**Testing POST function:**

1. Start the development server

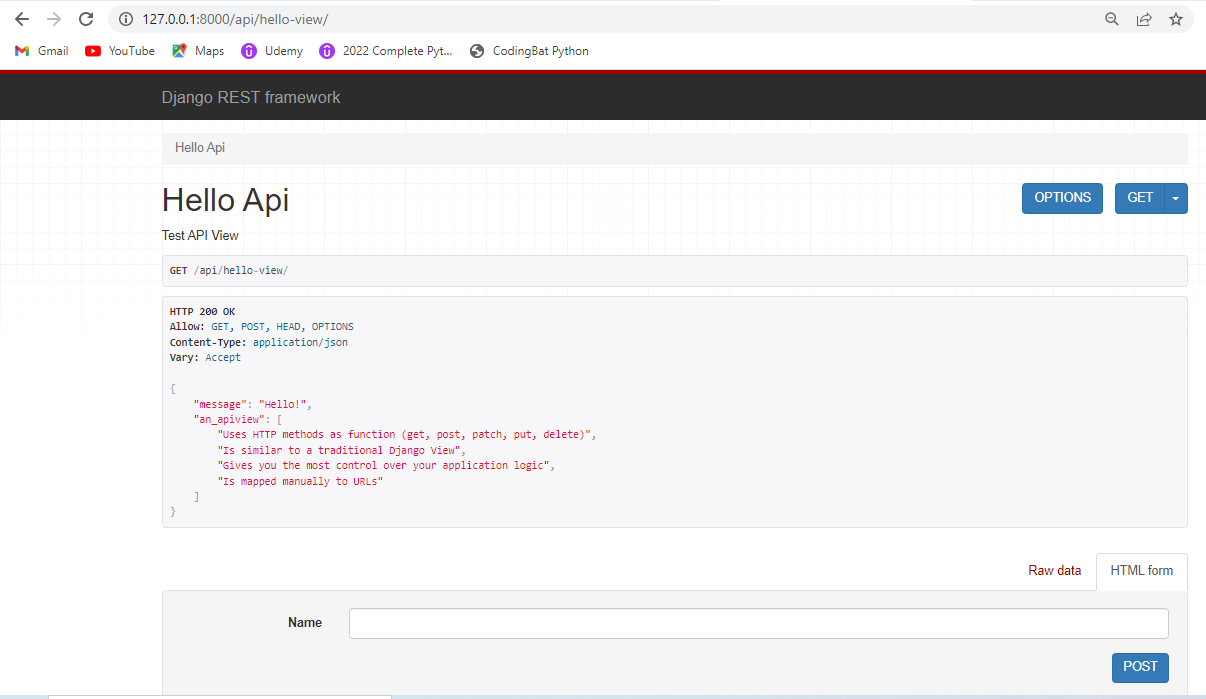
- vagrant up

- vagrant ssh

- cd /vagrant

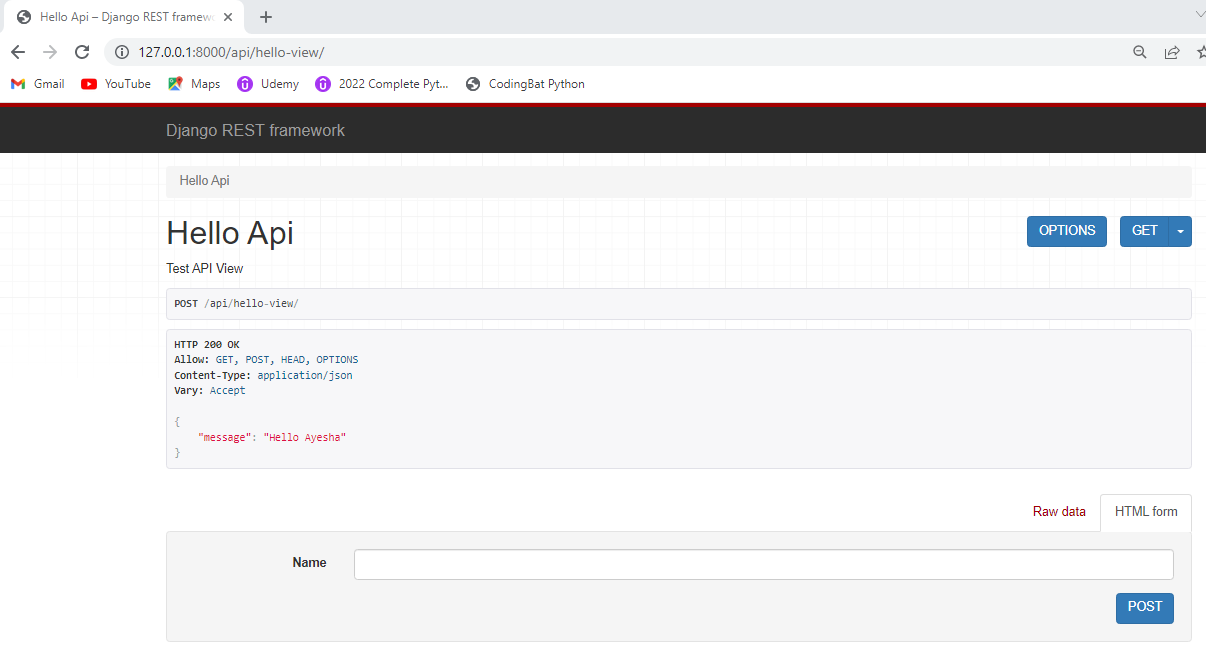
- source ~/env/bin/activate

2. Go to your browser and enter address ‘http://127.0.0.1:8000/api/hello-view/’



