sofos Documentation

Release 0.8.5

Ted Lazaros

Aug 29, 2018

Contents:

CHAPTER 1

Quick startup tutorial

from your terminal run:

```
$ python3 -m venv ~/sofos-venv
$ source ~/sofos-venv/bin/activate
(sofos-venv) $ sofos-project sofos-test
(sofos-venv) $ cd sofos-test
(sofos-venv) $ python3 main.py
```

You just created your first sofos skeleton project. Have fun !!!

CHAPTER 2

sofos package

2.1 Submodules

2.2 sofos.database module

Module Database

class sofos.database.**Database**(*models*, *dbf=None*)

Bases: object

Database Class

backup_database()

Backup database with timestamp

calc_md5()

Calculates the md5 of the models schema

Parameters **models** – normally models.py from your project folder

Returns md5 value

create_database(*dbf*, *init_db_file=None*)

Create tables from model definitions

Parameters

- **dbf** – Database filename
- **models** – The models module to use (Normally models.py in your application's root)
- **init_db_file** – The init_db_file.sql file to use if present

create_z_table()

Create a metadata keys/values table and insert at least the md5 of the models schema.

Parameters **models** – normally models.py from your project folder

`integrity`(*parent_table*, *id_value*)

`integrity_dict`()

Integrity dictionary

Returns Dictionary {parent1: {child1: fld1, child2: fld2, . . . }, . . . }

`is_connected`

`is_database_compatible`(*dbf*)

This function checks the databases creation md5 against current models md5 in order to evaluate equality of the two schemas

Parameters

- **`dbf`** – Database file
- **`models`** – normally models.py from your project folder

Returns True if database schema is the same with model schema

`restore_database`()

Not implemented yet

`set_database`(*dbf*)

Set database if compatible

Parameters **`dbf`** – Database file name

Returns True or False.

`sql_database_create`()

Create sql for table creation according to your model settings

Parameters **`models`** – normally models.py from your project folder

Returns create sql

`table_labels`(*as_dict=False*)

Returns table labels

`table_names`()

Returns table names

`table_object`(*table_name*)

Returns table object by table name

Parameters **`table_name`** – table name

`table_objects`()

models: models.py from our project folder

Returns Dictionary

return dictionary format:

```
{table_name1: table_object1, ...}
```

2.3 sofos.dbf module

`sofos.dbf.`**`backup`**(*dbf*, *backupfile*, *overwrite=False*, *inserts_only=True*)

Backup database

Parameters

- **dbf** – Database file to backup
- **backupfile** – backup destination filename
- **overwrite** – If is allowed to overwrite
- **inserts_only** – If True backup data only, if False backup everything

Returns True if backup was successful

`sofos.dbf.`**`delete`**(*dbf*, *sql*)

Delete from db

Parameters

- **dbf** – Database file
- **sql** – sql

Returns True if success, False else

`sofos.dbf.`**`read`**(*dbf*, *sql*, *returns*)

SELECT

Parameters

- **dbf** – Database file
- **sql** – sql
- **returns** – Return Type

Returns one, list of dicts, cols/rows, rows

`sofos.dbf.`**`ref_exists`**(*dbf*, *table*, *field*, *idv*)

`sofos.dbf.`**`save`**(*dbf*, *sql*)

Safely save (create or update) data to database

Parameters

- **dbf** – Database file name (full path)
- **sql** – select sql

Returns True is operation successful, else False

`sofos.dbf.`**`script`**(*dbf*, *sql*, *create=False*)

Run sql script against database dbf

Parameters

- **dbf** – Database file name(full path)
- **sql** – Sql to execute (Normally inside Transaction)
- **create** – If True creates new database file

2.4 sofos.gr module

Greek language functions

`sofos.gr.`**`cap_first_letter`**(*txt*)

Capitalize first letter.

Example:

```
>>> import gr
>>> gr.cap_first_letter('abcd')
'Abcd'
```

`sofos.gr.`**`date2gr`**(*date*, *no_trailing_zeros=False*)

Create Greek Date

Parameters

- **`date`** – iso date ‘yyyy-mm-dd’
- **`date`** – iso_date
- **`no_trailing_zeros`** – Month, Day without trailing zeros

Returns ‘dd/mm/yyyy’

Example:

```
>>> import dategr
>>> dategr.date2gr('2017-01-05)
'05/01/2017'
>>> dategr.dat2gr('2017-01-15, no_trailing_zeros=True)
'5/1/2017'
```

`sofos.gr.`**`dec`**(*poso=0*, *decimals=2*)

Returns a decimal. If poso is not a number or None returns dec(0)

Parameters

- **`poso`** – the number to transofrm to decimal
- **`decimals`** – Number of decimals

Returns A decimal number with specific number of decimal digits

`sofos.gr.`**`dec2gr`**(*poso*, *decimals=2*, *zero_as_space=False*)

Returns string formatted as Greek decimal (1234.56 becomes 1.234,56)

Parameters

- **`poso`** – number to format
- **`decimals`** – Number of decimal digits
- **`zero_as_space`** – if True then zero values become one space

Returns Greek formatted number

Example:

```
>>> import gr
>>> gr.dec2gr('-2456')
'2.456,00'
>>> gr.dec2gr(0, zero_as_space=True)
' '
```

`sofos.gr.`**`gr2dec`**(*poso*, *decimals=2*)

Returns decimal (12.345,67 becomes 12345.67)

Parameters

- **`poso`** – text Greek formatted number
- **`decimals`** – decimal digits

Returns Decimal number

`sofos.gr.`**`gr2en`**(*txt*, *space=’ ’*)

Greek to Greeglish

Parameters

- **`txt`** – Text to translate to Greeglish
- **`space`** – If space == ‘’ then capitalize txt

`sofos.gr.`**`grup`**(*txtval*)

Trasforms a string to uppercase special for Greek comparison

`sofos.gr.`**`isNum`**(*val*)

Check if val is number or not

Parameters **`val`** – value to check

Returns True if val is number else False

`sofos.gr.`**`is_integer`**(*val*)

True if integer False otherwise

`sofos.gr.`**`is_iso_date`**(*strdate*)

Check if strdate is isodate (yyyy-mm-dd)

Parameters **`strdate`** – normally an iso formatted (yyyy-mm-dd) string

Returns True if iso_date False else

`sofos.gr.`**`is_positive_integer`**(*val*)

True if positive integer False otherwise

`sofos.gr.`**`is_weekdays`**(*value*)

Return True if value is weekdays False else

`sofos.gr.`**`rename_file`**(*fname*, *no_space=True*)

Rename a file

Parameters

- **`fname`** – file to rename
- **`no_space`** – remove spaces from filename

Returns A filename in greekglish

`sofos.gr.`**`triades`**(*txt*, *separator=’.’*)

Help function to split digits to thousants (123456 becomes 123.456)

Parameters

- **`txt`** – text to split
- **`separator`** – The separator to use

Returns txt separated by separator in group of three

Example:

```
>>> import gr
>>> gr.triades('abcdefg')
'a.bcd.efg'
>>> gr.triades('abcdefg', separator='|')
'a|bcd|efg'
```

2.5 sofos.models module

2.6 sofos.qt module

2.7 Module contents

CHAPTER 3

Indices and tables

- genindex
- modindex
- search

Python Module Index

Index