Ethan Daugherty

SNHU

CS-340-T4528

4-1 Milestone: Create and Read in Python

03/27/2022

**Create and Read in Python**

After completing the readings for this module, you will implement the fundamental operations of creating and reading documents (the C and R of CRUD) in Python. You will use the PyMongo driver to create CRUD functional access to your document collection.

1. Upload the Austin Animal Center (AAC) Outcomes data set into MongoDB by **importing a CSV file using the appropriate MongoDB import tool**. This file is in the /usr/local/datasets/ directory in Apporto and the filename is “aac\_shelter\_outcomes.csv”. Use the database name “AAC” and collection name “animals”. Complete the import using the mongoimport tool and **take screenshots** of both the import command and its execution.  
     
   Note: If you completed the Module Three milestone, you have already completed this step.

Graphical user interface, text, application

Description automatically generated

1. Next, you must develop a Python module in a PY file, using object-oriented programming methodology, to enable **create** and **read** functionality for the database. To support code reusability, your Python code needs to be importable as a module by other Python scripts.  
   **Develop** a CRUD class that, when instantiated, provides the following functionality:
   1. **A method that inserts a document into a specified MongoDB database and collection**
      1. Input -> argument to function will be set of key/value pairs in the data type acceptable to the MongoDB driver insert API call
      2. Return -> “True” if successful insert, else “False”
   2. **A method that queries for documents from a specified MongoDB database and specified collection**
      1. Input -> arguments to function should be the key/value lookup pair to use with the MongoDB driver find API call
      2. Return -> result in cursor if successful, else MongoDB returned error message  
           
         Important: Be sure to use **find()** instead of **find\_one()** when developing your method.

As you develop your code, be sure to **use industry standard best practices** such as proper naming conventions, exception handling, and in-line comments. This will ensure that your code is easy to read and reusable for future projects.  
  
TIP: Use the following sample code to get started. Note that the authentication to MongoDB is in the initialization method for the CRUD class.

**Example Python Code to Insert a Document**

from pymongo import MongoClient

from bson.objectid import ObjectId

class AnimalShelter(object):

""" CRUD operations for Animal collection in MongoDB """

def \_\_init\_\_(self):

# Initializing the MongoClient. This helps to

# access the MongoDB databases and collections.

self.client = MongoClient('mongodb://%s:%s@localhost:YOUR\_PORT\_NUMBER' % (username, password))

self.database = self.client['project']

# Complete this create method to implement the C in CRUD.

def create(self, data):

if data is not None:

self.database.animals.insert(data) # data should be dictionary

else:

raise Exception("Nothing to save, because data parameter is empty")

# Create method to implement the R in CRUD.

Graphical user interface, text, application

Description automatically generated

1. Finally, **create a Python testing script that imports your CRUD Python module to call and test the create and read instances of CRUD functionality**. Be sure to use the username and password for the “aacuser” account for authentication when instantiating the class. This script should be created in a separate Jupyter Notebook IPYNB file, and should import and instantiate an object from your CRUD library to effect changes in MongoDB. After creating your script, execute it in Jupyter Notebook and take screenshots of the commands and their execution.

Animal Creation:Graphical user interface, text

Description automatically generated

Invalid animal Creation:   
Graphical user interface, text, application, email

Description automatically generated

Searching for created pet “Rex”:

Text

Description automatically generated

Invalid Search query:

Graphical user interface, text, application

Description automatically generated