3. Enter a valid input (Name)

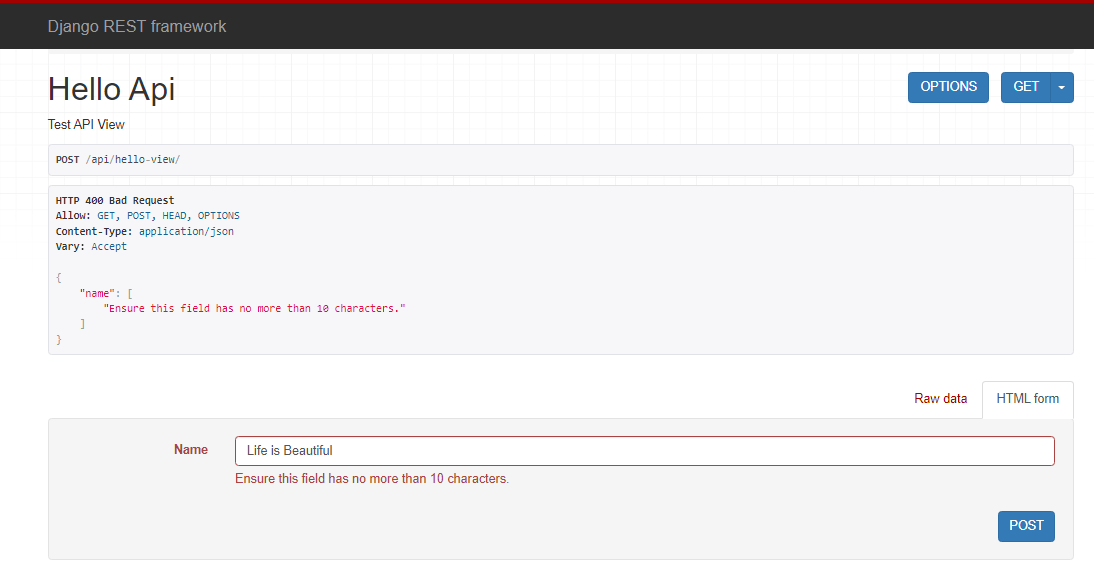
Click POST



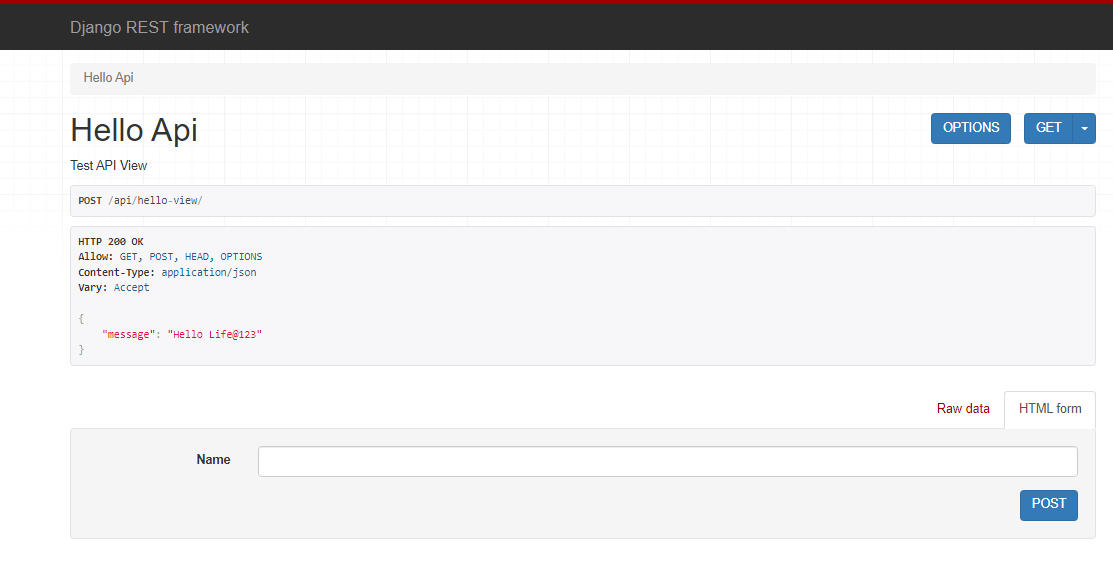
It displays the message correctly Hello + name (Hello Ayesha)

4. Enter an invalid data

Case 1.



Case 2.



**ADD PUT PATCH AND DELETE METHODS:**

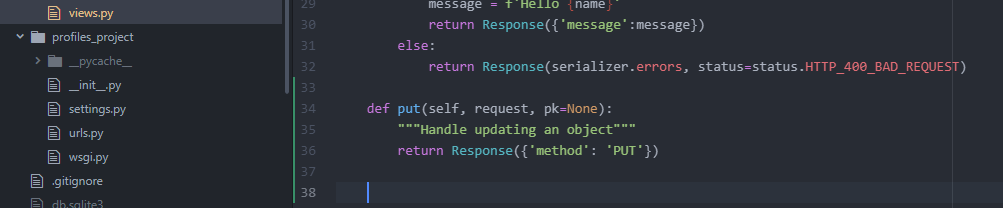
1. In atom editor open view.py file.

2. Adding new function for the HTTP PUT method.

HTTP PUT – used to update an object

(Make an HTTP PUT request and it will update (or replace) an entire object with what we have provided in the request)

pk – takes the id of the object to be updated with the put request.



