Southern New Hampshire University

CS-340 README Template

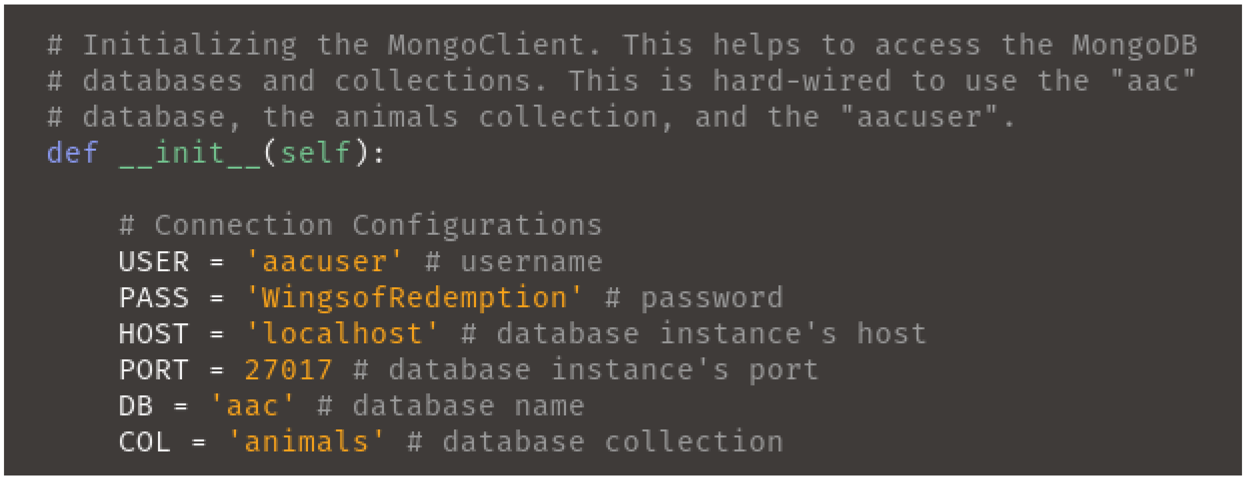
About the Project
The following is a simple “Python CRUD library” that has been developed to interface with a MongoDB instance storing BSON documents. This library will allow for programmers to insert new documents into the MongoDB instance, query the database for BSON documents, and to update and delete documents.

Motivation
The Python CRUD library was invented for our client, *Grazioso Salvare*, to be used for data modelling and data management. Its primary purpose is to display BSON documents as quantitative data in a Python web application, but the module can be repurposed for other kinds of data projects involving CRUD operations with the MongoDB.

Installation
The only apparatus needed to use the toolkit is the Python interpreter, and the PyMongo module.

Getting Started
To get a local copy up and running, follow these simple example steps:

1. Copy the library Python file into the same directory as the Python application being developed.
2. Ensure that the `pymongo` module is up to date with the following command: `pip3 -m pymongo –upgrade`
3. Configure the module to connect to the MongoDB instance; specifically by setting the following variables in the module:



```
# Initializing the MongoClient. This helps to access the MongoDB
# databases and collections. This is hard-wired to use the "aac"
# database, the animals collection, and the "aacuser".
def __init__(self):

    # Connection Configurations
    USER = 'aacuser' # username
    PASS = 'WingsofRedemption' # password
    HOST = 'localhost' # database instance's host
    PORT = 27017 # database instance's port
    DB = 'aac' # database name
    COL = 'animals' # database collection
```

Usage

Note: This template has been adapted from the following sample templates: Make a README, Best README Template, and A Beginners Guide to Writing a Kickass README.

