**Project 1: Mongo DB**

Sarah Cooney, Mingyaun Li, Jason Meng

# **1. Importing to a Database in MongoDB**

The first step was setting up the database in MongoDB. We downloaded the zipcodes.json file and used the import command to create the database and upload the data. Figure 1 shows a sample of data from the file. Figure 2 shows the import command used to create the database. The mongod command must be run to establish a connection to the server before running the import command.

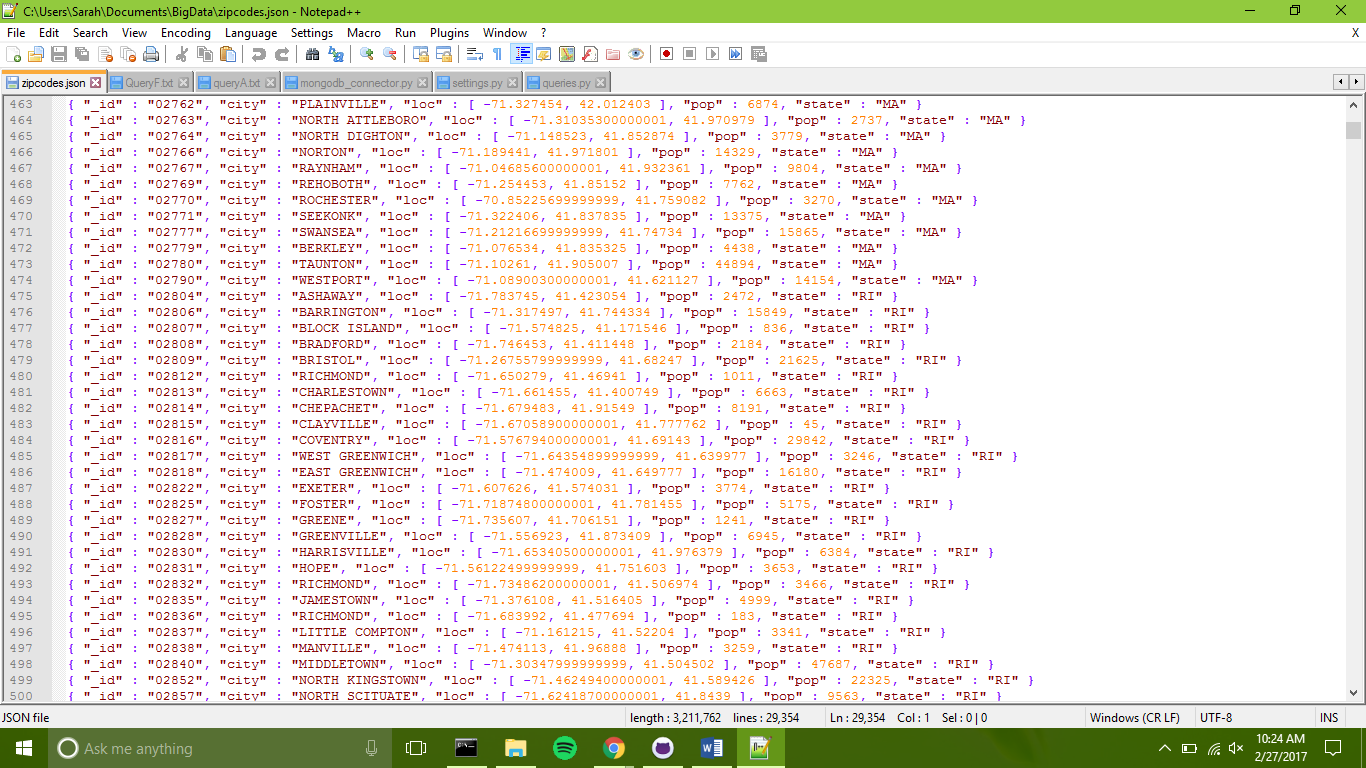
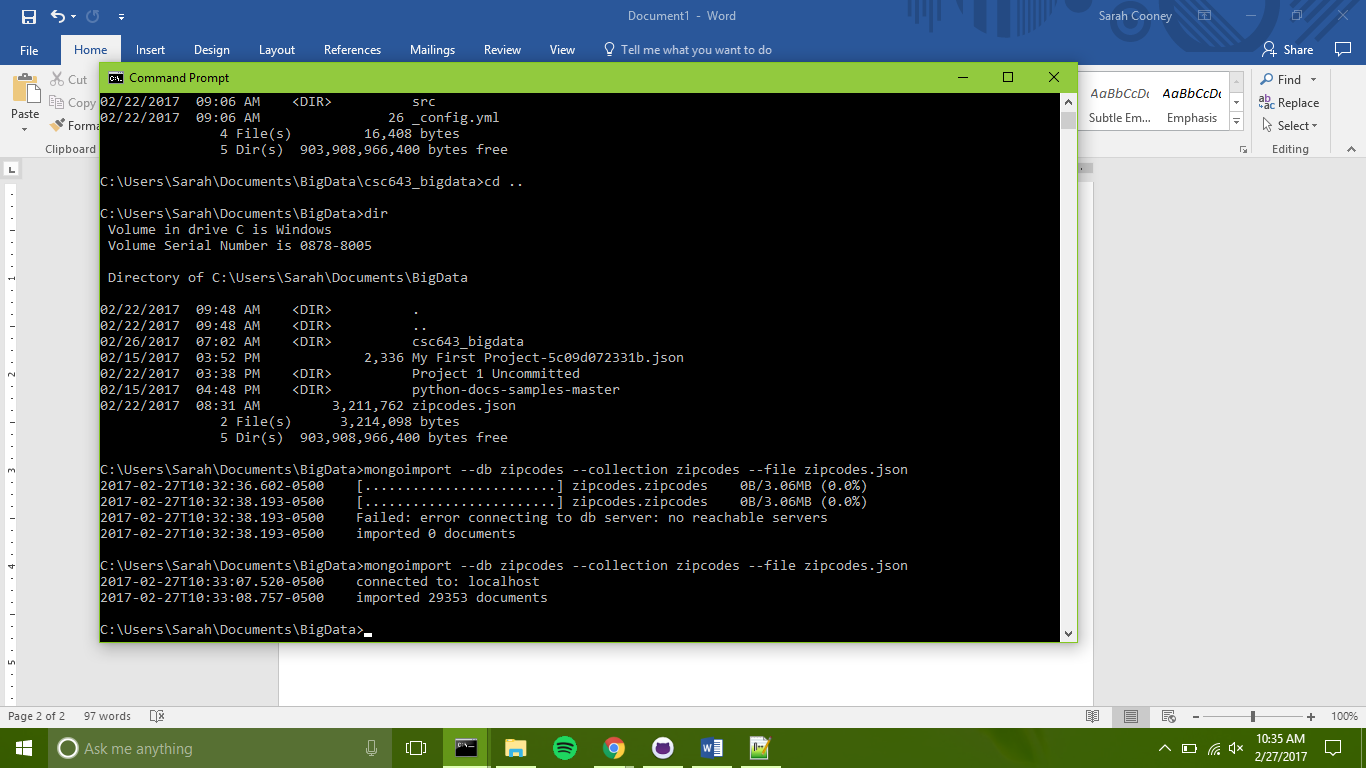


