Canvas Install and Set-Up Instructions

Requirements

- Canvas Admin Account login information
- template_password_file.txt
- new_refresh_token.py
- create_tables.py
- get_historical_data.py
- main.py
- automate_me.bat

Part One: Get Canvas Information and Install Python

1. Download and install python 3.8 here
    a. Scroll to the bottom of the page and click on "Windows x86-64 executable installer"
    b. Run the .exe file and on the first screen MAKE SURE TO CLICK the box that says "Add Python 3.8 to PATH". All other options can be left to default.
2. Create a folder in Documents with the name 'Canvas' and put all the files in the same folder
3. Get values for these variables (link to canvas instructions):
    a. Client Secret
    b. Client ID
    c. Login URL
    d. Redirect URI
    e. Token URL
4. Edit the template_password_file.txt and make sure to fill all the values, but leave the refresh token an empty string. Save and note the new name of the password text file.
5. Open new_refresh_token.py in a text editor and change the variable password_file to the name of the password text file that was changed in the previous step
6. If python has finished installing. Open command prompt, type the following lines of code and hit enter after every line to execute the code
    a. pip install requests
    b. pip install requests_oauthlib

c. pip install pyperclip
d. pip install pyodbc
e. pip install sqlalchemy
f. cd Documents\Canvas
g. python new_refresh_token.py
    i. Must be logged in as the canvas admin account for this step to work
    ii. To check if the code ran properly. Open the Canvas passwords .txt file previously made and a string of letters and numbers should be present next to 'Refresh Token'

Part Two: Create User Environment Variables and Get Refresh Tokens

1. Navigate to the windows environment variable settings and create these user variables
    a. Name: DB_HOST Value: host name of the database
    b. Name: DB_PASSWORD Value: password to the database
    c. Name: DB_PORT Value: port number
    d. Name: DB_ USERNAME Value: username to database
    e. Name: DB_ADDRESS Value: the email address that sends error emails
    f. Name: DB_PASSWORD Value: the password to the email address account
    g. Note: The email account should be a Microsoft account otherwise research will be needed to set the correct server name and port

(Optional) Part Three: Use python to create the tables in EK12 Database

1. Open create_tables.py in a text editor and set the DATABASE variable to a string of the name of the desired school's database. NOTE: If a variable is in all caps that is the convention for declaring a variable that is a constant and the value of it never changes throughout the code. If you decide to change the variable name, make sure to change all instances of the variable otherwise the code will error
2. Open command prompt, type the following lines of code and hit enter after every line to execute the code
    a. cd Documents\Canvas
    b. python create_tables.py

Part Four: Run and Test Code

1. Open main.py in a text editor change the values of the variables. DO NOT CHANGE ANY VARIABLE NAMES AND ALL VARIABLES ARE STRINGS AND MUST BE SURROUNDED BY QUOTES. **NOTE**: If a variable is in all caps that is the convention for declaring a variable that is a constant and the value of it never changes throughout the code. If you decide to change the variable name, make sure to change all instances of the variable otherwise the code will error
    a. FIRST_DAY_OF_SCHOOL
        i. Should be formatted as 'YYYY-MM-DD'
    b. RECIPIENTS
        i. Don't set these variables until after testing code, otherwise the recipients' inboxes may get spammed with error messages
        ii. Lists are denoted by brackets [] and every item in the list is separated by a comma. In the case of the recipients variable, every item in the list represents a different person, with the information about that person stored as a dictionary data type.
        iii. Dictionaries contain "keys", "value" pairs and are denoted with curly braces {}. Calling the name of the key returns the value associated with the key. In our case, every dictionary has three keys: 'address', 'subject', 'message' with their respective values. The colon denotes a pairing between a key and value and the commas separate the key/value pairs from one another. ALL KEYS/VALUES MUST ALL EXIST EVEN IF YOU LEAVE THE SUBJECT LINE TO BE AN EMPTY STRING. All values are strings.
        iv. When adding new people to the recipients variable, make sure all dictionaries are enclosed in curly braces and separated by commas
        v. When deleting recipients, make sure to remove trailing commas
            1. Ex: recipients = [person1, person2 ,] (The last comma will cause an error)
    c. DATABASE_NAME
        i. Name of school's database in EK12 database
    d. MAIN_URL
        i. this is the api url
        ii. EX: https://schoolname.instructure.com/api/v1
    e. CANVAS_PASSWORD_FILE_NAME
        i. Name of the text file where canvas credentials are stored

2. Open automate_me.bat with a text file editor. Edit where the code says <name of the user> and save the file
3. Make sure the tables are created in the EK12 database with the correct column names before running the code to retrieve the data.
4. For testing purposes, change the variable FIRST_DAY_OF_SCHOOL in main.py to a recent day. Double click the .bat file to run the main.py program. Check for errors. Runtime will be at least an hour
5. If no errors automate .bat file in the task scheduler
    a. Go here for those instructions or simply google "automate bat file in windows scheduler"

Part Five: Getting Previous Years' Data

1. Simply edit and run the get_historical_data.py just like the main.py. The only true difference between these two scripts is that the get_hostorical_data.py script has a THRESHOLD_DATE variable rather than a FIRST_DAY_OF_SCHOOL variable. The only coding difference is that the get_historical_data.py will delete all data previously existing in all tables and load Canvas data after the THRESHOLD to the current date. Run the get_hostorical_data.py yearly and after, run the main.py daily.

For more help, email ***clarguerwill@gmail.com*** :)