Faversham Associates Ltd

Python and MySQL

Connecting to a MySQL database using mysqlclient

Version: 1.0.0

Gavin Bungay

gavin.bungay@favershamassociates.com

December 12, 2017

In previous versions the `MySQLdb` package was commonly used to connect to a MySQL Python database. This package does not work with Python 3.x; it has been replaced by the `mysqlclient` package, this a a fork from the orginal `MySQLdb` project and it is hoped that they will be reunited at some point in the future.

It was only by bitter experience that it was discovered that the `MySQLdb` package was incompatible and what follows is a condensed version of all the faffing about we had to go through before we got things to work.

Contents

1 Installing `mysqlclient`

In order for Python to connect to any database, it needs an interface. Different ones exist, here's a list: `'https://wiki.python.org/moin/DatabaseInterfaces'`.

In the past we used `MySQLdb`, it implements the Python Database API v2.0 and is built on top of the MySQL C API. Unfortunately it does not work with Python 3.x and above. Luckily there is a fork from the original MySQL-python project that has Python 3 support: `mysqlclient`.

Slightly ambiguously, and due to the above forking, the module name is still `MySQLdb`.

To check whether the MySQLdb module has been installed, try importing it:

```
1 import MySQLdb
```

Listing 1: Trying to import the MySQLdb module.

If the response is:

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-1-dd22983d5391> in <module>()
----> 1 import MySQLdb

ModuleNotFoundError: No module named 'MySQLdb'
```

then the module is not installed.

There is a lot of discusison online about how to install `mysqlclient`, people suggesting numerous and varied ways that they used to get the installation to succeed sucessfully. The method that (eventually) worked for us was:

```
$ sudo add-apt-repository ppa:jonathonf/python-3.6
$ sudo apt-get update
$ sudo apt-get install python3.6-dev libmysqlclient-dev
```

(We say 'eventually' as it took several hours of 'pushing buttons' to see what worked.)

2 Using `mysqlclient`

Once installed, using `mysqlclient` is fairly straightforward. First one creates a connection to the MySQL database, then a 'cursor object' that can be used to execute SQL statements.

To illustrate the above in action we are going to present some code that:

- connects to a MySQL server,
- creates a database,
- creates a table in that database,
- inserts some data in the table,
- queries the table.

2.1 Connecting to a MySQL Server

Once the `mysqlclient/MySQLdb` module has been imported, to connect to a MySQL server, use the `connect()` method it.

`con = MySQLdb.connect(⟨host⟩, ⟨user⟩, ⟨password⟩, ⟨database⟩)`

```
1  # Import the MySQL interface (it is from the mysqlclient package, the
2  #    module is named MySQLdb)
3  import MySQLdb as sql
4
5  # Create a connection to the MySQL db.
6  #    The syntax is
7  # con = MySQLdb.connect(<host>, <user>, <password>, <database>)
8  strHost = "localhost"
9  strUser = "gavin"
10 strPassword = "dontbecheeky"
11 strDB = ""  # Leave blank as the database might not exist as of yet
12
13 # Create a connection to the database
14 con =  sql.connect(strHost, strUser, strPassword, strDB)
```

Listing 2: Connecting to a MySQL Server.

2.2 Creating a Cursor

Once a connection has been established, use its `cursor()` method to create a cursor object. This cursor can thenm be used to execute SQL statements

`con = MySQLdb.connect(⟨host⟩, ⟨user⟩, ⟨password⟩, ⟨database⟩)`

```
1  # Create a Cursor object.
2  cur = con.cursor()
3
4  #  Play around with some SQL commands - set up the commands in a
5  #     string and then use the cursor object to execute them.
6
7  # 1, Create a database called "Mailer"
8  #    If it already exists, delete it
9  strSQL = "DROP␣DATABASE␣IF␣EXISTS␣Mailer;"
10 cur.execute(strSQL)
11
12 #    Create it
13 strSQL = "CREATE␣DATABASE␣Mailer"
14 cur.execute(strSQL)
15
16 # Switch to it
17 strSQL = "USE␣Mailer"
18 cur.execute(strSQL)
19
20 # 2, Add a table
21 #    If it already exists, delete it
22 strSQL = "DROP␣TABLE␣IF␣EXISTS␣Contacts"
23 cur.execute(strSQL)
24
25 #    Create it - a simple table called Contacts with 2 columns: Id, Name,
       email
26 strSQL = ("CREATE␣TABLE␣Contacts"
27           "("
28                 "Id␣INT␣PRIMARY␣KEY␣AUTO_INCREMENT,"
29                 "Name␣VARCHAR(50),"
30                 "Email␣VARCHAR(50)"
31           ")")
32 cur.execute(strSQL)
33
34 # 3, Add some data
35 strSQL = "INSERT␣INTO␣Contacts(Name,␣Email)␣VALUES('Fred',␣'fred@fredscompany.
       co.uk');"
36 cur.execute(strSQL)
37
38 strSQL = "INSERT␣INTO␣Contacts(Name,␣Email)␣VALUES('Bert',␣'bert@bertscompany.
       co.uk');"
39 cur.execute(strSQL)
40
41 strSQL = "INSERT␣INTO␣Contacts(Name,␣Email)␣VALUES('Fredrika',␣'
       fredrika@fredrikascompany.co.uk');"
42 cur.execute(strSQL)
43
44 strSQL = "INSERT␣INTO␣Contacts(Name,␣Email)␣VALUES('Bertha',␣'
       bertha@bertscompany.co.uk');"
```

```
cur.execute(strSQL)

# Commit the changes
con.commit()
```

Listing 3: Creating and using a cursor object.

2.3 Querying a Table

The Contacts table has had some data added, query it.

```
# 4, Select and go through the data we have just added
strSQL = "SELECT␣*␣FROM␣Contacts"
cur.execute(strSQL)

# 5, Iterate through the records
#    Firstly by column index
print("By␣Index:")
print("---------")
rows = cur.fetchall()
for row in rows:
    print ("Name:␣" + str(row[0]) + ",␣Email:␣" + str(row[1]))


#    Secondly, just for demonstration, by column name
print("\nBy␣Name:␣")
print("---------")

# We need to reselct the data
strSQL = "SELECT␣*␣FROM␣Contacts"
cur.execute(strSQL)
rows = cur.fetchall()
for row in rows:
    print ("Name:␣" + str(row[0]) + ",␣Email:␣" + str(row[1]))


# Close the connection
con.close()

print("Finished")
```

Listing 4: Creating and using a cursor object.

3 See Also

'http://www.favershamassociates.com/Documents/lamp.pdf - LAMP, Using Apache, MySQL and Python on Linux'

'http://www.favershamassociates.com/Documents/mysql.pdf - Using MySQL, A brief introduction'

'http://www.favershamassociates.com/Documents/Django.pdf - Django, Basic Installation using Python 3'

4 Copyright