Overview

The aim of this task is to pull information from an API, into a database, and be able to query it. We would like you to use SQLAlchemy for this task, and store the results in a SQLite database. We have provided some skeleton code to get you started, and will be reviewing how you use Pythonic style, and OOP.

Task Description

Given a list of train station codes (three letters), and a starting time, we want you to find the arrival time at the destination station.

For example, given the following set of train codes, a start_time of `'2022-02-09 14:17'`:

```
['LBG', 'SAJ', 'NWX', 'BXY']
The code should return the arrival time of: 2022-02-09 14:49:00

['LBG', 'SAJ', 'NWX', 'SAJ']
The code should return the arrival time of: 2022-02-09 14:41:00
```

Notes:

- You should store information in the database to speed up subsequent runs of the code.
- You should not find the shortest time between start and destination stations, but follow the route.
- You will be given the maximum time (in minutes) that someone is willing to wait at a station. If they have to wait longer you should display an error.
- The Transport API can be found here. You need to sign up for a free trial account.

Skeleton Code

```
SQLALCHEMY_DATABASE_URL = "sqlite:///./trains.db"

engine = create_engine(SQLALCHEMY_DATABASE_URL, connect_args={"check_same_thread":
False})
Session = sessionmaker(autocommit=False, autoflush=False, bind=engine)

Base = declarative_base()
MAX_WAIT_TIME = 60 # minutes

with Session() as session:
   Base.metadata.create_all(session.get_bind()) # Create tables in database

   # Run code and output time here
```

Bonus Points

- Implement an API that allows you call your function through a REST/gRPC call