PORTLAND STATE UNIVERSITY

ECE 508 Python and Scripting Workshop Project

Spring 2020

Adya Pandey

Aman Singh Solanki

Final Project Report: Trimet Arrivals Information

This project displays the Trimet MAX and Streetcar arrival details for any specific stop for the next 1 hour. Our objective was to learn how Python interacts with APIs and to display the data it provides which will be useful for an everyday public transport user.

Our landing page takes two inputs from the user i.e. StopID (mandatory) and duration(in minutes and optional). The duration of 60 minutes will list all the arrivals in the next 60 minutes. If the duration is left blank, the default value is 20 minutes. Based on the resultSet received from the Transit API, we display the stop name, arrival details like vehicle name, scheduled time of arrival, and its status. We also display an error pop-up if anything doesn't work as expected or an invalid input has been provided.

GitHub: https://github.com/amans330/Python-Project

Requirements

The application requires some packages to be installed.

1. **Tkinter:** The tkinter package ("Tk interface") is the standard Python interface to the Tk GUI toolkit. Both Tk and tkinter are available on most Unix platforms, as well as on Windows systems[1]. We use this package to provide GUI to the user to take input from the user and display the search results.
2. **Webbrowser:** The webbrowser module provides a high-level interface to allow displaying Web-based documents to users[2]. We use this package to redirect the users to the Trimet Website for finding the station id (stop id in our application). Trimet does not expose an API for doing this, hence it needs to be done manually by the user.

3. **urllib.request:** The urllib.request module defines functions and classes which help in opening URLs (mostly HTTP) in a complex world — basic and digest authentication, redirections, cookies, and more[3].
4. **AppID:** The owner of this application needs to create an AppID for making the requests to the Trimet API. The owner would need to register themselves with the Trimet using their email and an AppID would be sent on his/her email. The AppID is sent with all subsequent requests. We have registered ourselves with Trimet and are using our own AppID for this project.

How to Run the Application

On the terminal, go to the folder where the application is saved and execute the command python3 index.py. It will open the landing page which will ask the user to provide some input. We have included a tutorial for the user, which lists the input needed and its correct format which can be accessed from the Help -> Tutorial button. It explains the use case for this application and the information it provides. Help button is on the top left and also has the Trimet site listed to lookup Stop Ids. The application runs infinitely until the user clicks on the “Quit” button. By keeping the application infinitely, the user can keep giving different search parameters and search results without any hassle.

Code Description:

on_submit(): This method checks the input and makes sure it is in the correct format. The Stop Id and time should both be digits. The duration is optional, but the maximum duration we are allowed is 60 minutes. If everything is correct, we make the Http GET request to the Trimet API and get the response in the JSON format. We convert it into a dictionary on which we perform the operations as needed.

display_data(json_obj): Display data method shows the arrival records to the user in an easy to understand format. The method takes the json_obj (a dictionary) as input. It searches through the json_obj to find the full name of the stops, the scheduled time, and the estimated time. It also displays a status for the method of transportation by checking if the estimated time is later

than the scheduled time or if it's on time. The method spawns up a new window to display this information for the user.

open_readme(): This method displays the tutorial for the user. It explains the use of our application and the input user needs to provide. It supports Trimet Max Rail, buses, and Streetcar arrivals.

open_station_browser(): A simple method which opens the Trimet web page in the user's default web browser to find station id for any location. The API for finding station id from address has not been made available by TriMet. Later, the user will use this station id to search for arrivals data for the station of his choice.

Effort and Challenges

The project took us around 7 days to plan, implement, and test the application. One of the main challenges we faced was working with tkinter. We haven't used it before and it was a challenge to correctly format it. We also worked on getting the most out of the Trimet API by also including the current status feature in the application.

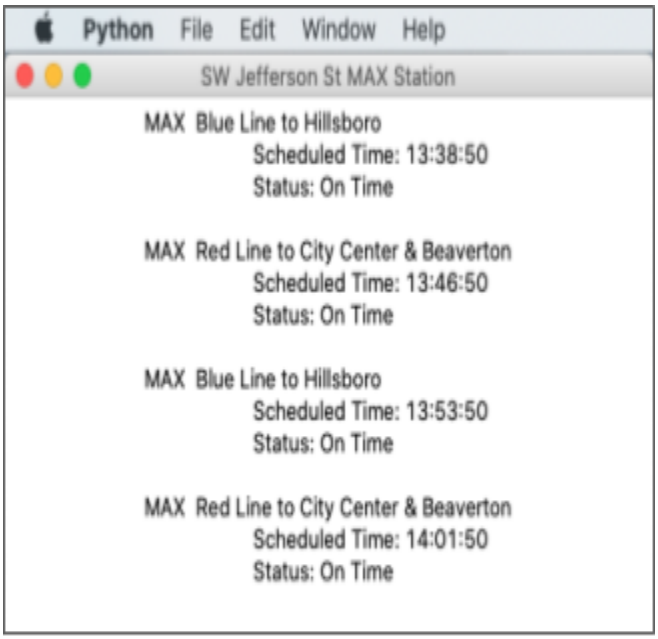
We haven't used any external code templates, modules, or projects. The work is purely our own.

Future Work

Given more time, we would like to experiment with other Python GUI frameworks such as Kivy or QT. We would like to create user interfaces that are more in line with the applications being built in today's day and age. A good project can also be to create a Python Django project and host it on a website to help students get arrival information on the most frequented stops in a single click. In addition to being a desktop application, it would be interesting to see how a Python android app could be built which provides the same functionality.

Results:

Landing Page



Results displayed for
Stop 10117

References:

1. https://docs.python.org/3/library/tkinter.html
2. https://docs.python.org/2/library/webbrowser.html
3. https://docs.python.org/3/library/urllib.request.html#module-urllib.request