Data Engineering

Reading Data into Python

June, 2018

GSA

D2D
DATA TO DECISIONS

Goal For Today:

- Finish Reading Data from APIs
- Reading Data from Databases
- Reminder: CSV, TSV…etc
  - And Excel files

GSA

D2D
DATA TO DECISIONS

Using APIs – Test Case

- Goal: Write a python program that uses the WMATA API to get train predictions for a given station

  - ➢Lines will be in a CSV file

  - ➢Use Pandas Data frames

  - ➢User should input a Line and a Station, and the program should display a list of train predictions from WMATA's API

GSA

D2D
DATA TO DECISIONS

Using APIs – Test Case

- Code:

```
import http, pandas
headers = {'api_key': xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx'}
df_csv = pandas.read_csv("File_Location\\Lines.csv")
print(df_csv['LineCode'])
lineColor = input("Input a line colour from the list above:")
conn = http.client.HTTPSConnection('api.wmata.com')
conn.request("GET", "/Rail.svc/json/jStations?LineCode="+lineColor, "{body}", headers)
response = conn.getresponse()
data = response.read()
df = pandas.read_json(data)
for index, row in df.iterrows():
    print(row['Stations']['Code'] +"-" +row['Stations']['Name'])
stationCode = input("Input a stationCode from the list above:")
conn.request("GET", "/StationPrediction.svc/json/GetPrediction/"+stationCode, "{body}", headers)
response = conn.getresponse()
data = response.read()
df = pandas.read_json(data)
print("Here are your Train Predictions:")
for index, row in df.iterrows():
    print(row['Trains']['DestinationName'] +"["+row['Trains']['Line']+"] - " +row['Trains']['Min'] +" Minutes")
conn.close()
```

GSA

D2D
DATA TO DECISIONS

Using APIs – Test Case

Additional Notes:

- Check the Request status using HTTP Codes (getcode method)
  - 200 → OK
  - 401 → Unauthorized
  - 403 → Forbidden
  - 503 → Service Unavailable
  - Useful for Error handling
- Can we make this more user friendly?
  - Try to Improve on the code
  - Example: Enter number for station instead of code

GSA

D2D
DATA TO DECISIONS

Reading Data from Databases

Reading Data from Databases

Databases:

- Collection of data
- Stored in a way that makes its access and management easy
- We will be discussing relational databases
  - ➢ Data stored in tables
  - ➢ Tables are made of columns and rows
  - ➢ Tables have relations
- We will be discussing reading data into python
  - ➢ You can issue all DDL and DML statements in python to!
- Even though it is doable, you better know your schema !
- We will be using MySQL – Same Logic applies to other relational DBs

GSA

D2D
DATA TO DECISIONS

Reading Data from Databases

To connect to a Database you will need:

- DB engine
- Hostname (or IP address)
- Port
- Username
- Password
- Database/Schema
- Other optional or DB specific Attributes
  - ➢ Character Set
  - ➢ Domain (SQL Server Only – to use Windows authentication)
  - ➢ … etc

GSA

D2D
DATA TO DECISIONS

Reading Data from Databases

To connect to a MySQL DB, we will use the **pymysql** package (is it installed?)

End Goal: Read the data from a table into python

- **Pseudo Code:**
  - Define Connection Parameters
  - Establish a Connection
  - Execute A Query with a cursor
    - A control structure that allows the traversal over the records in a database table
  - Loop over the cursor to display the results

GSA

D2D
DATA TO DECISIONS

Reading Data from Databases

Code:

```
import pymysql
# open connection to the database
conn = pymysql.connect(host='XXXXXXXXXXXX',
                       port=3306,
                       user='my_user',
                       passwd='YYYYYYYY',
                       db='MY_DB',
                       charset='utf8')
#Defining Cursor on the connecion
cur = conn.cursor()
cur.execute("SELECT * FROM My_Table" )
data = cur.fetchall()
#Loop over the result set and print record
for i in data:
    print(i)
# close connection to the database
cur.close()
conn.close()
```

GSA

D2D
DATA TO DECISIONS

Reading Data from Databases

…. Or Use Pandas!

