Friday Code Challenge
Applicant: Niraj Dev Pandey
Position: Senior Software Engineer (Data Focus)

FRI:DAY

Table of Contents

1 Statement as to the sole authorship of the Report....1
2 Introduction....2
3 Tech Stack....2
3.1 Input format....2
3.2 Programming Language....3
3.3 Output Format....3
3.4 Requirements....3
3.5 Configuration....3
4 Project Structure....4
5 How to run....4
6 Test Coverage....4
7 Conclusion....5
8 Thank you note....5

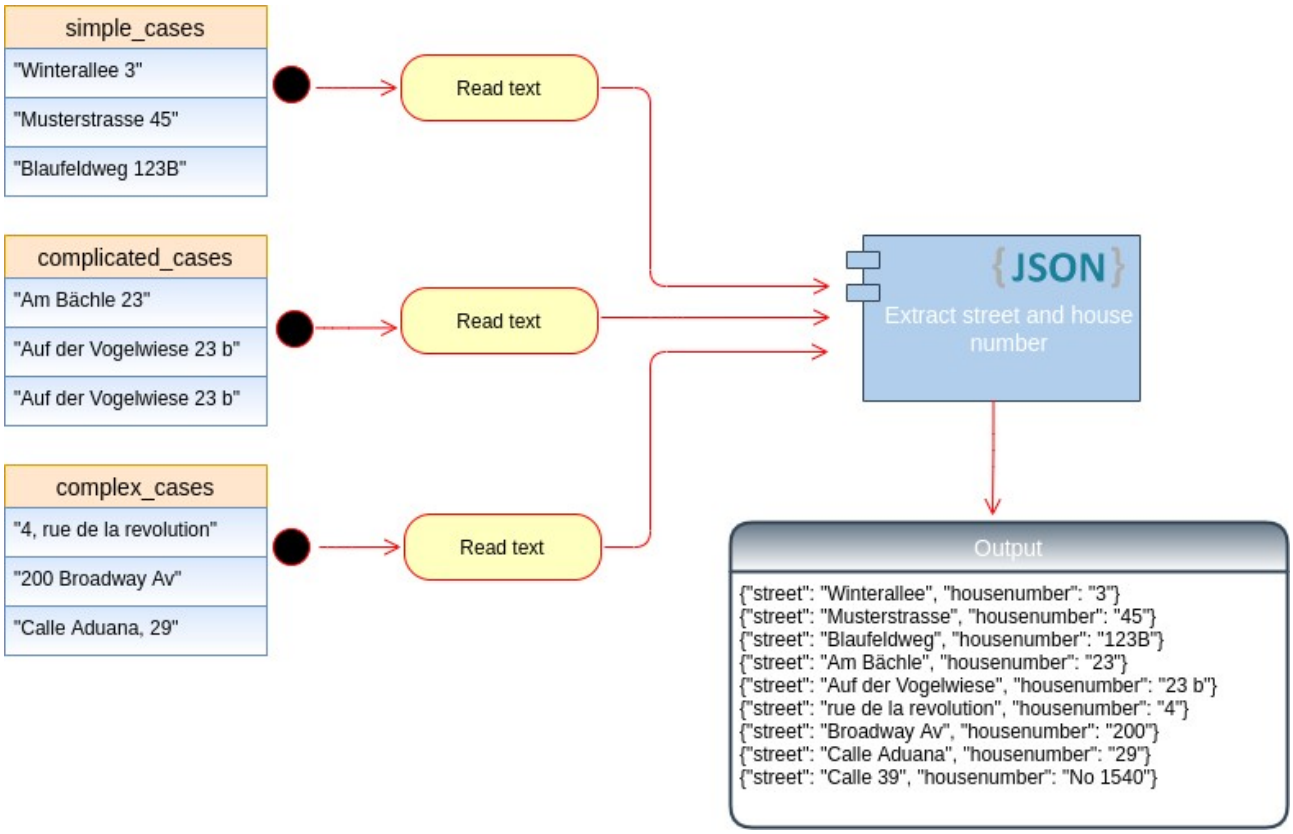
1 Statement as to the sole authorship of the Report

Senior Software Engineer (Data Focus)

I hereby certify that the report named above was solely written by me and that no assistance was used other than that cited. The passages in this report that were taken verbatim or with the same sense as that of other works have been identified in each individual case by the citation of the source or the origin, including the secondary sources used. This also applies for drawings. sketches, illustration as well as internet sources and other collections of electronic texts or data, etc. The submitted report has not been previously used for the fulfillment of a job application requirements and has not been published in English or any other language. I am aware of the fact that false declarations will be treated as fraud.

2 Introduction

An address provider returns addresses only with concatenated street names and numbers. Our own system on the other hand has separate fields for street name and street number. That being said we want to have a system where provided that concatenated string of addresses we can separate street name and house number. So the project boiled down to the following technical specification.



