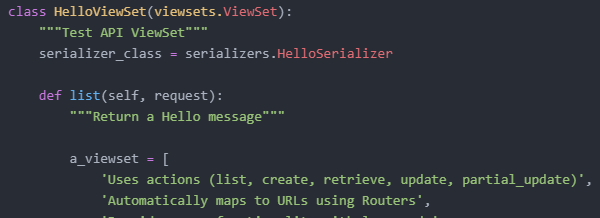
**Add Create, Retrieve, Update, Partial\_Update and Destroy Functions:**

1. Go to views.py in profiles\_api

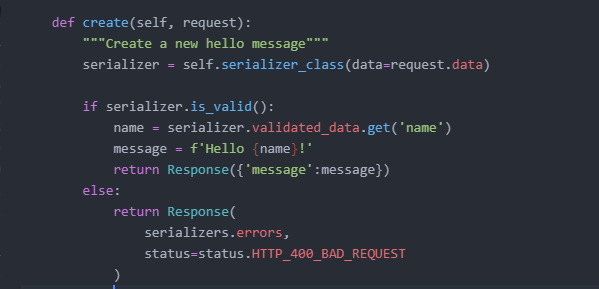
2. Add the serializer class

{We can use the same serializer (serializers.py) which we created for the APIView}

Specify the serializer in the ViewSet.



3. Add create function.



Retrieve our serializer.

We pass the data that was made in the request. It is passed using the data attribute of our serializer which is retrieved using the serializer class.

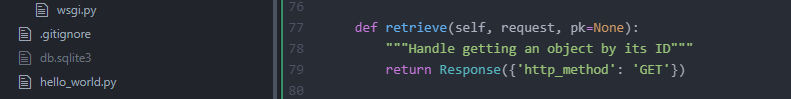
Validate the serializer.

List of errors that happened while validating.

4. Create retrieve function.

For retrieving specific object in our viewset.

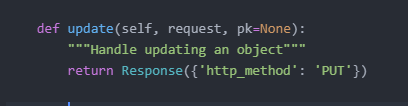
It is required to pass in primary key ID to the URL in the request and it will return/retrieve the object with that primary key ID.



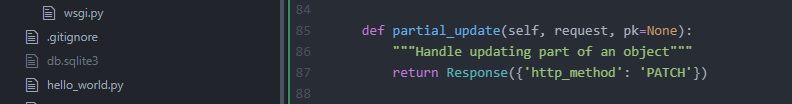
pk – Primary Key.

5. Add Update function.

Maps to a HTTP PUT on the primary key item of our viewset.

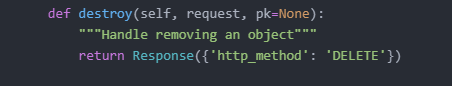


6. Add partial\_update function.



partial\_update maps to the PATCH request and that is for updating the fields that were provided in the request only (i.e. only part of the object)

7. Add destroy function.



Remove an object.

Maps to HTTP DELETE.

