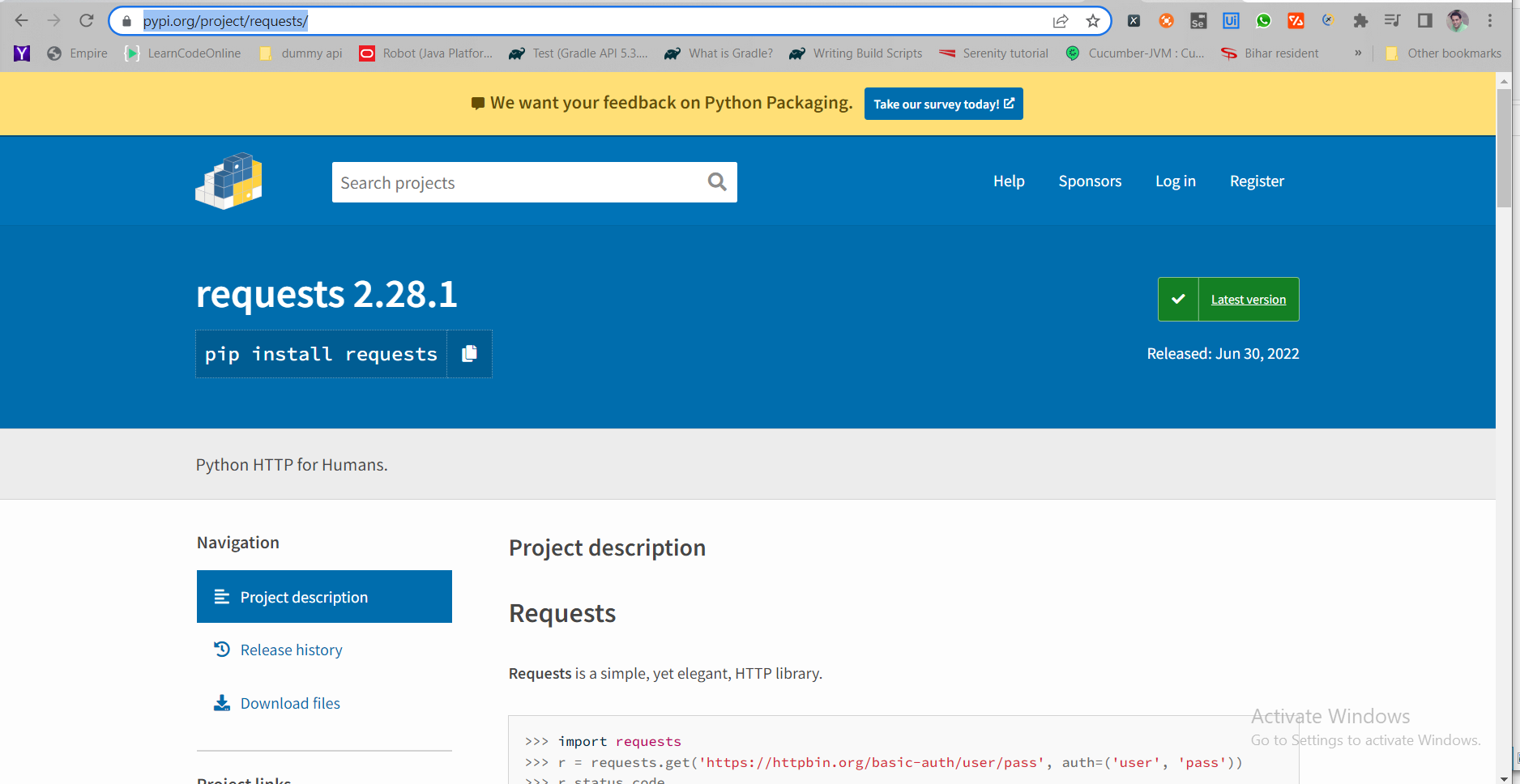
**Python – API Test Tutorial** –

There is a requests package which helps in Python doing the API Testing

Official site of Python – Requests Module: -

<https://pypi.org/project/requests/>



Graphical user interface, text, application

Description automatically generated

Install the requests module – by going into the terminal and typing – pip install requests