Figure 1. Sample of Zipcodes.json

Figure 2. Command to import zipcodes.json as a mongo database



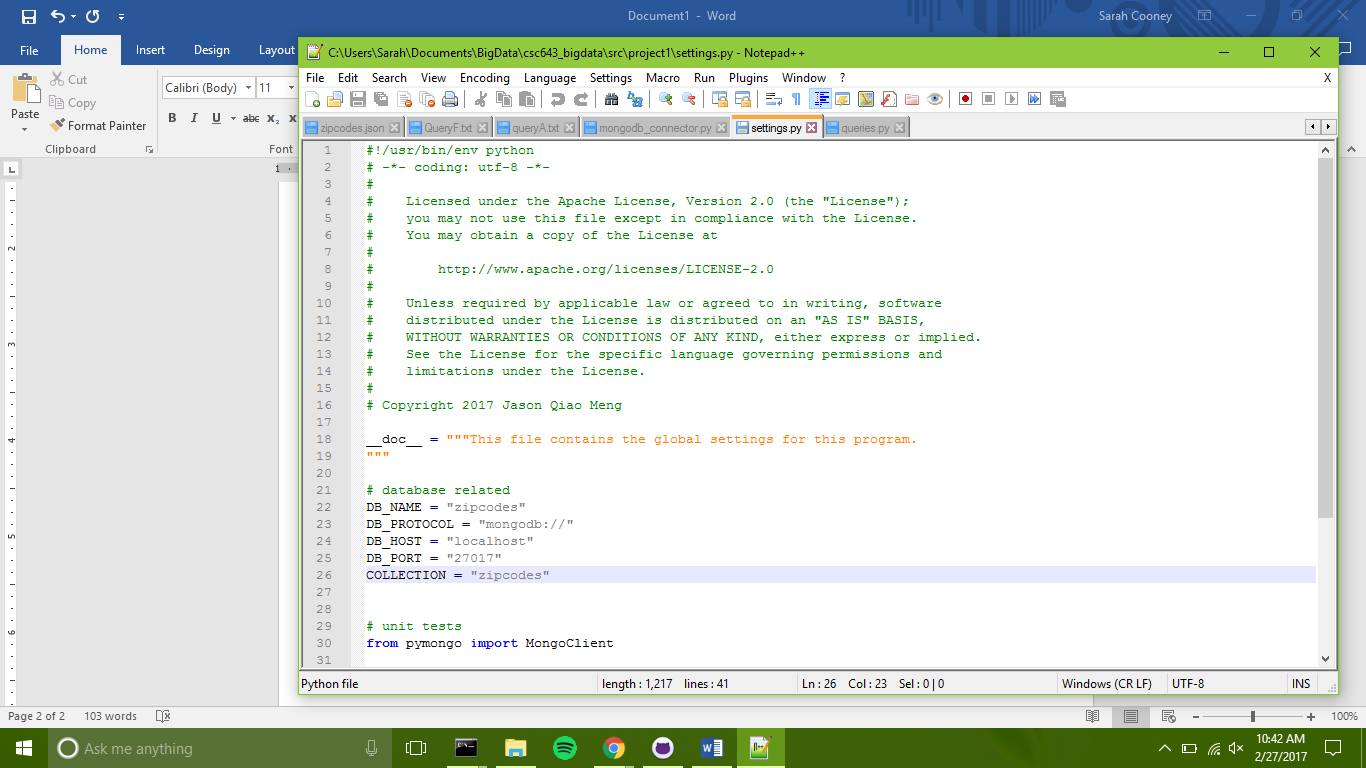
# **2. settings.py and mongodb\_connector.py**

These files use python to establish a connection to the zipcodes mongo database. The mongod command must be run in the command line first to start the server before using these files to establish a connection to the database.

## **2.1 settings.py**

The settings.py holds variables containing the global settings for connecting to a database with mongod\_connector.py. The variables are shown in Figure 3. DB\_NAME and COLLECTION are used to specify which database and collection to use after the connection has been establish. DB\_PROTOCOL specifies the type of database connection, in this case MongoDB. DB\_HOST and DB\_PORT specify the server and port on which to connect.

