1 The script from hell

[get_deliveries_from_backend.py]
Some months ago an external consultant wearing an expensive suit created the attached Python script for our company. This script performs a simple ETL function, copying delivery data from the backend of a food delivery platform to a PostgreSQL database table on a daily basis. This script works, but has some issues, the first of which being that it is not pretty to look at :-)
Let's discuss some improvements that might be made. Note: The idea here is to describe at a high-level how would you improve the script: there is no need to write code or to discuss individual lines within the code, just the approach you would use to make improvements. The questions below provide more details.

1. In order to improve the get_deliveries_from_backend.py script, we want to put it under an external version control system (like Github). Therefore we first need to remove the credentials from the code. But we do need to store them somewhere and to make them available when the script runs.
    - What ways can you propose to achieve this? - Please, start with the simpler ideas and give an advantage and an disadvantage of each
2. We want to schedule this script to be run periodically in an automated way.
    - How would you change the script so that it is easy to check whether it ran successfully? [Note: "Run successfully" means that the script performed the task of copying the data]
    - How would you change the script so that, if it fails, it is easy to diagnose what the problem was?
    - Which ways (or tools) do you know that could be used to run scheduled jobs like this one? [Again, start with the simpler ones and give an advantage and disadvantage of each]
3. We know that we will need to write another script soon that will access the backend API and generate some PDF report. Since the requirements include accessing the backend API, a possible starting point would be to copy this script and modify it. Is this a good approach? What alternatives exist?
4. We want to move to a development approach based on continuous integration.
    - How would you improve this script in order to make it suitable for a development !environment with a continuous integration infrastructure? The same code should run on the developers' machines, staging environment, production environment and the continuous integration system.
    - What building blocks would such an environment have? What tools would you use?

2 Financial Times

A well-known online food ordering company have just built a data warehouse to monitor financial data from several of their international websites. Unfortunately, all of the financial data is in the local currency for each website - meaning there is currently no way to have a consistent overview and comparison of all the websites together.
In order to tackle this problem, they will need to have access to historical exchange rates for all currencies in order to convert the values to Euros.
Your job is to do the following:
Write a python class that extracts historical and current exchange rate data for all currencies from the http://fixer.io/ API and displays the results neatly.
Your script should have the following characteristics:

- Since only the past 2 years' worth of data is required for reporting, the date range should be parameterized and configurable.
- While Euros are the intended base currency for the initial reporting, it should also be possible to choose USD as a base for future reporting.
- The data should be retrieved and stored in a suitable construct. E.g: consider using SQLite to create a table to hold rows containing the following columns: base_code (e.g: EUR), date, rate, currency_code
- Please also enable the calculation and display of the AVERAGE exchange rate for any given currency for a specified time range

N.B: Feel free to use any library (Flask, Petl, Pandas....) that might speed up your developments. No need to reinvent the wheel :)

Answers

Please, send us your answers back via email (for the script, send all the code). The idea is that the whole exercise shouldn't take more than about 2 hours of your time, but we may have underestimated the amount of work needed. In that case, please do not invest more than 2 hours, and do tell us so we can shorten the tasks.
Thank you for your time.