**Install the JSON Path**:–

Text

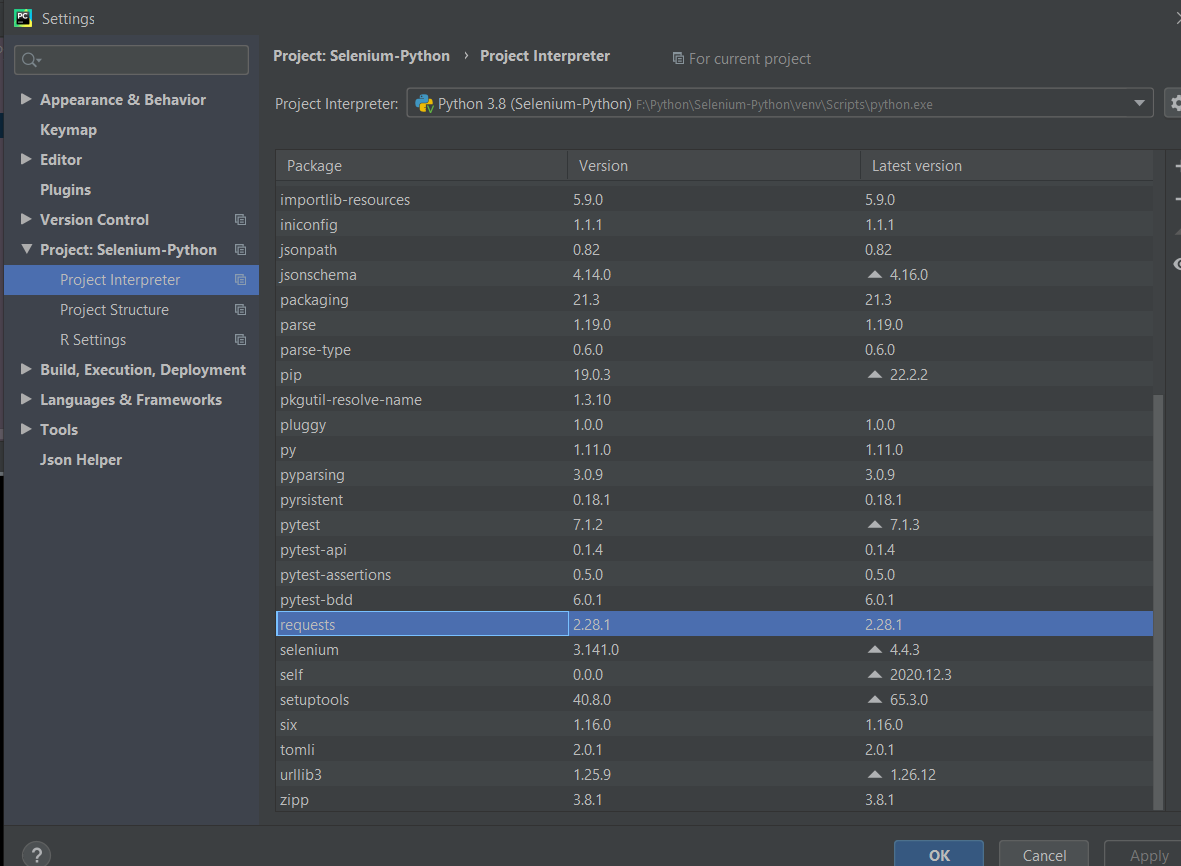
Description automatically generated

**Jsonpath -version (0.82 is Installed)**



Check the **Requests and jsonpath package Installed is visible in the Python Interpreter Settings**:-

