Coding assignment: Quantitative Strategist Associate

March 21, 2023

1 Question

In this scenario, the strategy desk runs a systematic trading book. Every week, our proprietary algorithm generates a suggested target portfolio allocation (units of %). This will take the form of a JSON file. The task is to write a system that generates the necessary trades to be executed by execution traders. And so the ouput of the program should be a JSON file of trades (units of number of stocks), while the input is a JSON file specifying the target portfolio allocation; both of which have been provided. To simulate the market, I have provided a Broker class, which will randomly generate prices. The Broker will also store the position holdings of your portfoilio. I have also included a Positions class which is a dictionary wrapper class that tells us how many units of a stock are being held. Both JSON inputs and outputs are included. Write a RebalancingSystem class that uses these components to generate the json file of trades for the execution trader. Do not modify the logic of the existing classes but feel free to extend or to create new classes. You can initiate the broker with a hardcoded Positions class with 0 units of inventory. Please make sure that the code is as abstracted as possible and please use python's native logging clsss to output logs. Please create a github repo and post your completed solutions there.

```python
class Broker:

    def __init__(self, initial_positions: Positions, initial_aum: float):
        self.positions: Positions = initial_positions
        self.aum: float = initial_aum

    def get_live_price(self) -> Dict[str, float]:
        prices = {asset: random.uniform(10, 30) for asset in \
                    self.positions.get_universe()}
        return prices

    def get_positions(self) -> Positions:
        return self.positions

    def execute_trades(self, execution_positions: Positions) -> None:
       pass

class Positions:

    def __init__(self, positions: Dict[str, float]):
        self._pos = positions

    def get_universe(self) -> List[str]:
        return list(self._pos.keys())
```