Lawrence Artl III
CS 340 22EW6
Assignment 4-2
July 23, 2022

CRUD.py with MongoDB Cloud Service / Local Host Service

About the Project/Project Title

The CRUD.py is a MongoDB project that aims to use CRUD operations (Create, Read, Update, and Delete) on a noSQL Mongo database using dictionaries. The project uses simple input validation in the CRUD module and is written in object notation for encapsulation of data.

Motivation

Every database needs a way to add and remove, read, and update data; this project provides a way to do that in as simple a format as possible, while still maintaining data validity within the database.

Getting Started

Download the “CRUD.py” file to a folder on your local machine. In that same folder, create another .py file for testing purposes. In your testing file, be sure to import CRUD, and from CRUD import the CRUD class. In the CRUD.py module change the connection string in MongoClient to match your log-in information for your own MongoDB server in the cloud. You can also follow this video here.

Installation

This project was originally built with Sublime Text 3, with Python enabled. See this tutorial here for setting that up. The tutorial includes instructions for installing Python as well. Once that is complete, follow the tutorial here to get started using MongoDB Atlas and Python to connect.

Usage

All methods take a JSON dictionary object as a parameter, save for “#UPDATE” which takes in two such objects. Error checking is done in the CRUD module’s methods.

Code Example

```
create(self, data):

        if data is None or data == {}                #empty object is passed

                print error message

        else

                results =                            #search for the data
                self.collection.find_one(data)

                if results is None                   #data is not already in db

                        self.collection.insert_one(da   #insert the data to the db
                        ta)

                        print success message

                else                                 #data is already in db

                        print error message
```

Tests

To test methods, first create an object of CRUD in your testing file:

```
c = CRUD()
```

Call each method with the CRUD object while passing in the appropriate data object or string:

```
#CREATE
c.create({“name”:”David”})
```

or

```
d = {“_id”: 5, “name”:”Lawrence”}
c.create(d)
```

All methods in CRUD take in one dictionary object, save for #UPDATE, as shown below:

```
#UPDATE
e = {“_id”: 6, “name”:”Larry”}
c.update(d, e)
```

Where object ‘e’ is the updated object and object ‘d’ is the object to be updated.

Screenshots

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

The following screenshots show a generic set of tests that produce the results on the right. These tests are labeled as to which method they are focused on at the time.

```
c = CRUD()

c.builddb()

#CREATE
d = {"_id": 5, "name":"Smoke", "Age": 15}
c.create(d)
c.create(d)


#READ
c.read(d)
c.read(d)
c.read(None)
c.read({"name":"Puck"})
c.read({"name":"Lawrence"})
c.read({"name":"lawrence"})
c.read({"name":"Larry"})
c.read({"name":"Lawrence"})


#UPDATE
e = {"name":"Puck"}
c.update(d, e)
c.read(d)
c.update({"name":"Lawrence"},{"name":"Larry"})
c.read({"name":"Larry"})


#DELETE
c.delete({"name":"Puck"})
c.read({"name":"Puck"})
c.delete({"name":"Puck"})
```

Figure1: Tests

```
SUCCESS - accessed database
SUCCESS - built a new database
SUCCESS - Smoke was successfully added.
ERROR - Data Smoke already in database.
SUCCESS - found Smoke
SUCCESS - found Smoke
ERROR - No data was passed in
ERROR - Puck was not found
SUCCESS - found Lawrence
ERROR - lawrence was not found
ERROR - Larry was not found
SUCCESS - found Lawrence
SUCCESS - updated Puck
ERROR - Smoke was not found
SUCCESS - updated Larry
SUCCESS - found Larry
SUCCESS - Puck deleted
ERROR - Puck was not found
ERROR - That data was not found
[Finished in 2.8s]
```

Figure 2: Results

Roadmap/Features

Future updates will include a more interactive way to use the database CRUD module, including:

- better error checking during updating
- use of more advanced features like “$in” and “$set” for querying within arrays or creating new data fields within documents, respectively
- use of “$regex” for querying partial-match strings in documents

Contact

Lawrence Artl III: GitHub LinkedIn Email artllj.com

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