**Python - > Settings -> Project Interpreter**



**Dummy API End Points URL for practice**: -

<https://reqres.in/>

Graphical user interface, application, website

Description automatically generated

**Do a simple Get Requests**: -

Logo

Description automatically generated

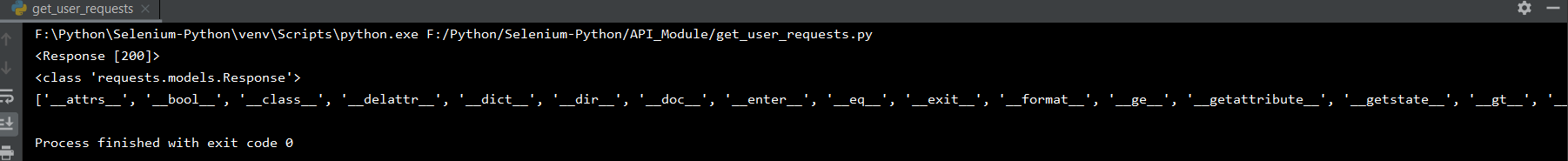
**Print the Response**: -

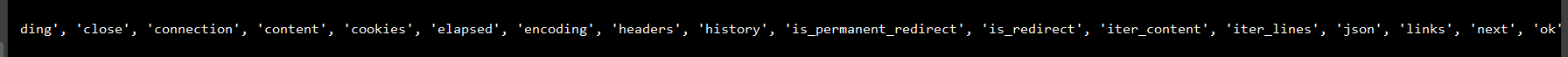
Text

Description automatically generated

To check the Object of the type of response: -

To check the **different methods available with the response object**:-



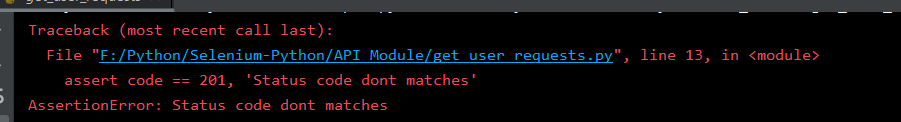


To **add assertion on the status code – to check if the status code is coming correctly or not** :-

Graphical user interface

Description automatically generated

Now if we give 201 – it will throw assertion error

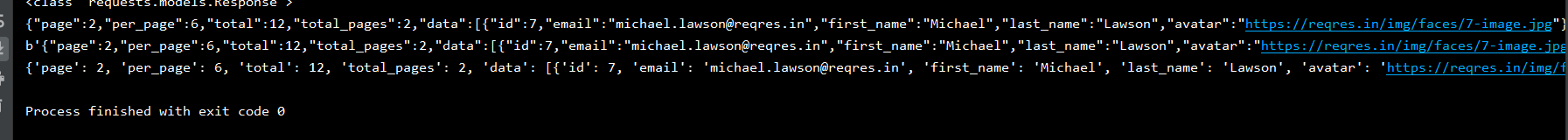


Now try to **fetch the response in text, content – (byte) and json – format**

Text

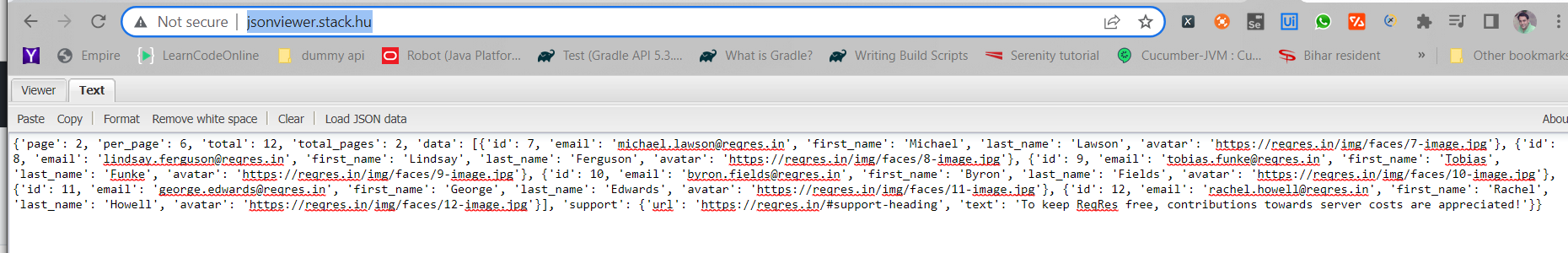
Description automatically generated

Output: -



Go to JSON viewer site to check the JSON Response:-

<http://jsonviewer.stack.hu/>



Graphical user interface, text, application, Word

Description automatically generated

It will give **the proper view of the JSON Data**.

Now let’s try **to get all the headers of the response**: -

Graphical user interface, text

Description automatically generated

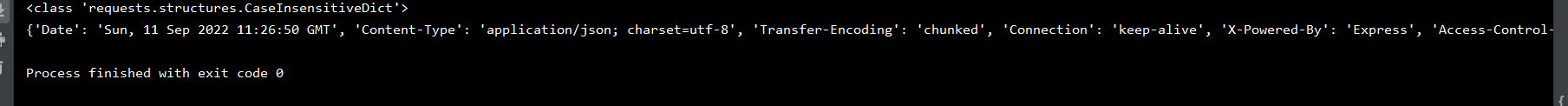
**Output**: -



**Headers – type is Dictionary**: -

Text

Description automatically generated



Now to get **the final url – which we are hitting, to get the response encoding , and the response cookies :-**

**A picture containing text

Description automatically generated**

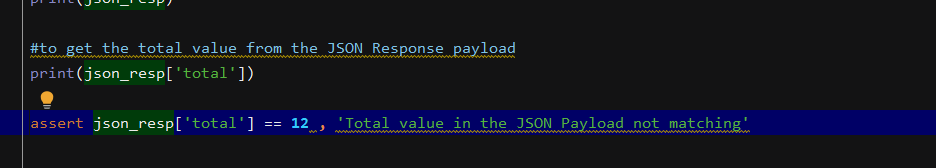
**Output :-**

Text

Description automatically generated

TO get the value from the JSON Payload: -

TO fetch the particular attribute value – such as **total from the json payload**:-



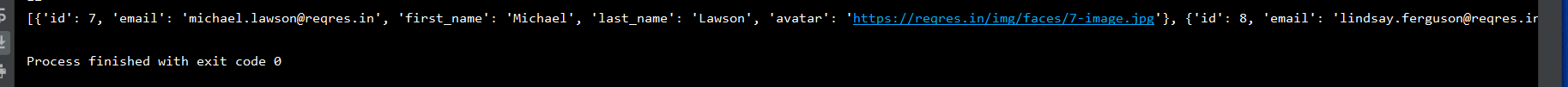
This will give **the value as 12 – and we are adding assertion to verify**:-

We want to get the **value of the JSON Array – data –(it will fetch all the values from the JSON Array)**

Text

Description automatically generated with low confidence

Output:-



TO get **the first element value inside the response data**:-

Text

Description automatically generated

Output: -

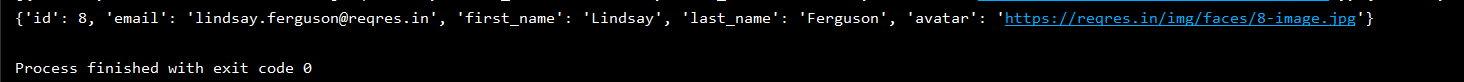


To get the element value inside it: -

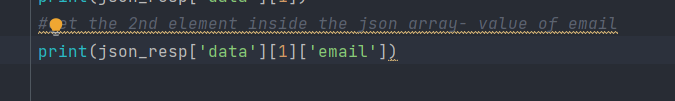
Text

Description automatically generated

Output: -



To get the element **value of email inside the array** :-



It will give the email value: -



We can pass parameter in the get request call – by mentioning the parameter as dictionary – and passing the dictionary inside the params parameter inside the get method

**Get Method Definition**: -

Text

Description automatically generated

Text

Description automatically generated

Now lets do POST Method call – to check whether we are able to create new user or not :-

Check the **posts method definition**: -

Text

Description automatically generated

Text

Description automatically generated

**Pass the payload** – **which we are passing in the post method as dictionary object**

Text

Description automatically generated

**Get the Response**: -

Text

Description automatically generated

Text

Description automatically generated

Now **pass the json payload into a json file and try to use it from there**: -

Graphical user interface, text

Description automatically generated

Pass the payload from the **json file and using the help of loads method – convert the string object to the python dictionary Object**

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application, website

Description automatically generated

Get the **headers response**: -



Same **thing we can do for the PUT Request**: -

PUT **Requests updates the entire resource** – and if the resource not found – it will **behave as a POST Request**: -