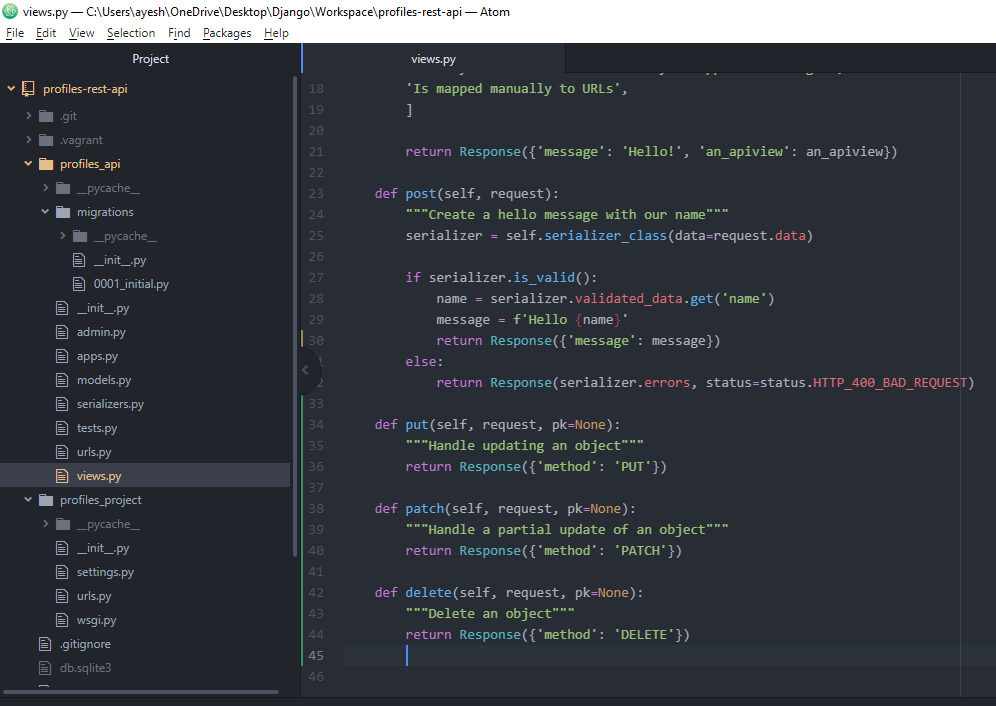
3. Adding new function for HTTP PATCH method.

HTTP PATCH – perform an update but only update the fields that were provided in the request.



4. Adding a new function for HTTP DELETE method

HTTP DELETE – used for deleting objects in the database.



**TESTING PUT, PATCH AND DELETE METHODS:**

1. Make sure that the development server is up and running.

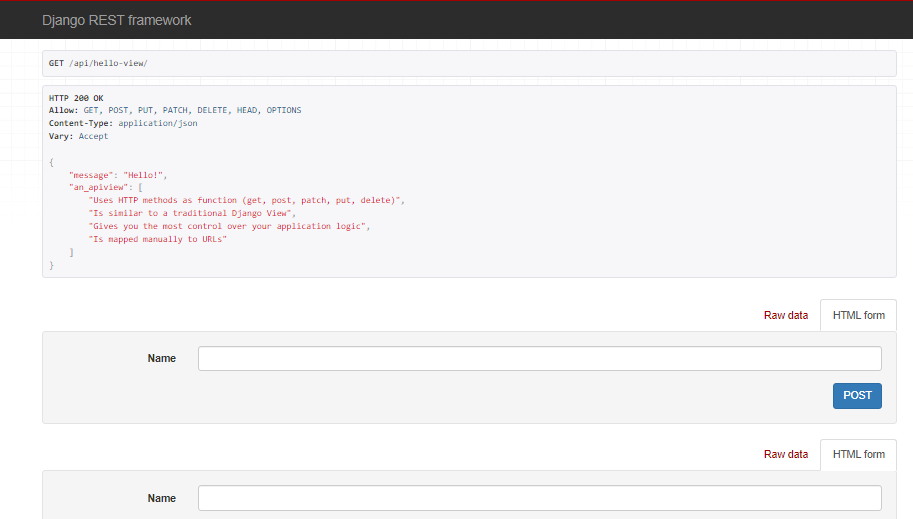
If it is already running, make sure that it is updated with the latest changes.

