**Week 11 Programming Assignment documentation**

**Task 1:** Take a look at the code your wrote for your backup-system and -API. By now, you should have a reasonably complex project consisting of multiple files. It might be getting a bit complex to navigate around and have a good idea of how everything fits together.

* 1. Start improving this situation, by adding type hints (for both the arguments and return types) to all of your functions and methods in every file.
  2. If the IDE/code editor gives you any warnings or errors in places where you call functions/methods, look at what is happening and fix the issue.
  3. Wherever you initialize a variable with something that doesn’t give a clear indication of what kind of data it’s going to contain, also add type hints there. Examples of these are (amongst others):
     1. Initializing a variable with “None” (e.g.: user\_input = None)
     2. Initializing an empty list (e.g.: student\_list = [])
     3. Initializing an empty dictionary (e.g.: student = {})

**Answer:** Firstly, I skimmed through links in order to know how I can correctly add type hints and applied gained knowledge to my code (backup.py, app.py, centralized\_log\_api.py). I don’t think that I need some explanation why and where I added this hints, because it will be the same explanation in many lines. You can check changes in the code itself by accessing my Gitlab account

**Task 2:** For this exercise, installing and enabling an extension/plugin might help you; for example for VSCode the extension “autoDocstring”.  
  
Add docstrings for:

* 1. Any Python file in your project (at the top of the file)
  2. Any class you define
  3. Any function or method you define, including:
     1. A description of what it does
     2. A description of each parameter
     3. A description of what it returns

**Answer:** As in task 1 I think that I don’t need to provide some screenshots or explanations what I did and why, because what I did it’s basically explanation of code itself. You can check everything in my Gitlab repository

**Task 3:** Create documentation (for example a readme file) describing how the software should be used. That includes general end-user usage instructions, installation instructions and configuration instructions.

**Answer:** I can copypaste instructions from readme file here and leave the file in Gitlab repository.

Welcome to backup job app readme! (instructions file)

My app has following functionalities and arguments:

("-s", "--source", “Source directory")

("-d", "--destination", "Destination directory")

("-l", "--log", "Log filename")

("-v", "--verbose", "Verbose mode (log successful copies)")

("-db", "--database", "Path to the database file")

("-q1", "--query-files", "Query files in a certain directory")

("-q2", "--query-logs", "Query logs related to a certain file")

("-q3", "--query-all-logs", "Query all logs for files in a certain directory")

("-dt", "--date", "Filter logs by date")

("--display-job-info", "Display information for a specific backup job")

("--display-job-logs", "Display log entries for a specific backup job")

In order to create a backup for your directory you need to launch terminal, go to path with your backup.py file. After that you can create a backup of directory by the example below.

python3 backup.py -s /Users/spiceindeedx/Desktop/test\_db -d /Users/spiceindeedx/Desktop/backups -l log\_file.log -v

Also there are provided opportunity to make query of the database

Query the database to display a list of all files from a certain directory. (The directory is a parameter to your script):

python3 backup.py -s /Users/spiceindeedx/Desktop/test\_db -d /Users/spiceindeedx/Desktop/backups -l query\_log.log -v -q1 /Users/spiceindeedx/Desktop/test\_db

Query the database to display a list of all log messages related to a certain file. (The directory and filename are parameters to your script):

python3 app.py -s /Users/spiceindeedx/Desktop/test\_db -d /Users/spiceindeedx/Desktop/backups -l log\_file.log -v -q2 /Users/spiceindeedx/Desktop/test\_db test\_text1

Query the database to display all log messages for all files from a certain directory. (The directory is a parameter to your script):

python3 backup.py -s /Users/spiceindeedx/Desktop/test\_db -d /Users/spiceindeedx/Desktop/backups -l query\_log.log -v -q3 /Users/spiceindeedx/Desktop/test\_db -dt 2023-01-01

Display job information

Display log entries

python3 app.py -s /Users/spiceindeedx/Desktop/test\_db -d /Users/spiceindeedx/Desktop/backups -l log\_file.log -v --display-job-info 1

With app as a gift you will receive centralised log api server. You can also use it with terminal. Examples of commands you can find below:

# Add a new system

curl -X POST -H "Content-Type: application/json" -d '{"name": "System7", "ip\_address": "192.168.1.7"}' http://127.0.0.1:5000/systems

# Add a new log entry

curl -X POST -H "Content-Type: application/json" -d '{"system\_name": "System4", "log\_date": "2023-01-01", "log\_level": "INFO", "message": "Log message", "directory": "/path/to/directory"}' http://127.0.0.1:5000/logs

#retrieve all logs

curl http://127.0.0.1:5000/logs

# retrieve log of a specific system

curl http://127.0.0.1:5000/logs/system/System1

# retrieve log of a specific system

curl http://127.0.0.1:5000/logs/date/2023-01-02

# delete logs before specific date

curl -X DELETE http://127.0.0.1:5000/logs/delete-before/2023-01-01

# update a log entry

curl -X PUT -H "Content-Type: application/json" -d '{"system": "UpdatedSystem", "log\_level": "ERROR", "message": "Updated message", "directory": "/updated/directory"}' http://127.0.0.1:5000/logs/update/8

The apps were designed by Artem Tiutenko