

Let me know how you feel...

Thank you for your interest in our Python Developer position, here we outline a task which we would like you to complete. We have attempted to explain as much as necessary on what needs to be done and what the expected result is, **please take the time to read the entire document first.**

The main purpose of this process is to give the candidate a clear understanding of the knowledge required for this position, and for the candidate to evaluate from their side whether we are a good fit or not for the next step in their career.

As a developer you will have the freedom to layout and code your project as best as you see fit, however, keep in mind that code structure, naming conventions and formatting will play a big role in the evaluation of your submission.

Provide your submission either as a **Git repository**, or a Zip(.zip) archive, as a reply to your invite.

There are different *levels* to this assignment, you are required to reach up to *Level 2* for a successful submission. Each level assumes the completion of the previous one, they are not self-contained.

You have 5 *calendar* days upon receiving this document to complete the levels and submit your work.

Technical Specification of Submission

- 1) Upon accessing your code, we will run it under on a clean **venv** Python 3.9.0 which will be generated as

```
fotis→ ~ ▷ python3 --version
Python 3.9.0
fotis→ ~ ▷ python3 -m venv virtualenv-[candidatename]
fotis→ ~ ▷ ls virtualenv-candidatename/bin/activate
virtualenv-candidatename/bin/activate
```

- 2) We will run *pip -r* to install any requirements which should be listed in the root of your project in a *requirements.txt* file.
- 3) After installing your requirements we will run a single executable which should be capable of managing the different tasks using command line arguments. The name of the arguments, and their restrictions are up to you.
- 4) We will ***NOT*** call you, ask you nor fix any bugs in order to run your program. If it doesn't work, we truly apologize for your time and effort, but we will not consider this application.

- 5) You can provide *some* documentation, however it is **not** required, we will mainly run the commands and look at the code.

Level 1

The main purpose of the task, is to be able to let us know how the world feels...

You first will need to consume the <https://newsapi.org> API, which provides several thousands of free calls, upon a *free* registration.

Once you have accessed your API key, your first step is to be able to provide a unique list of sources which match a given query.

- a) Your output *may* look like this

```
(virtualenv) macbook01 → src ▷ python main.py topic 'brexit'
```

```
{
  totalSources: 4,
  totalArticles: 1932,
  sources:[
    {
      'name': 'Forbes',
      'count': 3
    },
    {
      'name': 'The Newyorker',
      'count': 5
    }
  ]
}
```

Level 2

Having been able to consume information from the API, the next step is to analyze the data **using any open source** ML library, in order to provide a sentiment index. You can not use an external API, assume your submission will be run *offline*.

The name of the index itself as well as its granularity is up to the developer. For this task it can be binary 'sad'/'happy', '0'/'1', along with the percentage of articles categorized to each sentiment.

```
(virtualenv) macbook01 → src ▷ python main.py topic 'brexit' --sentiment
```

```
30% happy
20 % sad
50% neutral
```

Needless to say, greater granularity, better submission.

Level 3

If you would like to take it a step further, you can provide, through command line arguments, access to multiple ML algorithms.

```
(virtualenv) macbook01 → src ▷ python main.py topic 'brexit' --sentiment --methodA
```

happy

```
(virtualenv) macbook01 → src ▷ python main.py topic 'brexit' --sentiment --methodB
```

sad

Level 4

This last step is not a requirement, and can be considered advanced for the realms of this submission. However, if given enough time, and in order to solidify your submission, we would like you to be able add a *temporal* aspect, in which we are able to specify *any* temporal restriction on the query. We only need a single one, so it can be for example *year*.

```
(virtualenv) macbook01 → src ▷ python main.py topic 'brexit' --sentiment --year=2021
```

happy

```
(virtualenv) macbook01 → src ▷ python main.py topic 'brexit' --sentiment --year=2020
```

sad

Thank you, and looking forward to receive your submission.