Step 3: Proof of concept connecting to SQL using pyodbc

Article • 11/18/2022

This example is a proof of concept. The sample code is simplified for clarity, and doesn't necessarily represent best practices recommended by Microsoft.

To get started, run the following sample script. Create a file called test.py, and add each code snippet as you go.

```
> python test.py
```

Connect

Python

```
import pyodbc
# Some other example server values are
# server = 'localhost\sqlexpress' # for a named instance
# server = 'myserver,port' # to specify an alternate port
server = 'tcp:myserver.database.windows.net'
database = 'mydb'
username = 'myusername'
password = 'mypassword'
# ENCRYPT defaults to yes starting in ODBC Driver 18. It's good to al-
ways specify ENCRYPT=yes on the client side to avoid MITM attacks.
cnxn = pyodbc.connect('DRIVER={ODBC Driver 18 for SQL
Server};SERVER='+server+';DATABASE='+database+';ENCRYPT=yes;
UID='+username+';PWD='+ password)
cursor = cnxn.cursor()
```

Run query

The cursor.execute function can be used to retrieve a result set from a query against SQL Database. This function accepts a query and returns a result set, which can be iterated

over with the use of cursor.fetchone().

Python

```
#Sample select query
cursor.execute("SELECT @@version;")
row = cursor.fetchone()
while row:
    print(row[0])
    row = cursor.fetchone()
```

Insert a row

In this example, you see how to run an INSERT statement safely, and pass parameters. The parameters protect your application from SQL injection.

Python

```
#Sample insert query
count = cursor.execute("""
INSERT INTO SalesLT.Product (Name, ProductNumber, StandardCost,
ListPrice, SellStartDate)
VALUES (?,?,?,?,?)""",
'SQL Server Express New 20', 'SQLEXPRESS New 20', 0, 0,
CURRENT_TIMESTAMP).rowcount
cnxn.commit()
print('Rows inserted: ' + str(count))
```

Azure Active Directory and the connection string

pyODBC uses the Microsoft ODBC driver for SQL Server. If your version of the ODBC driver is 17.1 or later, you can use the Azure Active Directory interactive mode of the ODBC driver through pyODBC.

This interactive option works if Python and pyODBC permit the ODBC driver to display the dialog. The option is only available on Windows operating systems.

Example connection string for Azure Active Directory

interactive authentication

The following example provides an ODBC connection string that specifies Azure Active Directory interactive authentication:

```
server=Server;database=Database;UID=UserName;
Authentication=ActiveDirectoryInteractive;Encrypt=yes;
```

For more information about the authentication options of the ODBC driver, see Using Azure Active Directory with the ODBC Driver.

Next steps

For more information, see the Python Developer Center .