**TEST VIEWSET:**

1. Activate the development server.

> vagrant up

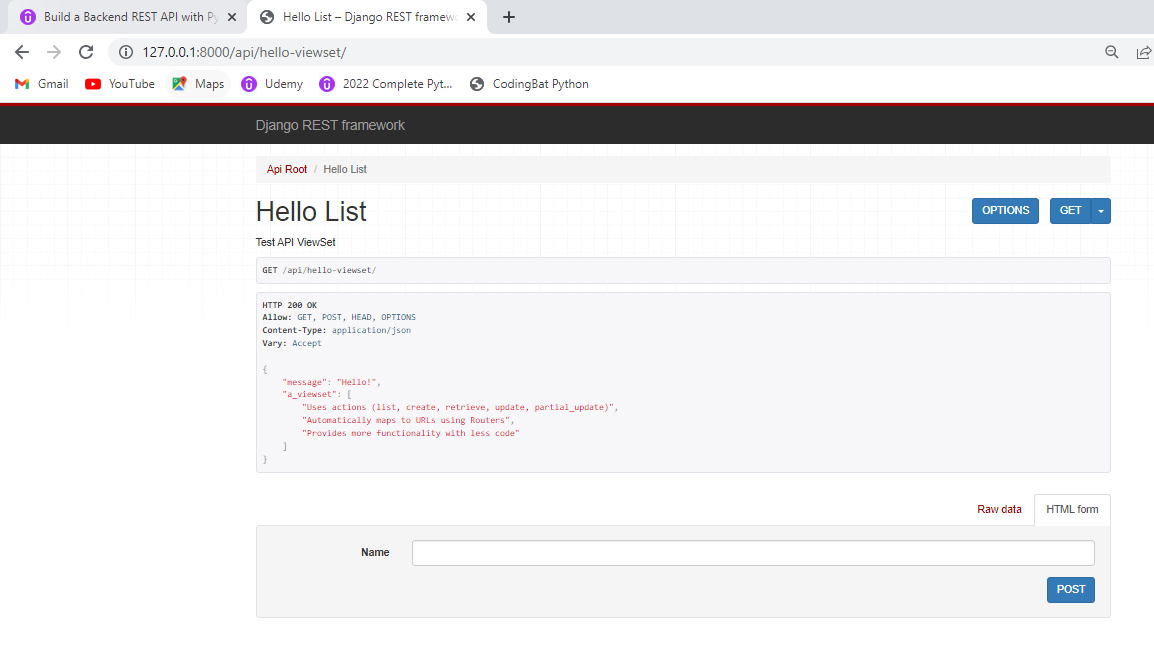
> vagrant ssh

> cd /vagrant

> source ~/env/bin/activate

> python manage.py runserver 0.0.0.0:8000

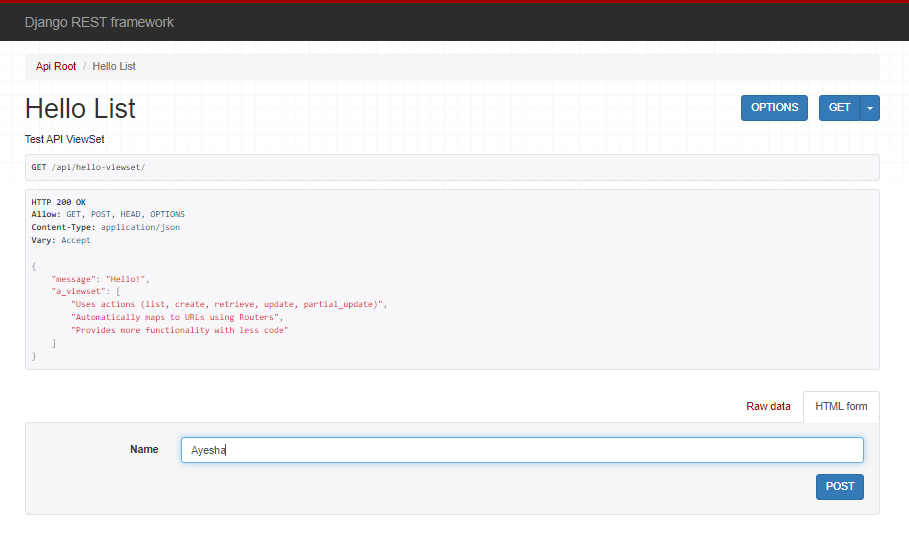
2. Go to your browser and enter <http://127.0.0.1:8000/api/hello-viewset/>