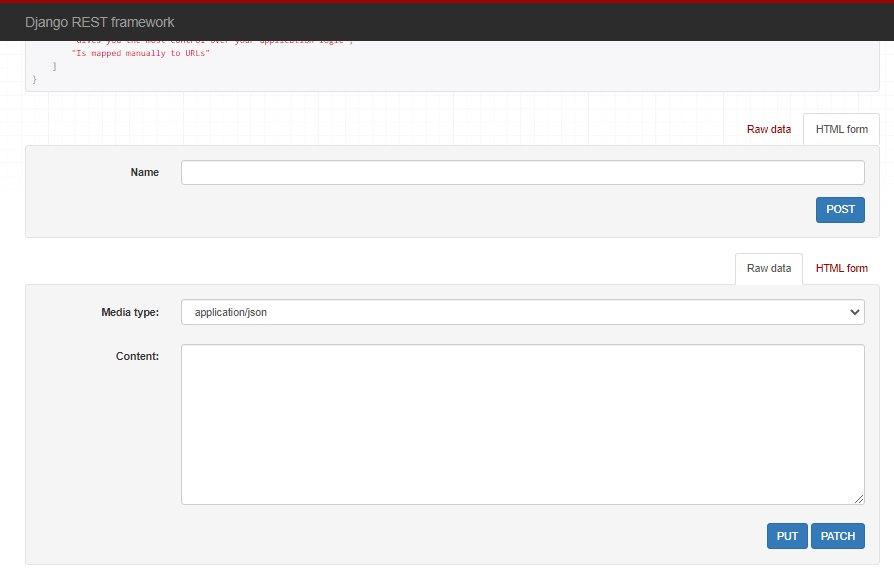
If it is not refreshed, press Ctrl-c to stop the server and restart it using

‘python manage.py runserver 0.0.0.0:8000’

2. Go to the browser and enter <http://127.0.0.1:8000/api/hello-view/>

\*\*\* I was not getting the PUT and PATCH option under HTML Form although I had started the dev server fresh (was only getting it under Raw data)





Try the following:

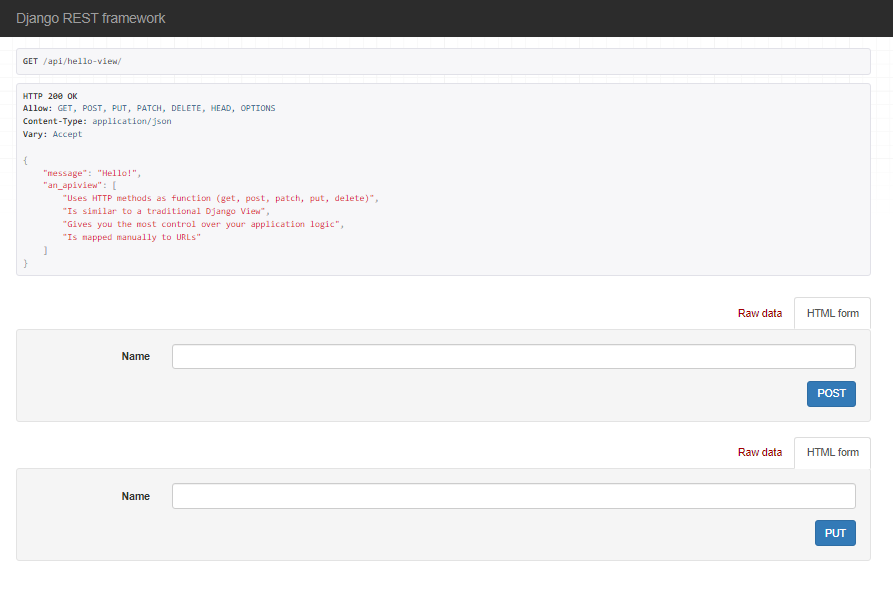
In git bash, enter Ctrl-c to stop the server and restart the server using

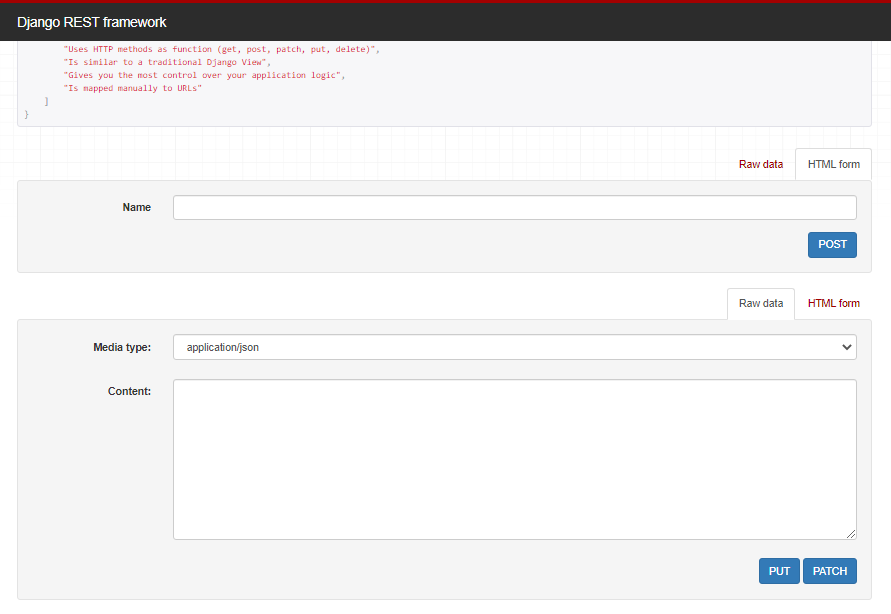
‘python manage.py runserver 0.0.0.0:8000’

Then check in HTML tab and Raw data tab.

