plainDB

version 0.0.1

Animal Logic Assignment

July 28, 2020

Contents

Welcome to plaindb’s documentation!

Usage

Help

```
$ python ../plaindb/app.py -h
usage: app.py [-h] [-n NAME] [-a ADDRESS] [-p PHONE] -f FTYPE [-v VALUE] [-o]
              (-i | -q | -u | -b)

This is a complex db of the world that's so simple.

optional arguments:
  -h, --help            show this help message and exit
  -n NAME, --name NAME  Name of the Person.
  -a ADDRESS, --address ADDRESS
                        Address of the Person.
  -p PHONE, --phone PHONE
                        Phone Number of the Person.
  -f FTYPE, --file FTYPE
                        Core Database File Type. json/csv. Default is csv.
  -v VALUE, --value VALUE
                        New Value to be passed with -e/--edit option.
  -o, --out             Display the result in a web page. Used with -q option.
  -i, --insert          Insert Records to the DB.
  -q, --query           Query Records of the Person.
  -u, --update          Update the Record of the Person.
  -b, --blow            With power comes responsibility. This blows your
                        database.
```

Sample Commands

```
python plaindb/app.py -i -n smruti -p 0480253590 -a "4 VICTA ST" -f csv
```

Note

The above syntax inserts a record in a csv based file.

```
python plaindb/app.py -i -n smruti -p 0480253590 -a "4 VICTA ST" -f json
```

Note

The above syntax inserts a record in a json based file.

```
python plaindb/app.py -q -n smruti -f json
```

Note

The above syntax queries record from json based plain db. -q indicates the begining of a **select** statement. -n indicates querying for the table using name predicate. The above command translates to following query.

SELECT * from TABLE WHERE name = ‘smruti’ ;

```
python plaindb/app.py -q -n sm* -f json
```

Note

The select statement also supports regex characters. The above command returns all records from the table that matches the name starting with **sm**.

```
python plaindb/app.py -q -n smruti -p 480976000 -f csv
```

Note

The select statement also supports multiple predicates. The above command translates to :

SELECT * FROM table WHERE name = ‘smruti’ OR phone = ‘480976000’;

```
python plaindb/app.py -e -n smruti -p mohanty -f csv
```

Note

The Update statement gets triggered with the selection of -e option. The above command translates to :

UPDATE table_name SET name = ‘mohanty’ WHERE name = ‘smruti’;

Run Tests

```
`tox`
```

Plain DB API reference

“Core” module

core.**customlogger** (log_level=None, log_file=None)

customlogger Generates a custom Logging mechanism at INFO or DEBUG level. The Error logs are marked as ERROR. In case the user misses the log file name, it generates a default file by name VM_Restore_dmYHMS.log

Parameters:
- **log_level** –
- **log_file** –

Returns: A Custom logger to console and a file.

class core.**plainDB** (filename: str)

plainDB is classy blueprint for the plain db. Currently this supports 2 version. A csv (default) or a json based database file is created.

blow (logger)

blow Destroys the database instance.

Parameters: **logger** – A logger object.

query (logger, querytype, **parameters)

query function initiate DML function on the table.

Parameters:

- **self** – The instance of the object.
- **logger** – A logger object.
- **querytype** – select/insert/update.
- **parameters** – name/phone/address

:returns : A status code and a response.

Release Notes

Current Release : 0.0.1

Note

This is for the assignment purpose.

Index

Python Module Index

c