Score API

This is an example of the "score" exercise : https://www.hackerrank.com/challenges/breaking-best-and-worst-records/problem (https://www.hackerrank.com/challenges/breaking-best-and-worst-records/problem)

The service is exposed it through an API, also all the required unit test are provided

The API takes into consideration the following validation:

- Convert to positive every negative number
- Validate the correct input data
- Verify the existence of the endpoint
- Verify the correctness of the response data
- Validate the error messages

Installation

Via Docker

a DockerFile and DockerCompose files are provided just run:

- docker-compose up

The docker-compose file first run the test cases and after run the application exposing it in the port 8000

```
(base) C:\Users\fpitolre\Documents\develop\newteam>docker-compose up
Starting newteam_python_1 ... done
Attaching to newteam_python_1
python_1  | Creating test database for alias 'default'...
python_1  | System check identified no issues (0 silenced).
python_1  | ........
python_1  | ----------------------------------------------------------------------
python_1  | Ran 8 tests in 0.036s
python_1  |
python_1  | OK
python_1  | Destroying test database for alias 'default'...
python_1  | Watching for file changes with StatReloader
python_1  | Performing system checks...
python_1  |
python_1  | System check identified no issues (0 silenced).
python_1  |
python_1  | You have 17 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
python_1  | Run 'python manage.py migrate' to apply them.
python_1  | March 22, 2020 - 09:25:43
python_1  | Django version 3.0.4, using settings 'scores.settings'
python_1  | Starting development server at http://0.0.0.0:8000/
python_1  | Quit the server with CONTROL-C.
```

Directly

install python 3

located inside the app folder run

- pip install -r requirements.txt

Execute test cases

located inside the scores folder run

- python manage.py test

```
(newteam) C:\Users\fpitolre\Documents\develop\newteam\scores>python manage.py test
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
........
----------------------------------------------------------------------
Ran 8 tests in 0.091s

OK
Destroying test database for alias 'default'...

(newteam) C:\Users\fpitolre\Documents\develop\newteam\scores>
```

The test file is located at scores/test/test_ScoresReturn.py

scores > test > test_ScoresReturn.py > ...

```python
import json
from rest_framework.test import APITestCase
from rest_framework import status
from django.urls import reverse
from django.conf import settings
from django.urls.exceptions import NoReverseMatch

class test_Score(APITestCase):
    '''
    verify "score" url name exists
    '''
    def test_existScoreRoute(self):
        d = reverse("score")
        self.assertIsNotNone(d)

    '''
    verify that the post method is working
    '''
    def test_APIScores(self):
        data = {'scores' :[10,5]}
        d = reverse("score")
        response = self.client.post(f"{settings.TESTHOSTPORT}{d}",data,format='json')
        self.assertEqual(response.status_code,status.HTTP_200_OK)

    '''
    verify result 1
    '''
    def test_APIScoresResult_1(self):
        data = {'scores' :[10,5,20,20,4,5,2,25,1]}
        d = reverse("score")
        response = self.client.post(f"{settings.TESTHOSTPORT}{d}",data,format='json')
        self.assertEqual(response.status_code,status.HTTP_200_OK)
        self.assertEqual(int(response.data["max"]),2)
        self.assertEqual(int(response.data["min"]),4)

    '''
    verify result 2
    '''
    def test_APIScoresResult_2(self):
        data = {'scores' :[171645, 4199460, 1792941, 7634143, 5126340, 8930592, 3440006, 6437607, 3736481, 3236750, 5410071, 1
        d = reverse("score")
        response = self.client.post(f"{settings.TESTHOSTPORT}{d}",data,format='json')
        self.assertEqual(response.status_code,status.HTTP_200_OK)
        self.assertEqual(int(response.data["max"]),5)
        self.assertEqual(int(response.data["min"]),1)

    '''
    verify result when negatives values are provided since it is impossible to have negative punctuation
    these values will be converted to positives
```

Run application

located inside the scores folder execute the command

- python manage.py runserver 0.0.0.0:8000 --noreload

This will execute the application in localhost in the port 8000

```
(newteam) C:\Users\fpitolre\Documents\develop\newteam\scores>python manage.py runserver 0.0.0.0:8000 --noreload
Performing system checks...

System check identified no issues (0 silenced).

You have 17 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
March 22, 2020 - 04:09:35
Django version 3.0.4, using settings 'scores.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CTRL-BREAK.
```

The code of the API is located at scores/API/views_score.py

```python
from ServiceLayer import util
from rest_framework.response import Response
from rest_framework.views import APIView
from rest_framework import status
from rest_framework import exceptions

class Scores(APIView):
    '''
    http post method that expose the _calculateScore function
    '''
    def post(self, request,format=None):
        try:
            if not "scores" in request.data:
                raise exceptions.ParseError("please provide scores parameter")

            scores = [abs(int(i)) for i in request.data.get("scores",[])]
            max_count, min_count = self._calculateScore(scores)
            return Response({"max":str(max_count), "min":str(min_count)}, status=status.HTTP_200_OK)
        except Exception as err:
            util.raiseParentExceptionIfApply(err,exceptions.APIException)
            return Response({"error":str(err)}, status=status.HTTP_500_INTERNAL_SERVER_ERROR)

    '''
    return an integer array containing the numbers of times the scores were broken
    '''
    def _calculateScore(self,scores:"List of values ex [1,2,3,4,5]")->"dict with max and min keys":
        tmpmin = tmpmax = scores[0]
        min_count = max_count = 0
        for i in scores[1:]:
            if i > tmpmax:
                max_count += 1
                tmpmax = i
            if i < tmpmin:
                min_count += 1
                tmpmin = i
        return (max_count, min_count)
```

See the following code for test and use the application

In [4]:
```
import requests
import os

headers = {}
headers['content-type'] = 'application/json'
args = {}
args['verify'] = False
os.environ['no_proxy'] = '127.0.0.1,localhost'
data = {'scores' :[10,5,20,20,4,5,2,25,1]}
r = requests.post('http://localhost:8000/api/scores',json=data,headers=headers
,**args)
print(r)
print(r.text)
```

```
<Response [200]>
{"max":"2","min":"4"}
```