|  |
| --- |
| # connecting to neo4j, performing queries and sending automated email. |
|  |  |
|  | get\_ipython().system('pip install neo4j') |
|  |  |
|  | from neo4j import GraphDatabase |
|  | import datetime |
|  | import pandas as pd |
|  | import smtplib |
|  | from email.mime.text import MIMEText |
|  | from email.mime.multipart import MIMEMultipart |
|  | from email.mime.base import MIMEBase |
|  | from email import encoders |
|  |  |
|  |  |
|  | def fetch\_customer\_details(): |
|  | # connection with neo4j |
|  | crs = GraphDatabase.driver(uri="bolt://localhost:11005") |
|  |  |
|  | # creating a session for command execution |
|  | s = crs.session() |
|  |  |
|  | # Query 1: to fetch the list of users to whom we can sent mail |
|  | q1 = "MATCH (h:HomeCity)-[way:TRAVEL\_ROUTE]->(m:DestinationCity) |
|  | OPTIONAL MATCH |
|  | (p:Rate\_Slab)-[t:Rate]->(u:User) |
|  | RETURN h.CityName as Homecity,m.CityName as DestinationCity,toInteger(u.NumOfPasngrs) as NumOfPasngrs, |
|  | u.TripType as TripType,u.CarType as CarType,round(way.travelDistance/1000,2) as DistanceKM,t.Expected\_Expense as Expected\_Expense,count(\*) as NumOfUsers |
|  | " |
|  | nodes = s.run(q1) |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | for j in nodes2: |
|  | print("the users list to send notification are:\n\n{}".format(j)) |
|  |  |
|  | # convert the neo4j object to dataframe and export the file. |
|  | data = data = nodes2.data() |
|  | df = pd.DataFrame(data) |
|  | df |
|  | df.to\_csv(r'C:\Users\Acer\Desktop\df.txt', index=None, sep=' ', mode='a') |
|  |  |
|  |  |
|  | fetch\_user\_details() |
|  |  |
|  |  |
|  | def send\_email(): |
|  | # the employee email ID - sender , LegalTeam email ID - receiver. |
|  | sender\_email = '<employee@carrental.com>' |
|  | email\_password = '<123456>' |
|  | receiver\_email = '<fos\_team@carpool.com>' |
|  |  |
|  | subject = 'Recommenation for pooling' |
|  |  |
|  | msg = MIMEMultipart() |
|  | msg['From'] = sender\_email |
|  | msg['To'] = receiver\_email |
|  | msg['Subject'] = subject |
|  |  |
|  | # Body of the email |
|  |  |
|  | body = 'Please find the list of users available for pooling matching same criteria and the expected expense' |
|  | msg.attach(MIMEText(body, 'plain')) |
|  |  |
|  | # attaching the customer list |
|  | filename = 'C:/Users/Nissy/Desktop/User.txt' |
|  | attachment = open(filename, 'rb') |
|  |  |
|  | part = MIMEBase('application', 'octet-stream') |
|  | part.add\_header('Content-Disposition', "attachment; filename= " + filename) |
|  |  |
|  | msg.attach(part) |
|  | text = msg.as\_string() |
|  |  |
|  | # the hostname and port number of carrental service should be updated below |
|  | server = smtplib.SMTP('smtp.gmail.com', 587) |
|  | server.starttls() |
|  |  |
|  | # senders email autentication |
|  | server.login(sender\_email, email\_password) |
|  |  |
|  | # send mail block. |
|  | server.sendmail(sender\_email, receiver\_email, text) |
|  | server.quit() |
|  |  |
|  |  |
|  | # callling the send email function |
|  |  |
|  | send\_email() |
|  | del df |

* © 2022 GitHub, Inc.
* [Ter](https://docs.github.com/en/github/site-policy/github-terms-of-service)