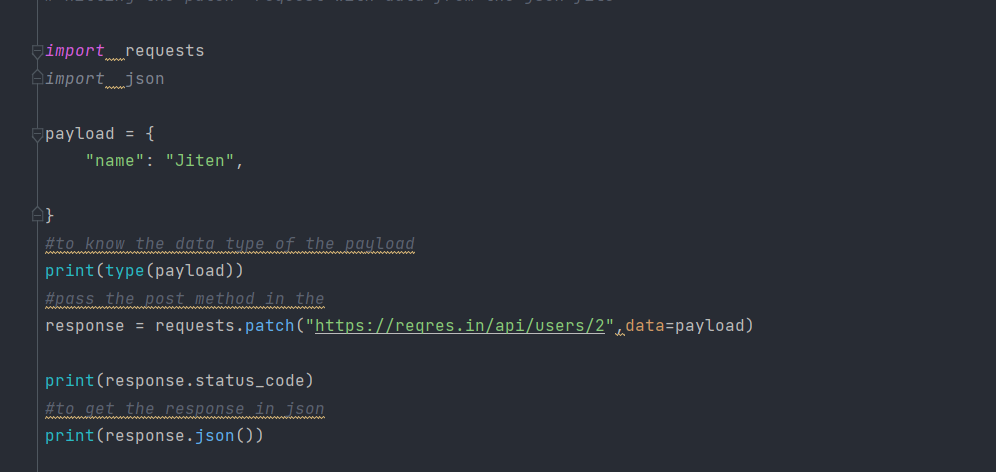


Output: -

Text

Description automatically generated

**PATCH Request** – Patch Request will **update a specific field which we wanted to update**: -



Delete the Request: -

Text

Description automatically generated

Text

Description automatically generated

**Timeout – the request parameter should match the timeout request**: -

Text

Description automatically generated

<https://requests.readthedocs.io/en/latest/user/authentication/>

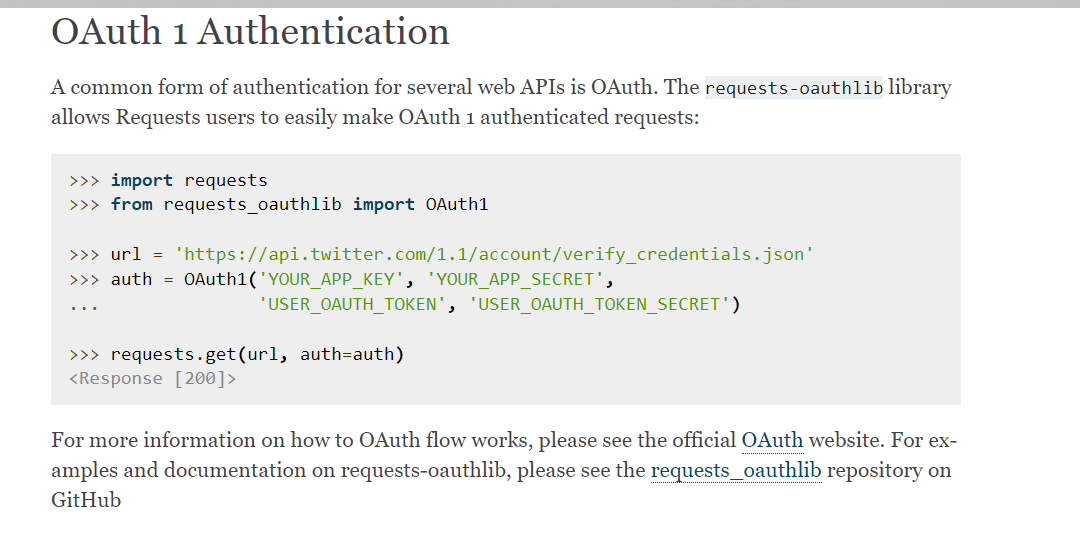
Passing Authentication Requests: -

**Basic Authentication**: -

Graphical user interface, text, application

Description automatically generated

OAuth 1 Authentication: -



**Examples showing the Basic Authentication**: -

Text

Description automatically generated

Output: -

Text

Description automatically generated