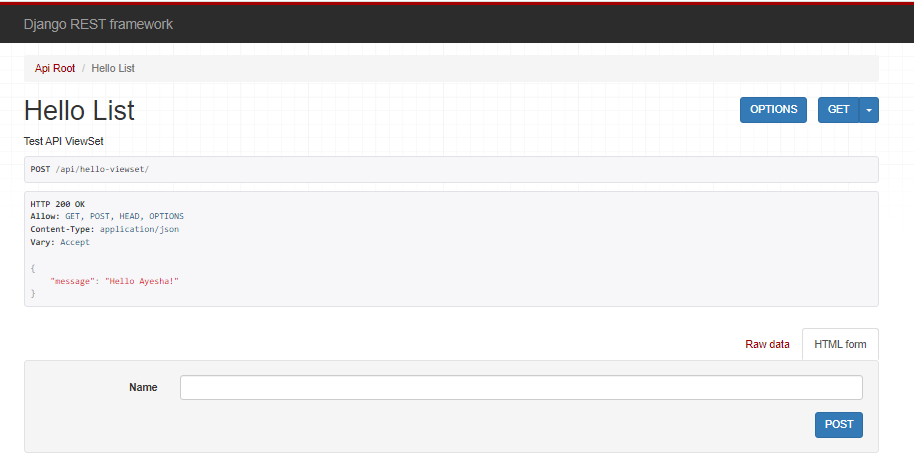
We do not find other options (PUT, PATCH, DELETE except POST) because the viewset expects us to use this endpoint to retrieve list of objects in the database.

POST:

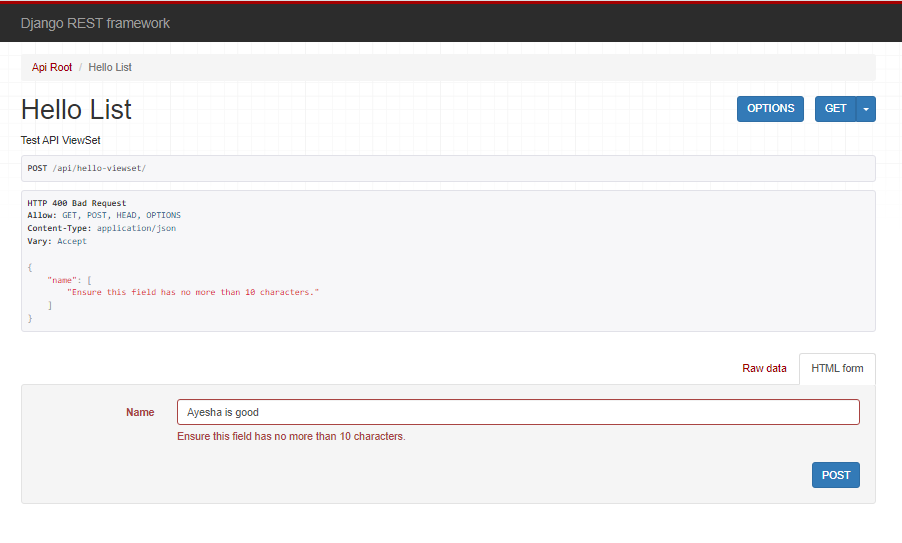
1. Provide valid input.



2. Post



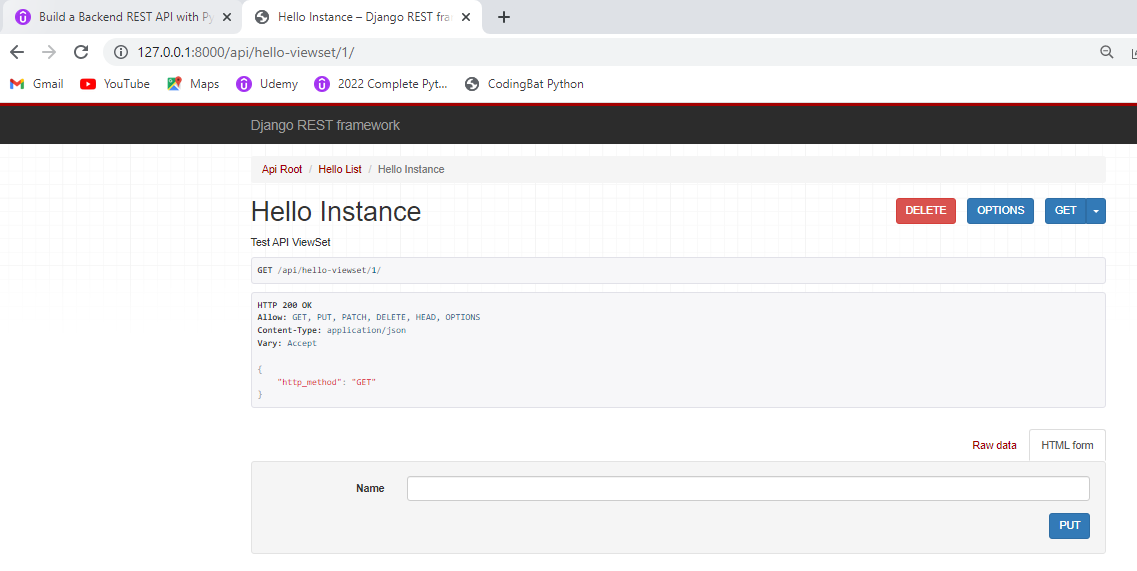
3. Provide invalid input.