Figure 3. settings.py



## **2.2 mongodb\_connector.py**

This file uses the pymongo library to create a connection to a MongoDB database. The e variables defined in settings.py are utilized in this file. The file contains a class called MongoDB which is used to connect to the mongo client and get an instance of a database. Figure 4 shows the \_\_init\_\_ method which creates and initializes an empty instance of the MongoDB class.

The code for the simple\_connection\_string method is shown in Figure 5. This method sets up and returns the string used by pymongo to connect to the Mongo client, and incorporates the protocol, host, and port variables defined in settings.py. The get\_client method actually establishes and returns a connection to the Mongo client. The method starts by checking if the client exists. If it does not, the simple\_connection\_string method is used with the pymongo MongoClient method to attempt to make a connection. If the connection fails, an error message is presented and the client is set to “None” and returned. Otherwise, the successfully connected client is returned.

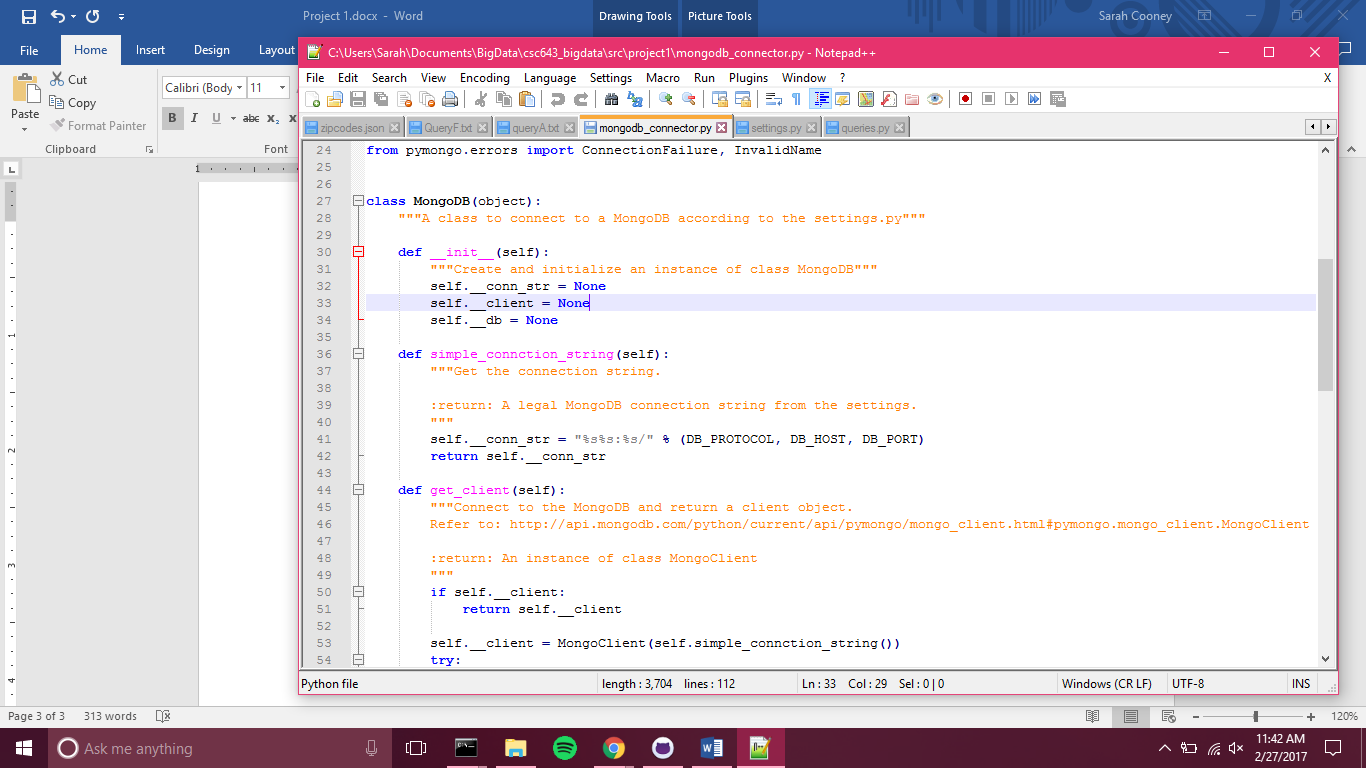


Figure 5. The method to set up the string used to connect to the Mongo client

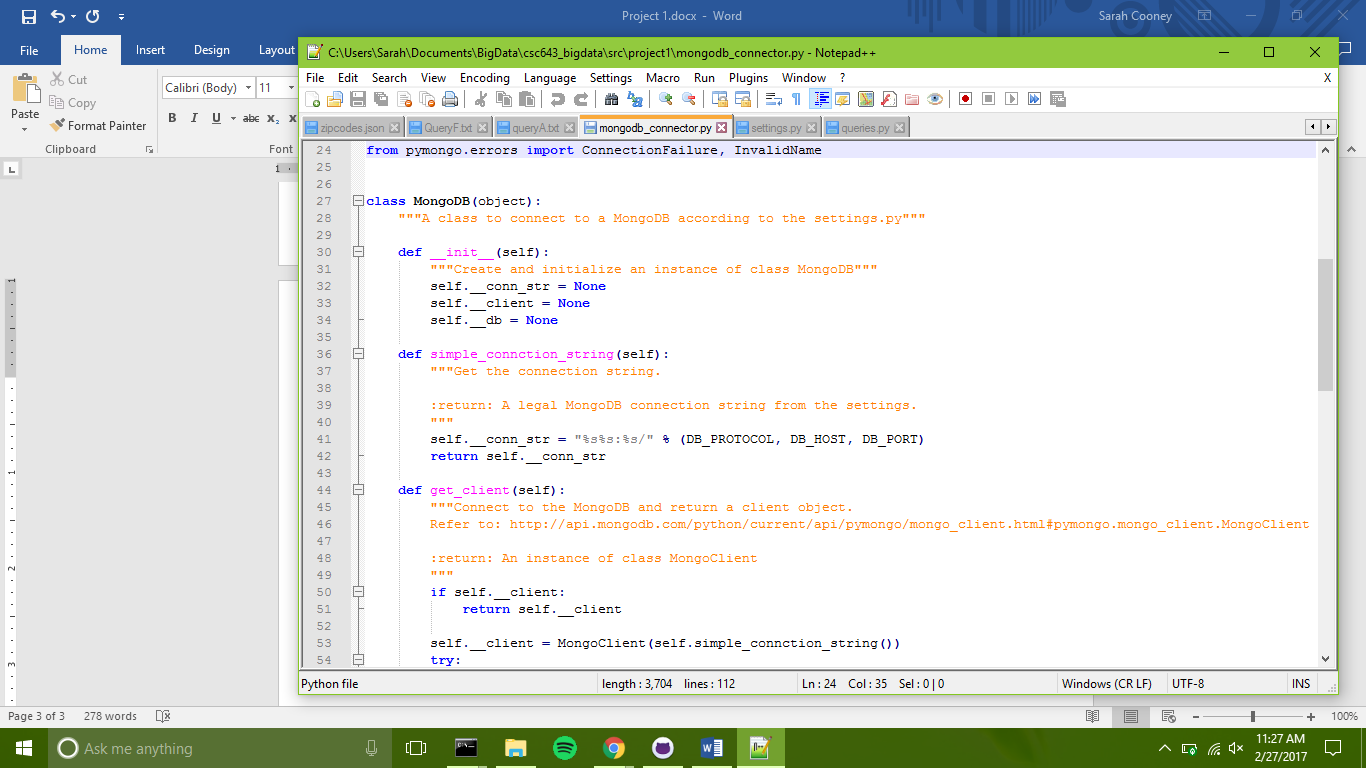


Figure 4. The initialization method for the MongoDB class