Progress Report 4

Group 3: Nanfwang Dawurang, Abigail Dina, Cameron Wilson, David Bogunjoko Benjamin

After meeting with Amen Ra from Inrupt, it’s clear that the main focus should be how we would collect the housing data and get as many data points as possible. For this, we first would have to gather different databases that are open to the public and see which house properties we would want to include when we aggregate the data.

We were able to find and retrieve the general data sets with various attributes on visual data from the Open Baltimore website:

<https://data.baltimorecity.gov/datasets/real-property-information-2/explore?location=39.284833%2C-76.620485%2C12.71&showTable=true>

We also found another data set from Open Baltimore on building permits which would give us extra information on buildings regarding permits and other useful information

<https://data.baltimorecity.gov/datasets/baltimore::housing-and-building-permits-2019-present/explore?location=39.286072%2C-76.620500%2C11.93&showTable=true>

On seeing this, our task was then to retrieve the data from the API, and Open Baltimore made use of the ArcGIS REST API, so we had to read the documentation and figure out how querying this particular API works, and the various fields in the response. The link to the API documentation is here:

<https://developers.arcgis.com/rest/services-reference/enterprise/query-feature-service-layer-.htm#>

We wrote a Python program to interact with the REST API and store the rows of the data in our program. We made use of the *requests* Python library to make network calls, and we were to write a class to manage transferring the data to an excel sheet using the *openpyxl* library.

Here is a snippet of the data captured in our program terminal:



This data was parsed and we were able to receive just the fields that we needed.

Next Steps:

Our next step would be to transfer the data to an excel worksheet which we already started writing the class for, and would be to find more datasets and incorporate them into our code. Code will also be uploaded to Github, but for now here is a snapshot of it:

import requests

from openpyxl import Workbook

OPEN\_BALTIMORE\_URL = "https://geodata.baltimorecity.gov/egis/rest/services/CityView/Realproperty\_OB/FeatureServer/0/query?where=1%3D1&outFields=PIN,LOT\_SIZE,SALEDATE,OWNER\_1,OWNER\_2,OWNER\_3,FULLADDR,ZIP\_CODE,NEIGHBOR,YEAR\_BUILD,EXTD\_ZIP,BLOCK,PERMHOME,VACIND&outSR=4326&f=json"

class NetworkManager:

def \_\_init\_\_(self, url: str):

self.url = url

self.data = None

def fetchData(self) -> None:

self.performRequest(self.url)

def performRequest(self, url: str) -> None:

response = requests.get(url)

if response.status\_code == 200:

self.data = response.json()

self.handleRequestSuccess()

else:

self.handleRequestFailure()

def handleRequestSuccess(self):

print(self.data)

def handleRequestFailure(self):

pass

class WorksheetWriter:

def \_\_init\_\_(self):

self.wb = Workbook()

self.workSheet = None

self.createWorksheet()

def createWorksheet(self) -> None:

self.workSheet = self.wb.create\_sheet("Building Data")

def writeDataToWorksheet(self, data) -> None:

pass

if \_\_name\_\_ == "\_\_main\_\_":

networkManager = NetworkManager(OPEN\_BALTIMORE\_URL)

networkManager.fetchData()

Other Useful Links:

<https://technology.baltimorecity.gov/guidelines-using-open-baltimore-data>

<https://cels.baltimorehousing.org/codemapv2/>

<https://realpython.com/api-integration-in-python/>