- Pandas.read_sql()
- Join data in the DB or in Python
- Same data structure as other Sources
- All benefits of pandas

GSA

D2D
DATA TO DECISIONS

Reading Data from Databases - Pandas

Code:

```
import pymysql
import pandas

# open connection to the database
conn = pymysql.connect(host='XXXX',port=3306,user='my_user',passwd='XXXX',db='my_db',charset='utf8')

df_db = pandas.read_sql('SELECT * FROM my_table',conn)
print(df_db.head(3))

# close connection to the database
conn.close()
```

GSA

D2D
DATA TO DECISIONS

Passing Database Credentials to Python

- **Problem:** Storing Credentials in reusable code is unsecure
- **Solution1:**Reading credentials from environment variables using **getenv**

```
import os
import pymysql
host= os.getenv("PYMSSQL_TEST_SERVER")
user = os.getenv("PYMSSQL_TEST_USERNAME")
password = os.getenv("PYMSSQL_TEST_PASSWORD")

conn = pymssql.connect(server, user, password, "MY_DB")
cur = conn.cursor()
cur.execute("SELECT * FROM My_Table" )
```

GSA

D2D
DATA TO DECISIONS

Passing Database Credentials to Python (Continued)

- **Problem:** Storing Credentials in reusable code is unsecure
- **Solution1:**Reading credentials from environment variables using **getenv**

Variables could be set from the OS or from python:

```
import os
def setEnvVariables():
 os.environ["PYMSSQL_TEST_SERVER"] = '4.16.4.16'
 os.environ["PYMSSQL_TEST_USERNAME"] = 'test_user'
 os.environ["PYMSSQL_TEST_PASSWORD"] = '4.16.4.16'
```

GSA

D2D
DATA TO DECISIONS

Passing Database Credentials to Python

- **Problem:** Storing Credentials in reusable code is unsecure
- **Solution2:** Create a Utility Module

```
import pymysql
def getConnection():
 connection = pymysql.connect(host='XXXXX',
                              user='test_user',
                              password='my_password',
                              db='test_db)
     return connection


# Use your utility module.
import MyConnectionUtil
connection = MyConnectionUtil.getConnection()
```

GSA

D2D
DATA TO DECISIONS

Passing Database Credentials to Python

- **Problem:** Storing Credentials in reusable code is unsecure
- **Solution3:** Use the **configparser** module

Config.txt

```
[MyDB]
UserName=myName
Password=myPassword
```

Python Code

```
import configparser
Import pymysql
config = configparser.ConfigParser()
config.read(' Config.txt ')
User_name = config['MyDB']['UserName']
Password = config['MyDB']['Password']
connection = pymysql.connect(host='XXXXX',
                             user=User_name,
                             password=Password,
                             db='test_db)
```

GSA

D2D
DATA TO DECISIONS

Reading Data from a CSV file

Reading Data from a CSV file

- df_csv = pandas.read_csv("C:\\DataFiles\\File.csv")

- You can specify the delimiter
  - ➢Pass a **sep='\t'** argument for example
  - ➢http://pandas.pydata.org/pandas-docs/version/0.23/generated/pandas.read_csv.html

- Source Does not have to be local:
  - ➢Starting Pandas 0.19.2 you can pass a URL straight to the read_csv function
    - pandas.read_csv("https://inventory.data.gov/dataset/04247624-1d6b-4e03-84eb-9eda1a6ea638/resource/63663b53-7cb6-4bea-bc4e-1e5897ef3158/download/datagovbldgrexus.csv")

GSA

D2D
DATA TO DECISIONS

Reading Data from an Excel file

- **Use** pandas.read_excel(“FileName.xlsx", sheetname=0)
  - ➢**OR** pandas.read_excel(“FileName.xlsx", sheetname=“sheetName”)

- http://pandas.pydata.org/pandas-docs/version/0.22/generated/pandas.read_excel.html

GSA

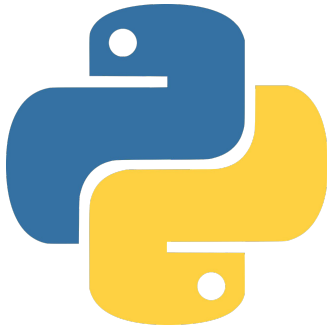
D2D
DATA TO DECISIONS

Questions?

GSA

D2D
DATA TO DECISIONS

Thank You



GSA

D2D
DATA TO DECISIONS