3 Tech Stack

No.	Item	Description
1.	Input Format	Text files (single or multiple)
2.	Programming language	Python 3.7.7
3.	Output format	Json object
4.	Requirements	Pip requirements.txt
5.	Configuration	Config.yaml where one can define data path for input and for output

3.1 Input format

The input can be any number of text files. You can put all input data into the folder names “data” in the home folder.

3.2 Programming Language

As described in the task itself that we at Friday feel home with Python. My choice is exactly the same. This project uses Python 3.7.7. Although I am sure that any Python version 3 will work exactly same. Remember not Python 2. There are huge differences in these two versions.

3.3 Output Format

The result will be saved as an Json object.. The output format follows the directives on the task itself. Here is a glimpse.


```
{"street": "Narrowway Av", "housenumber": "500"}
{"street": "Chile Aduana", "housenumber": "29"}
{"street": "Indrapuri 39", "housenumber": "No 1540"}
{"street": "Winterallee", "housenumber": "3"}
{"street": "Musterstrasse", "housenumber": "45"}
{"street": "Blaufeldweg", "housenumber": "123B"}
{"street": "Am Bächle", "housenumber": "23"}
{"street": "Auf der Vogelwiese", "housenumber": "23 b"}
{"street": "rue de la revolution", "housenumber": "4"}
{"street": "Broadway Av", "housenumber": "200"}
{"street": "Calle Aduana", "housenumber": "29"}
{"street": "Calle 39", "housenumber": "No 1540"}
```


After running the script one can see the output in the terminal as well. After that you can find the final outcome in the folder called “result” which resides in the home directory.

3.4 Requirements

I assume that you have Python 3 installed with Pip. If yes, the project doesn’t have any other big requirements as such. Almost all the module used have been provided by Python default. There is just one exception. We are using configuration file to provide input and output path. For that one has to install PyYaml. To install this prerequisite you can either use ->> *pip install -r requirements.txt* in the terminal *from project directory.* Or we have better way to do it which we will see in coming sections.

3.5 Configuration

The configuration file acts like a brain for projects in general. As we can see that this is very small project and therefore there are very less configuration to define. Nevertheless, config.yml file accepts two parameters. One is the location of the input file path and other is the location where you want to save the results.

4 Project Structure

```
.
├── data
│   ├── another_address.text
│   └── simple_address.txt
├── doc
│   ├── report.odt
│   └── report.pdf
├── result
│   └── result.json
├── test_coverage
│   └── test.py
├── address_filter.py
├── config.yml
├── main.py
├── read_text_file.py
├── requirements.txt
└── run.sh
```

5 How to run

I have created a bash file for the user to easily run the script. I am assuming the you are a Macintosh or Linux user and can execute bash script directly. Type ->> *bash –h* to see if you have it already. Now, since you have bash and you have already put your input files in the folder called “data”. Additionally you have provided your output path location in the configuration file. Finally open terminal inside the project home directory and type ->> *sh run.sh.* This command will automatically install desired Python library for you and execute the whole project.

In other case one can simple also execute *main.py* file after requirements are done and you will see the same result.

6 Test Coverage

Given this small task we don’t have much to test. Nevertheless, The “test_coverage” folder contains a file called “test.py” which has 4 test cases to verify. Starting from verifying if the files you are reading are indeed a text file. It will print information with the file name if they don’t have .txt extension. There are two hard test cases which see if the file name that this project was tested upon are present and valid. The rest 2 test cases are about finding if the Python regex works as expected. The doc string attached to them explain what is expected and when the test will fail.

7 Conclusion

The given input examples in the challenge can be seen to be working perfectly fine. I have tried some more examples myself and can see that those also returns fine results. The project can be simplified in just one single python script too but beautiful code is always better than ugly. Separating different module in separate files organize the whole process in a better way. Again, there was no need to have configuration file but in my view it's better for inexperienced user to provide their data file path there without looking inside the code to find that place. Readability counts therefore, all across the project Pep-8 standard was used. You can find doc strings wherever any function or classes have been used. The comments are at every lines to understand the process. The logging was used to raise right kind of message to the user. Since Explicit is better than implicit therefore, right naming conventions was used. If the implementation is hard to explain, it's a bad idea therefore my effort was to make it quite easy to understand and adopt.

8 Thank you note

Thanks a lot for such an exciting task. I learned a lot about regex and specially how I can make a project as such that it is going to be used by others. How the structure should be, how test cases help us to debug the error sooner than later. The code readability was always something I liked and I practiced with this task as mush as I could. I hope that you will like my solution and we will speak further to see each other compatibility. Have a nice day and Happy New Year :)

Submitted by : Niraj Dev Pandey

E-mail: nirajdevpandey@gmail.com

Website: https://nirajdpandey.github.io/

GitHub: https://github.com/nirajdpandey