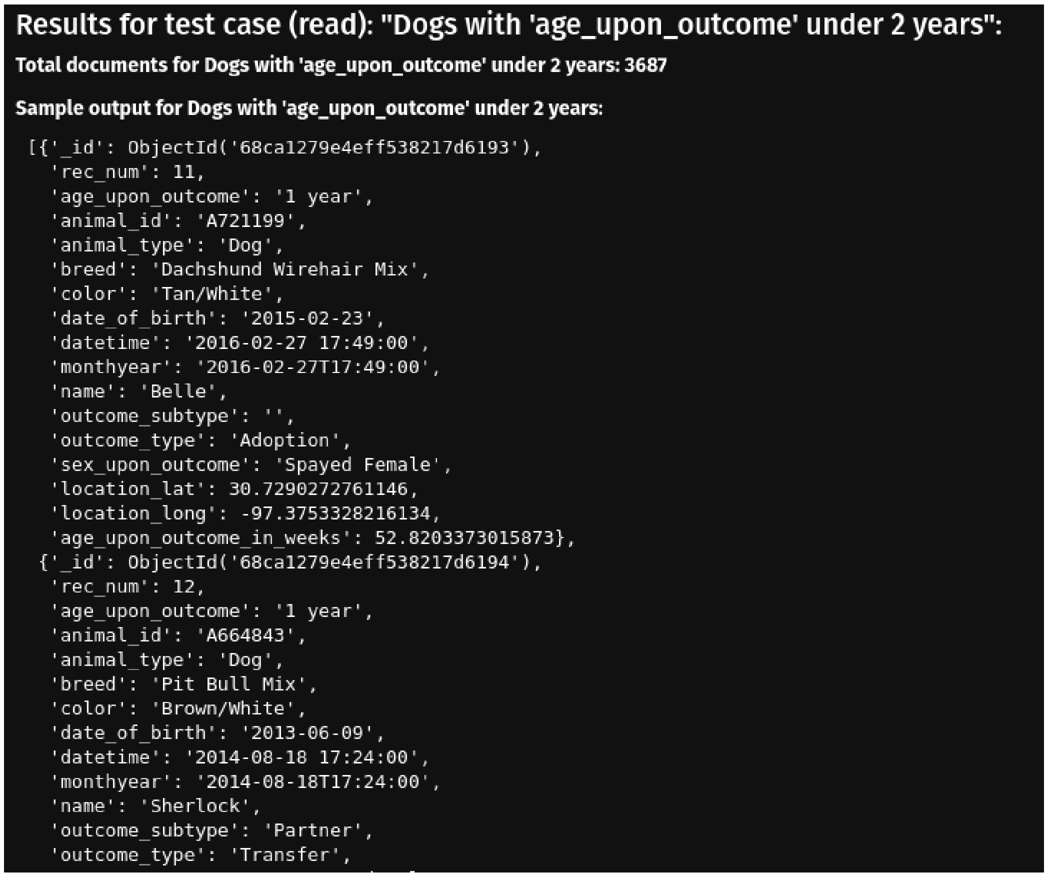
Begin by importing the Python CRUD library and then initiating the `AnimalShelter()` class:

```
from CRUD_Python_Module import *
mongo_instance = AnimalShelter()
```

To read from the database with a filter:

```
mongo_instance.read({
      "animal_type":"Dog", "age_upon_outcome": {
         "$in": [
            "0 years", "1 day", "1 month", "1 week",
         ]
      }
   }
})
```

The only argument for the `.read()` function is the `query`, which is a Python dictionary that acts as a filter for MongoDB's `find()` query. The following is an screenshot showing the output of a slightly modified version of the query:



Results for test case (read): "Dogs with 'age_upon_outcome' under 2 years":

Total documents for Dogs with 'age_upon_outcome' under 2 years: 3687

Sample output for Dogs with 'age_upon_outcome' under 2 years:

```
[{'_id': ObjectId('68ca1279e4eff538217d6193'),
  'rec_num': 11,
  'age_upon_outcome': '1 year',
  'animal_id': 'A721199',
  'animal_type': 'Dog',
  'breed': 'Dachshund Wirehair Mix',
  'color': 'Tan/White',
  'date_of_birth': '2015-02-23',
  'datetime': '2016-02-27 17:49:00',
  'monthyear': '2016-02-27T17:49:00',
  'name': 'Belle',
  'outcome_subtype': '',
  'outcome_type': 'Adoption',
  'sex_upon_outcome': 'Spayed Female',
  'location_lat': 30.7290272761146,
  'location_long': -97.3753328216134,
  'age_upon_outcome_in_weeks': 52.8203373015873},
 {'_id': ObjectId('68ca1279e4eff538217d6194'),
  'rec_num': 12,
  'age_upon_outcome': '1 year',
  'animal_id': 'A664843',
  'animal_type': 'Dog',
  'breed': 'Pit Bull Mix',
  'color': 'Brown/White',
  'date_of_birth': '2013-06-09',
  'datetime': '2014-08-18 17:24:00',
  'monthyear': '2014-08-18T17:24:00',
  'name': 'Sherlock',
  'outcome_subtype': 'Partner',
  'outcome_type': 'Transfer',
```

Programmers may consult the docstrings in the Python CRUD library for usage with the `create()`, `update()` and `delete()` functions.

Note: This template has been adapted from the following sample templates: [Make a README](), [Best README Template](), and [A Beginners Guide to Writing a Kickass README]().

Roadmap & Fixes Needed

Since the last module, further progress has been made in the development of this Python CRUD library. Thus, the “needed features” and given roadmap has changed:

- Refactor the Python library to remove redundant code.
    - Furthermore, use linters, unit testers, and software fuzzers to identify potential issues with code, and devise a more comprehensive set of test cases.
- Compare debugging and troubleshooting with `print()` and the `logging` module.
- Develop a front-end web application to visualize the data.
- Introduce more security features to prevent malicious attacks against the NoSQL instance done through exploitation of the client or any services.

Contact

Alexander Ahmann <alexander.ahmann@snhu.edu>

- Bluesky: @HypotheticalBlueskyAccount
- GitHub: @Alekseyyy

Note: This template has been adapted from the following sample templates: Make a README, Best README Template, and A Beginners Guide to Writing a Kickass README.