

Retrieving Building Snapshot ID

SEED project

December 3, 2015

Contents

1	Introduction	2
2	Outline	2
2.1	With template excel file	2
2.1.1	Pseudo Code	2
2.1.2	Diagram	3
2.2	With PM excel file	4
2.2.1	Pseudo code	4
2.2.2	Diagram	5
3	Discussion	6

1 Introduction

The document records the process of retrieving the building snapshot id and canonical building id from SEED database.

2 Outline

2.1 With template excel file

2.1.1 Pseudo Code

User:

Fill out the PM template table with their building information

Upload tables some remote folder (for us to download)

Upload tables to SEED database (making sure there is an ID created)

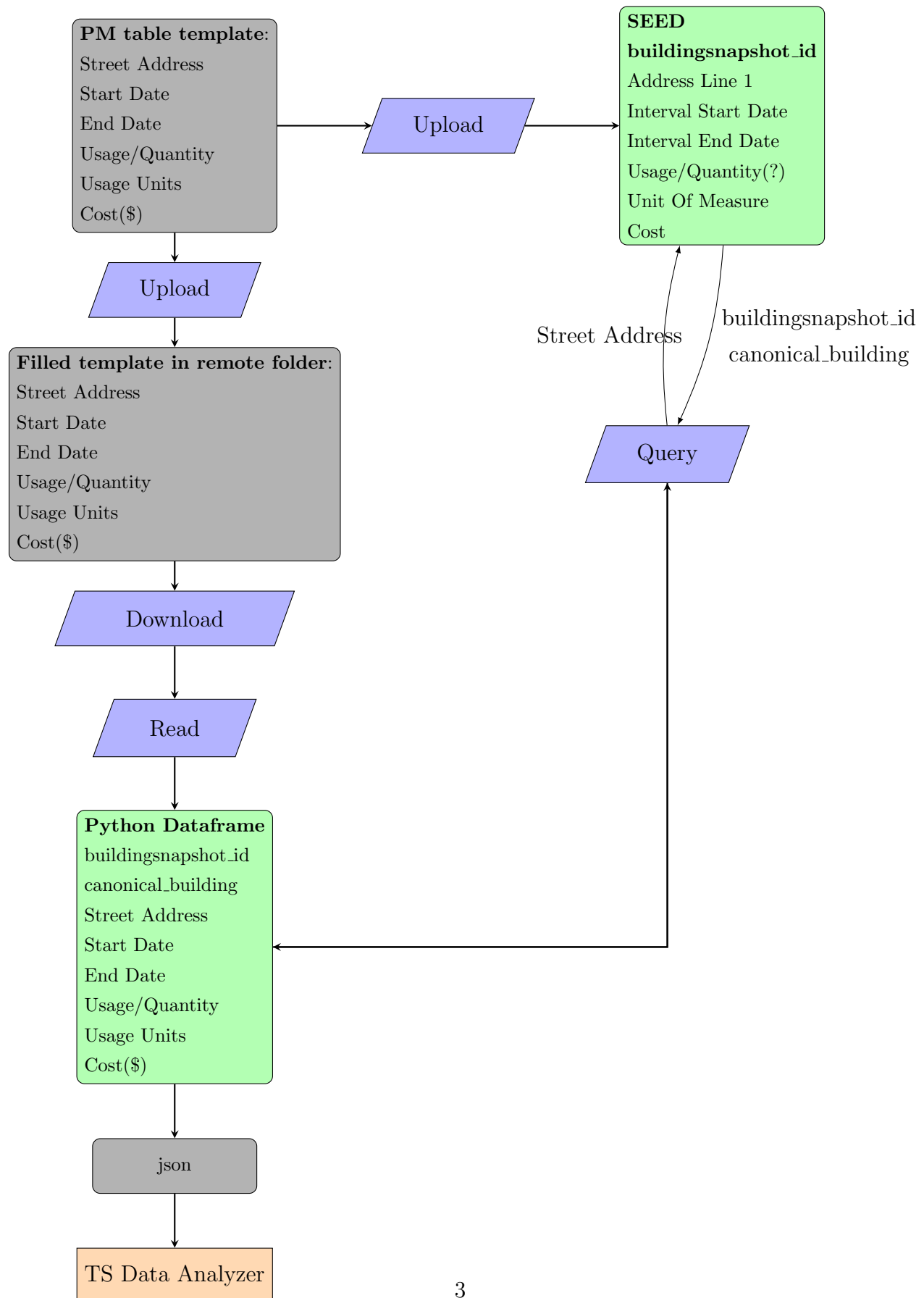
We:

Download the remote folder that contains the tables with the specified format.

For each table in the folder:

```
    Read the table into a data frame (Python pandas dataframe)
    //note: since the table is not uploaded to EnergyStar PM,
    //the Portfolio Manager ID might not be available
    Get a set (dictionary key), A, of addresses from the PM table
    Create an empty dictionary D (initializing all entries to None)
        for each address a in the set A:
            make a QUERY to SEED database with (address,
                                                Portfolio Manager ID)
                if address identifies a building b is in SEED database
                    //SEED.BuildingSnapshot.id
                    get the building snapshot id: s_id
                    make a QUERY with s_id to get c_id
                    //SEED.BuildingSnapshot.canonical_building
                    D[a] = {buildingsnapshot_id: s_id,
                           canonical_building: c_id}
            else
                D[a] = None
    Insert the c_id, s_id to the dataframe
    convert the dataframe to json format
```

2.1.2 Diagram



2.2 With PM excel file

2.2.1 Pseudo code

User:

Fill out EnergyStar PM table with their building information

Download the excel from EnergyStar

Upload tables to SEED database (making sure there is an ID created)

We:

Download the remote folder that contains the tables with the specified format.

For each table in the folder:

 Read the table into a data frame (Python pandas dataframe)

 Get a set (dictionary key), A, of ...

 (addresses, Portfolio Manager ID) pairs from the PM table

 Create an empty dictionary D (initializing all entries to None)

 for each address a in the set A:

 make a QUERY to SEED database with (address, Portfolio Manager ID)

 if (address, Portfolio Manager ID) identifies ...

 a building b is in SEED database

 get the building snapshot id: s_id

 make a QUERY with s_id to get the c_id

 //(SEED.BuildingSnapshot.canonical_building)

 D[a] = {buildingsnapshot_id: s_id,
 canonical_building: c_id}

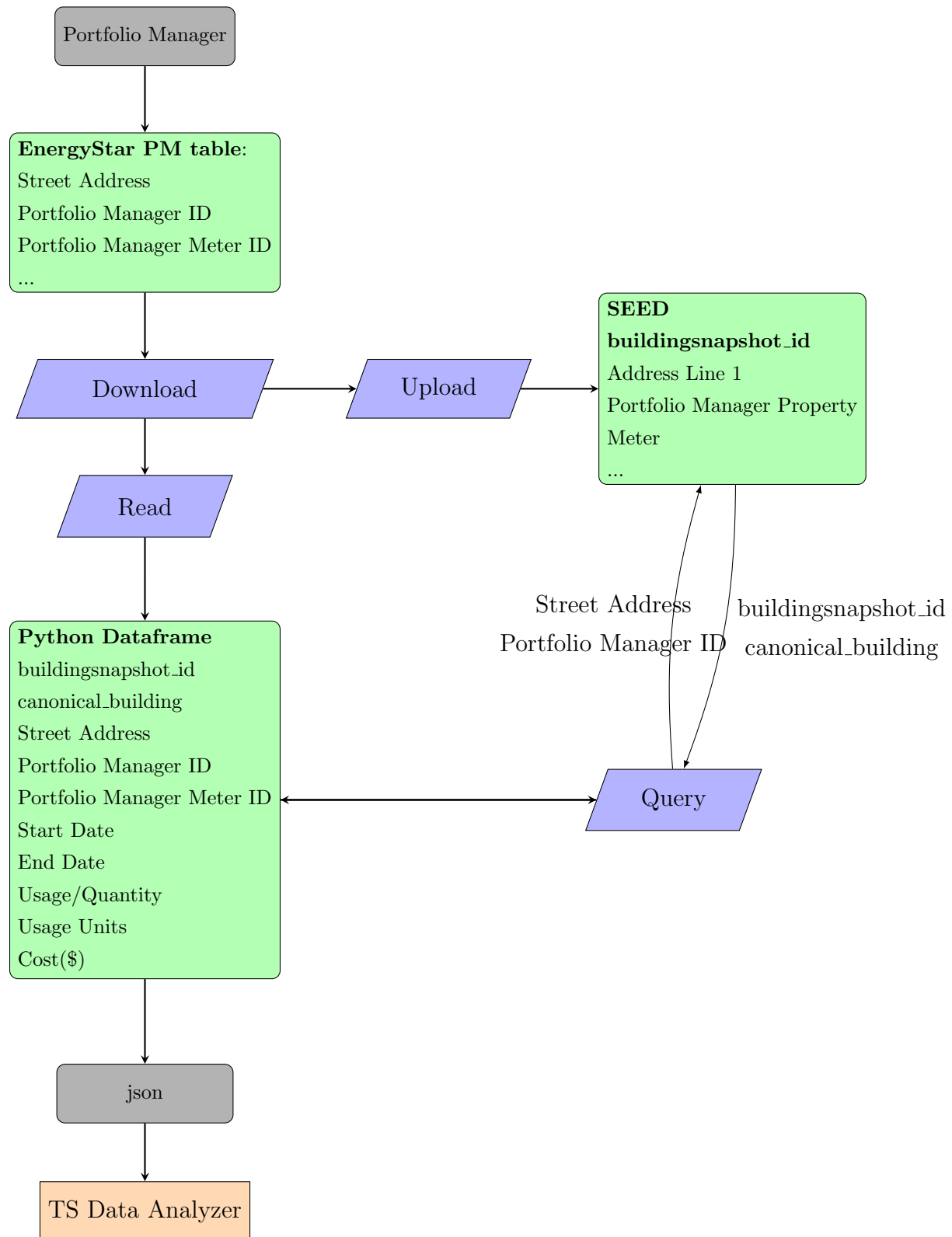
 else

 D[a] = None

Insert the c_id, s_id to the dataframe

convert the dataframe to json format

2.2.2 Diagram



3 Discussion

- Considering there might be buildings using the same address (one case is for *maybe all buildings in CMU are entered with the same “5000 Forbes Ave” address*) choose both “Address 1” and “Portfolio Manager ID” for retrieving the PM ID (Figure 1)

A	B	C	D	E
My Portfolio: yujie xu				
12/03/2015 10:27 AM EST				
Total Properties: 2				
Property Name	Portfolio Manager ID	Street Address	Street Address 2	City/Municipality
OSHER extension	4674008	5000 Forbes Ave		Pittsburg
test_hotel	4697182	5000 Forbes Ave		Pittsburg

Figure 1: PM table that has a same address but different Portfolio Manager ID

- using “None” rather than “-1” is because of this documentation: “None is frequently used to represent the absence of a value”
- **Question:** Is it better to directly read from PM table rather than create a template? because when people uploaded the table to PM, the “Portfolio Manager ID” is generate