Contents

[บทที่ 1 1](#_Toc157428297)

[หัวข้อที่ 1 1](#_Toc157428298)

[หัวข้อที่ 2 1](#_Toc157428299)

[บทที่ 2 2](#_Toc157428300)

[บทที่ 3 2](#_Toc157428301)

[บทที่ 4 2](#_Toc157428302)

# บทนำ บทที่ 1

## หัวข้อที่ 1

## หัวข้อที่ 2

# การจัดทำรายงานค่าใช้จ่ายของ NT

# บทที่ 2

## การ Load ข้อมูล จาก SAP Hana, ECC

## การตรวจ.....

## UPLOAD

## การตรวจ

ความต้องการขั้นพื้นฐานมีอะไรบ้าง

1. โปรแกรม thonny
2. Python version 3.10 ขึ้นไป
3. ติดตั้ง Library
   1. Pandas
      1. ติดตั้งด้วยคำสั่ง pip install pandas
   2. Openxlxxxxxxxx

การใช้โปรแกรม program1.py

โปรแกรมนี้คืออะไร

ใช้ทำอะไร

ก่อนใช้ต้องเตรียมข้อมูลอะไรบ้าง

ข้อมูลเก็บอยู่ที่ไหน

ทำแล้วได้อะไรออกมา

บทที่ 3

Tableau

Expense Report.twb

Data source:

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Master fie(s):

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

Relation:

ข้อ 2.1

ข้อ 2.2

# บทที่ 3

# บทที่ 4

ภาคผนวก

Source program.py <https://github.com/kaebmoo/bot_nt/blob/main/expense/expense_mapping_cc.py>

import firebase\_admin

from firebase\_admin import credentials

from firebase\_admin import firestore

from google.cloud.firestore\_v1.base\_query import FieldFilter    # https://github.com/GoogleCloudPlatform/python-docs-samples/blob/main/firestore/cloud-client/snippets.py

import json

import pandas as pd

cred = credentials.Certificate("/Users/seal/Documents/GitHub/questionnaire/survey\_data/serviceAccountKey.json")

firebase\_admin.initialize\_app(cred)

db = firestore.client()

def data\_tofile():

    # for doc in docs:

    #     print(f"{doc.id} => {doc.to\_dict()}")

    # https://firebase.google.com/docs/firestore/query-data/get-data#get\_all\_documents\_in\_a\_collection

    # Get all documents in a collection

    docs = db.collection("survey").stream()

    for doc in docs:

        print(f"{doc.id} => {doc.to\_dict()}")

    print()

    data\_file = {}

    docs = db.collection("survey").stream()

    for doc in docs:

        # https://estuary.dev/firestore-to-json/

        data\_file[doc.id] = doc.to\_dict()

        # data\_json = json.dumps(str(doc.to\_dict()).replace("'", '"'), separators=(',', ':'))

        data\_json = str(doc.to\_dict()).replace("'", '"')

        print(data\_json)

        data = json.loads(data\_json)

        print(data["dashboard"]["ท่านเคยเข้าดูรายงานหรือไม่"])

    with open("/Users/seal/Documents/GitHub/questionnaire/survey\_data/survey\_data.json", "w", encoding="utf8") as fp:

        # fp.write("\n".join(data\_file))

        json.dump(data\_file, fp, indent=4, ensure\_ascii=False)

# https://stackoverflow.com/questions/14048948/how-to-find-a-particular-json-value-by-key

def find\_json\_values(id, json\_repr):

    results = []

    def \_decode\_dict(a\_dict):

        try:

            results.append(a\_dict[id])

        except KeyError:

            pass

        return a\_dict

    json.loads(json\_repr, object\_hook=\_decode\_dict) # Return value ignored.

    return results

# https://firebase.google.com/docs/firestore/query-data/get-data#get\_a\_document

# Get a document

doc\_ref = db.collection("survey").document("032")

doc = doc\_ref.get()

if doc.exists:

    print(f"Document data: {doc.to\_dict()}")

else:

    print("No such document!")

print("walking into dashboard")

#

# Note: Use of CollectionRef stream() is prefered to get()

docs = (

    db.collection("survey").stream()

    # .where(filter=FieldFilter("capital", "==", True))

)

data\_file = {}

for doc in docs:

    data\_file[doc.id] = doc.to\_dict()

data\_str = json.dumps(data\_file, indent=4, ensure\_ascii=False)

print(data\_str)

data = json.loads(data\_str)

# print(data["2f1c"])

df = pd.DataFrame()

df\_report\_count = pd.DataFrame()

df\_report\_comment = pd.DataFrame()

for id in data:

    print(id)

    print(id, ":", data[id]["dashboard"]["ท่านเคยเข้าดูรายงานหรือไม่"])

    if len(id) > 4 or len(id) < 4:

        continue

    # new\_row = pd.DataFrame({'survey\_id': [id]})

    # df = pd.concat([df, new\_row]).reset\_index(drop=True)

    if data[id]["dashboard"]["ท่านเคยเข้าดูรายงานหรือไม่"] == "True":

        print(id, ":", data[id]["dashboard"]["ท่านเข้าดูรายงานใดบ้างบน Dashboard"])

        report\_list = data[id]["dashboard"]["ท่านเข้าดูรายงานใดบ้างบน Dashboard"]

        for report in report\_list:

            print(report, ":", data[id]["dashboard"][report] )

            report\_tab = data[id]["dashboard"][report]

            # นับจำนวน report ของแต่ละแบบสอบถาม

            new\_row\_report = pd.DataFrame({'survey\_id': [id], "report": [report]})

            df\_report\_count = pd.concat([df\_report\_count, new\_row\_report]).reset\_index(drop=True)

            for tab in report\_tab:

                # report column with tab data

                # ใส่ข้อมูล tab แต่ละ report

                print("tab", ": ", tab)

                if tab == "Download File":

                    print("Download: ", data[id]["dashboard"]["ความยากง่ายในการนำข้อมูล Excel ใน Download File Tab ไปใช้งาน"])

                    download\_score = data[id]["dashboard"]["ความยากง่ายในการนำข้อมูล Excel ใน Download File Tab ไปใช้งาน"]

                    new\_row\_report\_tab = pd.DataFrame({"survey\_id": [id], "report": [report], "tab" : [tab], "score":[download\_score]})

                elif tab == "Database Export":

                    print("Export: ", data[id]["dashboard"]["ความยากง่ายในการเรียกข้อมูลในรายงาน Database Export Tab"])

                    export\_score = data[id]["dashboard"]["ความยากง่ายในการเรียกข้อมูลในรายงาน Database Export Tab"]

                    new\_row\_report\_tab = pd.DataFrame({"survey\_id": [id], "report": [report], "tab" : [tab], "score":[export\_score]})

                else:

                    new\_row\_report\_tab = pd.DataFrame({"survey\_id": [id], "report": [report], "tab" : [tab], "score" : [0]})

                df = pd.concat([df, new\_row\_report\_tab]).reset\_index(drop=True)

                # print(df)

        comment\_list = ["ข้อเสนอแนะ", "ท่านต้องการให้เพิ่มเติมรายงานในมุมมองใด", "ท่านต้องการให้ปรับปรุงอะไรใน Database Export Tab", "ท่านต้องการให้ปรับปรุงอะไรใน Excel Report"]

        for comment\_index in comment\_list:

            if comment\_index in data[id]["dashboard"]:

                print(comment\_index, ": ", data[id]["dashboard"][comment\_index])

                comments = data[id]["dashboard"][comment\_index]

                new\_row\_report\_comment = pd.DataFrame({"survey\_id": [id], "comment":[comment\_index], "comment\_value":[comments], "flag":[True]})

                df\_report\_comment = pd.concat([df\_report\_comment, new\_row\_report\_comment]).reset\_index(drop=True)

    elif data[id]["dashboard"]["ท่านเคยเข้าดูรายงานหรือไม่"] == "False":

        print(id, ":", data[id]["dashboard"])

        # https://pynative.com/python-check-if-key-exists-in-json-and-iterate-the-json-array/

        if "ข้อเสนอแนะ" in data[id]["dashboard"]:

            print("ข้อเสนอแนะ: ", data[id]["dashboard"]["ข้อเสนอแนะ"])

            comments = data[id]["dashboard"]["ข้อเสนอแนะ"]

            new\_row\_report\_comment = pd.DataFrame({"survey\_id": [id], "comment":["ข้อเสนอแนะ"], "comment\_value":[comments], "flag":[False]})

            df\_report\_comment = pd.concat([df\_report\_comment, new\_row\_report\_comment]).reset\_index(drop=True)

        if "ท่านต้องการให้เพิ่มเติมรายงานในมุมมองใด" in data[id]["dashboard"]:

            print("ท่านต้องการให้เพิ่มเติมรายงานในมุมมองใด: ", data[id]["dashboard"]["ท่านต้องการให้เพิ่มเติมรายงานในมุมมองใด"])

            comments = data[id]["dashboard"]["ท่านต้องการให้เพิ่มเติมรายงานในมุมมองใด"]

            new\_row\_report\_comment = pd.DataFrame({"survey\_id": [id], "comment":["ท่านต้องการให้เพิ่มเติมรายงานในมุมมองใด"], "comment\_value":[comments], "flag":[False]})

            df\_report\_comment = pd.concat([df\_report\_comment, new\_row\_report\_comment]).reset\_index(drop=True)

        # print(find\_json\_values("ข้อเสนอแนะ", json.dumps(data[id]["dashboard"])))

        # print(find\_json\_values("ท่านต้องการให้เพิ่มเติมรายงานในมุมมองใด", json.dumps(data[id]["dashboard"])))

print(df)

print(df\_report\_count)

df.to\_csv("/Users/seal/Documents/GitHub/questionnaire/survey\_data/survey\_data.csv", index=False)

df\_report\_count.to\_csv("/Users/seal/Documents/GitHub/questionnaire/survey\_data/survey\_report.csv", index=False)

df\_report\_comment.to\_csv("/Users/seal/Documents/GitHub/questionnaire/survey\_data/survey\_report\_comment.csv", index=False)

'''

# List subcollections of a document

collections = db.collection("survey").document("2f1c").collections()

for collection in collections:

    for doc in collection.stream():

        print(f"{doc.id} => {doc.to\_dict()}")

'''

''' ชื่อ ต้องเป็น

        Must separate field names with a single period (.)

        May be passed as a string when all field names in the path are simple, otherwise must be passed as a FieldPath object (e.g. JavaScript FieldPath)

        A simple field name is one where all of the following are true: Contains only the characters a-z, A-Z, 0-9, and underscore (\_)

        Does not start with 0-9

# Note: Use of CollectionRef stream() is prefered to get()

# error

docs = (

     db.collection("survey")

     .where(filter=FieldFilter("ท่านเคยเข้าดูรายงานหรือไม่", "==", data["ท่านเคยเข้าดูรายงานหรือไม่"]))

     .stream()

)

# docs = db.collection("survey").where("ท่านเคยเข้าดูรายงานหรือไม่", "==", True).get()

for doc in docs:

    print(f"{doc.id} => {doc.to\_dict()}")

'''