you want!

§5. Conclusion

Postman is a simple and useful tool for making API requests. It includes all basic requests like POST, GET, PUT, DELETE, etc. You can use it as an API testing tool to reliably check HTTP requests.

 Report a typo

378 users liked this piece of theory. 14 didn't like it. What about you?




Start practicing

Unskip this topic

[Comments \(31\)](#)

[Useful links \(4\)](#)

[Show discussion](#)

 **JetBrains Academy**

Tracks


Pricing

For organizations

About

Contribute

Careers

 Become beta tester

Be the first to see what's new