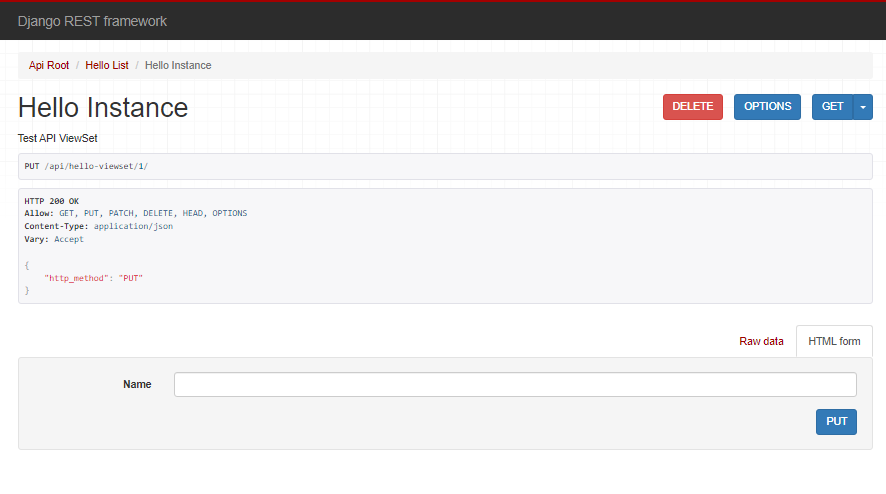
Need to specify a primary key id in the URL when making changes to a specific object.

GET:

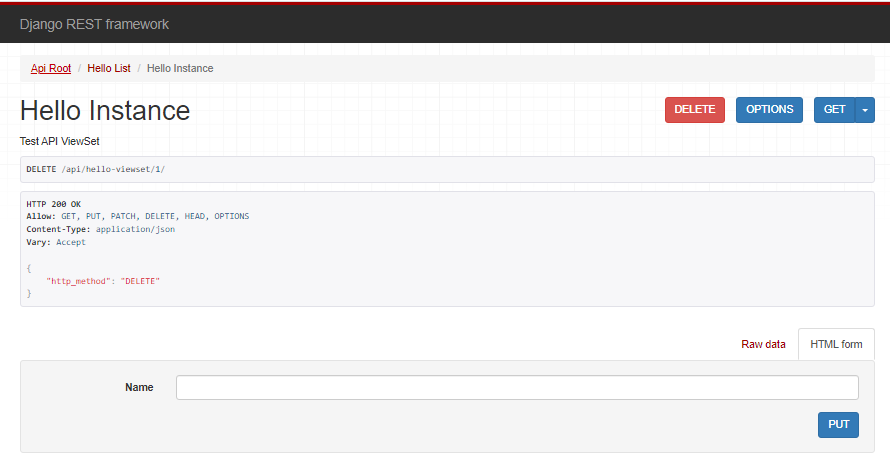
Enter primary key in the URL. Click Enter.



PUT:



DELETE:



PATCH:

