# CS-340-R1891 Project One

By Ryan LeChien

# Table of Contents

## README …......................................................................................................................................... 1.

**Milestone Three Screenshots ….......................................................................................................... 4.**

# Python-MongoDB CRUD README

## About the Project/Project Title

This project provides CRUD functionality via a Python API for a MongoDB database.

## Motivation

This project exists to provide a programmatic Python API for CRUD functionality.

## Getting Started

To get a local copy up and running, download the *animal\_shelter.py* script and place it in the directory of your project. Within your project, you can import it via the statement:



## Installation

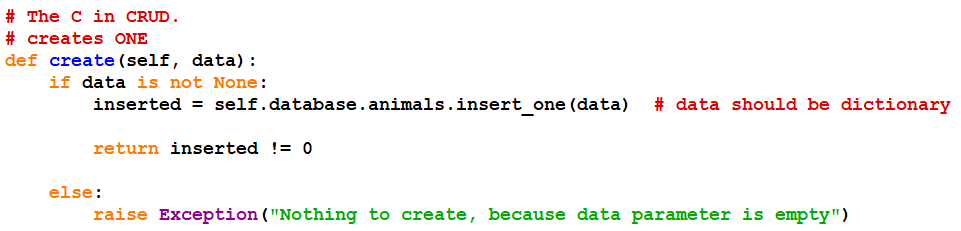
This project was built in Python version 3.10.7. The *animal\_shelter.py* script must be placed in the same directory of whichever project calls for this library.

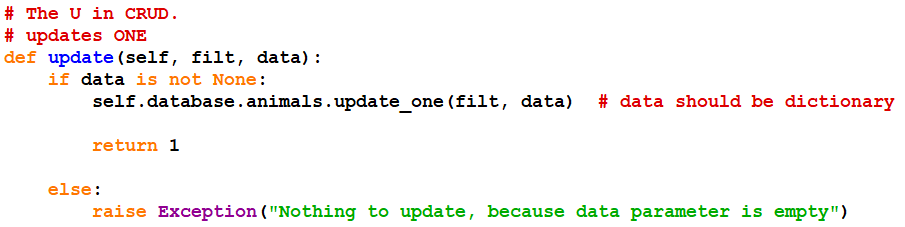
## Usage

**Available Methods**

* *\_\_init\_\_(self)*: The initializer method
* *create(self, data)*: The *data* parameter is a JSON object (parsed as a Python tuple). This method creates one.
* *read(self, data)*: The *data* parameter is a JSON object (parsed as a Python tuple). This method finds one.
* *update(self, filt, data)*: The *data* parameter is a JSON object (parsed as a Python tuple); the *filt* parameter specifies the matching filter. This method updates one.
* *delete(self, data)*: The *data* parameter is a JSON object (parsed as a Python tuple). This method deletes one, specifically the first one found.