HTML tab - shows only PUT

Raw data – shows PUT and PATCH

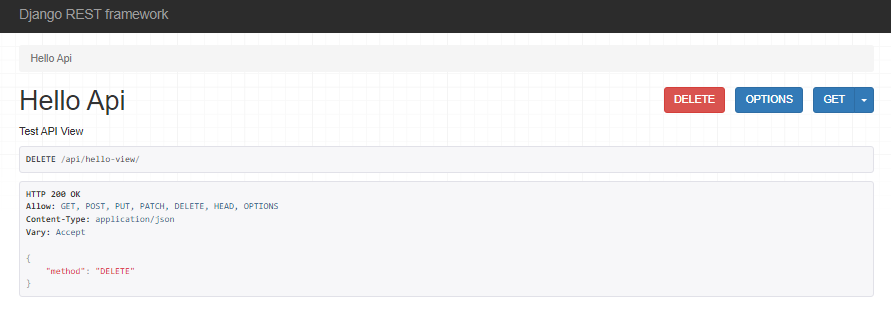




\*Note: Patch only appears in Raw data

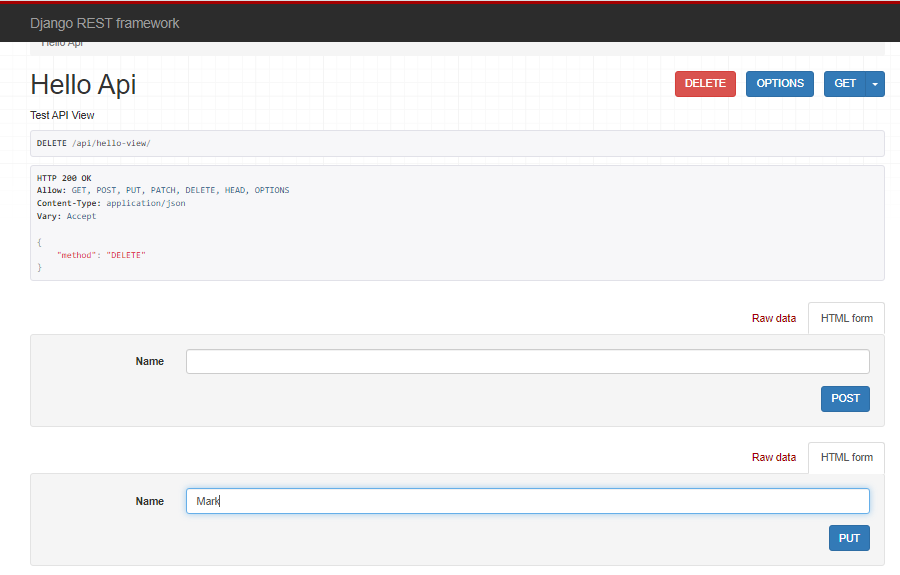
3. Testing DELETE

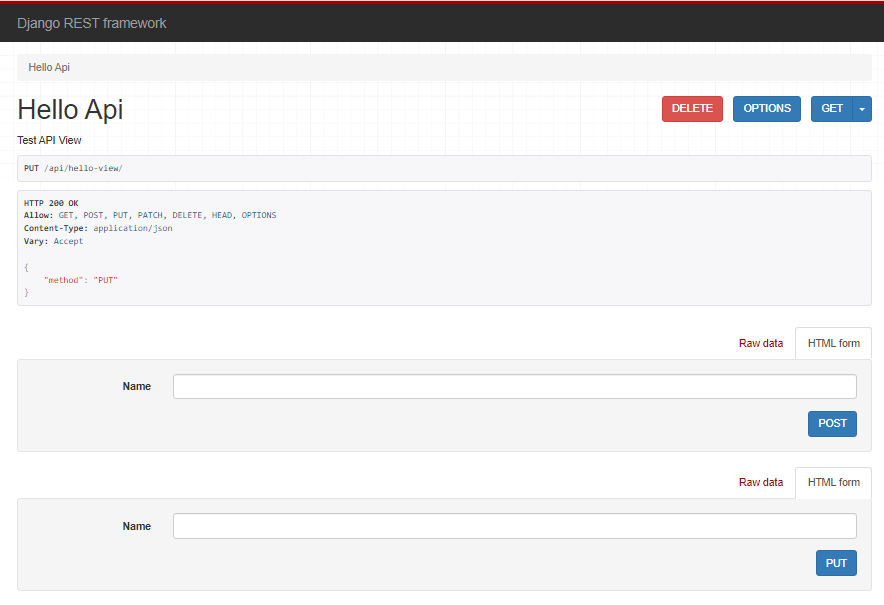
Click on DELETE button.



4. Testing PUT

Enter Name and Click PUT button.

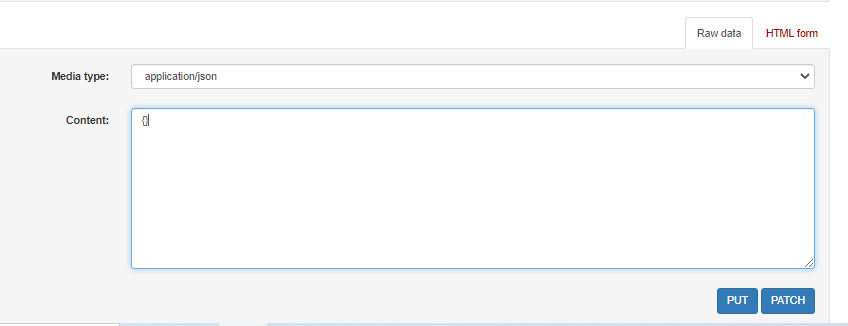


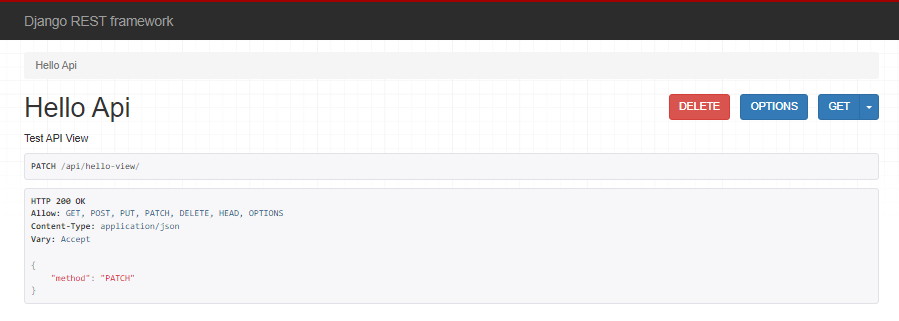


5. Testing PATCH

\*\*PATCH appears only in Raw data tab. Why?

The reason we need to use Raw data is because the way patch works is we need to provide only the fields we want to update in the API (HTML form has the list of all fields that are available). DRF requires that we define the content as json string.





**Push the Changes to GitHub**