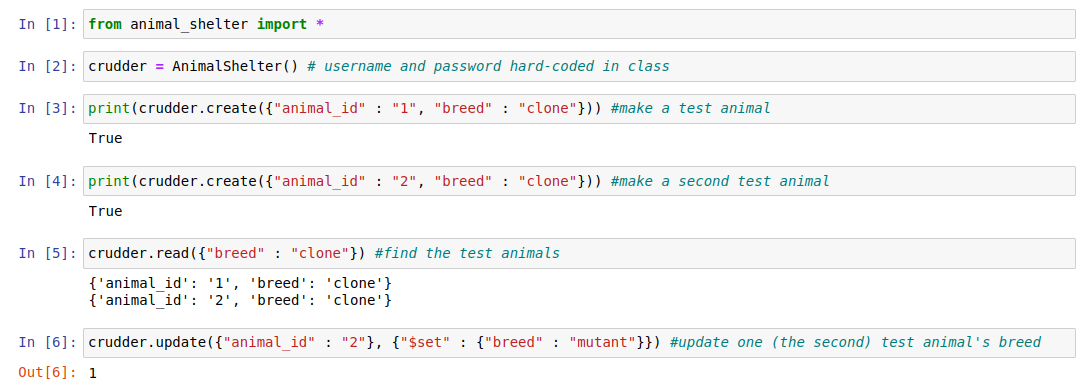
### Code Example

The implementation of the method *create()*

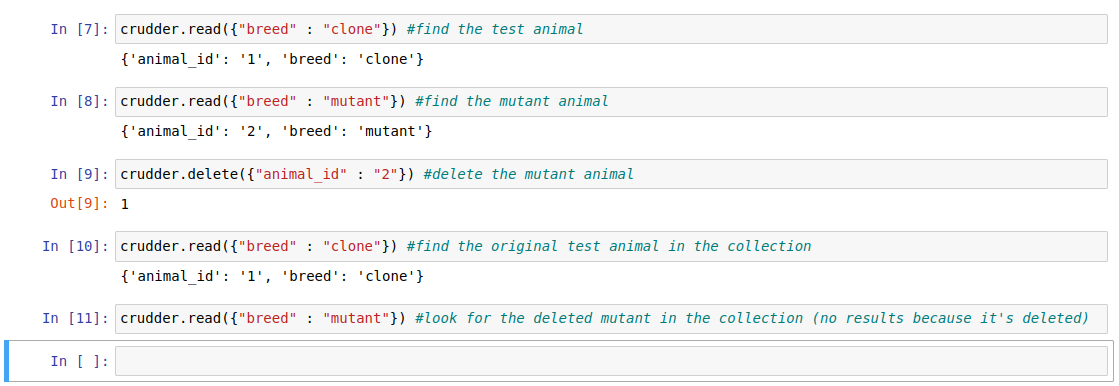


The implementation of the method *update()*

**Screenshots of Testing with Jupyter**



The Python CRUD module was tested in the Jupyter notebook per the following steps: import *animal\_shelter* module; instantiate the AnimalShelter class; **create** two animal entries, with different *animal\_id*s but of the same breed (“clone“); **read** all the animals of type ”clone”; **update** the *breed* of the animal with *animal\_id* 2 to ”mutant”; read the animals of type “clone” and “mutant,” verifying the update; **delete** the animal with *animal\_id* 2; read the animals of type “clone” and “mutant” again, verifying the deletion of “mutant” animal from the collection. These steps are illustrated in screenshots above and below.



## Roadmap/Features

* Full Python CRUD API for a MongoDB database
* Ability to pass credential arguments to the constructor to allow connection to different databases

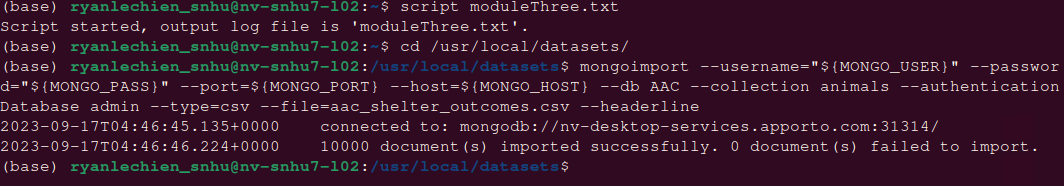
## Contact

Ryan LeChien

[ryan.lechien@snhu.edu](mailto:ryan.lechien@snhu.edu)

**Module Three Milestone**

**Part I: Importing and Indexing a Data Set**



Start a script output file, change the directory, and import the CSV file



Create a simple index on the key *breed*, then explain the finding from the collection *animals*



Drop the simple index

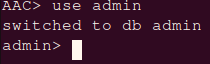


Create a compound index and explain a finding



Drop the compound index

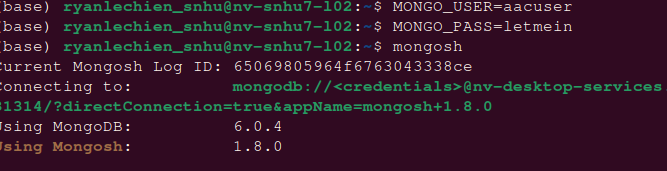
**Part II: User Authentication**



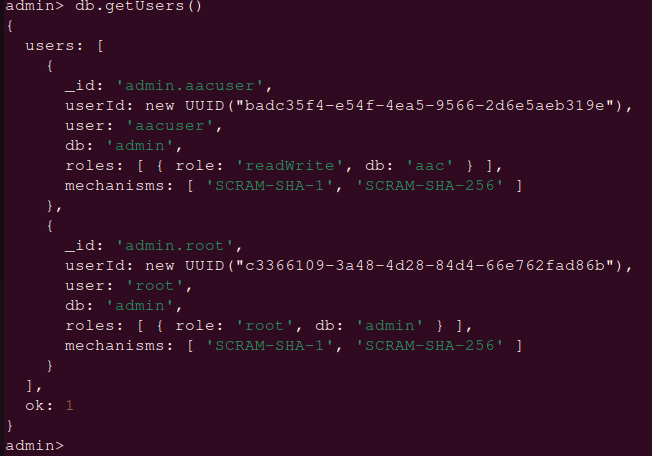
Switch to *admin* database



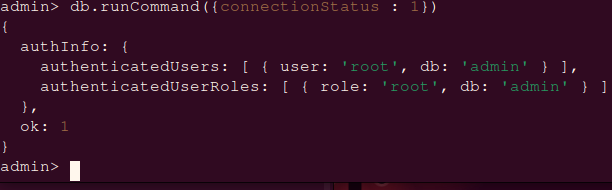
Create user *aacuser* with read-write privileges

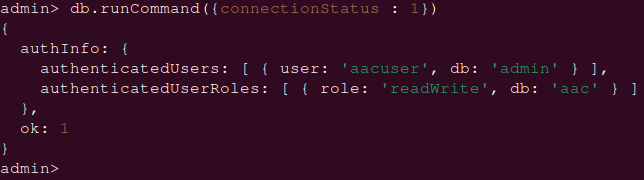


Login using new account credentials



From the *admin* terminal, verify both users are using *admin*

Connection status from the *admin* terminal



Connection status from the *aacuser* terminal