libJournal Documentation

Release 1.2.2

Aneesh Edara, Aviral Mishra

Aug 02, 2018

An open source library that lets you create and manage a digital journal:

1 Changelog 1
1.1 libJournal v.2.1.0 1
1.2 BROKEN VERSION: libJournal v.2.0.0 1
1.3 libJournal v1.2.2 2
1.4 libJournal v1.2.0 2
1.5 libJournal v1.1.0 2
1.6 libJournal v1.0 2

2 Version Documentation 3
2.1 libJournal v2.1.0 3
2.2 libJournal v2.0.0 4
2.3 libJournal v1.2.2 6
2.4 libJournal v1.2.0 7
2.5 libJournal v1.1.0 8
2.6 libJournal v1.0 8

3 Library Roadmap 11
3.1 libJournal v1.2.2 → v2.X.X 11

4 Features 13

5 Installation 15

6 Contribute 17

7 License 19

8 Indices and tables 21

CHAPTER 1

Changelog

1.1 libJournal v.2.1.0

Fixed Problems from v2.0.0, Syntax Changes, etc.

- Renamed format_db() to setup_db()
- Added consistency between parameters create_db() and setup_db()
- Fixed index error in __set_latest_entry_number()
- Fixed type error in __entry_iterate
- Updated documentation error from 2.0.0
- Fixed other outdated documentation

1.2 BROKEN VERSION: libJournal v.2.0.0

New Storage, Syntax, and Commands

- Moved to SQL storage instead of JSON
- New syntax for existing methods
- New search & read methods by name, date, or identifier numbers
- Allows for repeating entry names
- Added unique identifier numbers
- For more, please check the documentation

1.3 libJournal v1.2.2

Fixed Bugs Involving 'create_json()'

- Previous bug in which arguments passed through *create_json()* would not register has been fixed by adding *str* requirement
- Pypi upload bug has been fixed from 1.2.1

1.4 libJournal v1.2.0

New Method

- New method *create_json()* added: takes in *json_name* as a argument and creates *json_name.json*

1.5 libJournal v1.1.0

New Method, Syntax, and More!

- *find_entry()* method; finds entries and returns true or false
- *set_JSON_location()* has been changed to *set_json_location*
- *entry_read()* has been changed to *read_entry()* for consistency
- New self method *open_json()* has been added for profeciency
- Main file name has been changed from *json_journal.py* to *libj.py*
- All date & time functions have been moved to *libj_dt.py*

1.6 libJournal v1.0

Syntax & Pypi Fixes

- Fixed syntax of add & delete methods
- Fixed incorrect information inside *setup.py*

CHAPTER 2

Version Documentation

2.1 libJournal v2.1.0

This version of libJournal is very similar to v2.0.0 except it contains many fixes that ultimately broke v2.0.0. This version also contains different syntax for various methods. To install this version, user this command:

```
pip install libJournal==2.0.0
```

Since we no longer use JSON to store all the journal entries, we must go about making the storage differently. In order to start using libJournal 2.0.0, you must set it up as below in order to use any other features:

```
# Import libJournal
from libjournal.libj import libJournal

# Initialize libJournal object
libj = libJournal()

# Setup & mark the database (called test.db in this instance)
libj.setup_db("test.db")
libj.set_db_location("test.db")
```

NOTE: *create_db()* can be used to create a db if you would not like to format the database immediately. The parameters are the same as *setup_db()* The *entry_new()* method is used to add entries into the database, which records title, date, time, content, and number. The significance of number will be explained later:

```
# Adds an entry with the name "Dogs", and with the content "Cats"
libj.entry_new("Dogs", "Cats")
```

Unlike previous versions of libJournal, titles of entries **can** be the same!

In order to print/return all entries stored in certain database, use the *entries_all()* method, and for the most recent entry the *most_recent_entry()* method can be used, examples of both are below:

```
# Returns all entries in the database
libj.entries_all()

# Returns the most recent entry in the database
libj.most_recent_entry()
```

Due to the usage of SQLite3, more advanced search options are available, such as by title, date, and number. Number is an automatically assigned integer which is given to each entry upon creation, and can be used to specifically reference a single entry. Examples of all forms of searching are below:

```
# Searches all entries for titles named "Cats" (str). It will return a list of␣
↪entries each assigned a number (in order of time) which can be used to read them.
libj.search_by_title("Cats")

# Searches all entries for the specific identifier number 4 (int) and tt will return␣
↪a list of entries each assigned a number (in order of time) which can be used to␣
↪read them.
libj.search_by_number(4)

# Searches all entries for the entries created on the date and it will return a list␣
↪of entries each assigned a number (in order of time) which can be used to read them.
libj.search_by_date("2018-04-01")
```

In order to read the contents of entries, 3 commands exist each searching by one of the parameters listed above: title, number, date:

```
# Reads the second entry titled cats and will return it's contents
libj.search_by_title("Cats", 2)

# Reads the entry wit the identifier 4 and return it's contents
libj.search_by_number(4)

# Reads the third entry made on the date "2018-04-01" and returns it's contents
libj.search_by_date("2018-04-01", 3)
```

Finally, in order to delete an entry, you must delete it by it's identifier number using the *delete_by_number()* method:

```
# Deletes the entry with the identifier number 4
libj.delete_entry(4)
```

And this is all that is included in libJournal 2.1.0! Check the library road-map to see upcoming features and fixes. Downloads:

- Github
- pypi

2.2 libJournal v2.0.0

WARNING: This version of libJournal is BROKEN. Please refer to libJournal v2.1.0 which contains various fixes for problems within this version, along with some consistency fixes.

~~This version of libJournal is completely different from past versions, using all new behind the scenes libraries. Using this version is slightly more complicated, but becomes more effecient and has many new features to utilitze. To install this version, user this command:

```
pip install libJournal==2.0.0
```

Since we no longer use JSON to store all the journal entries, we must go about making the storage differently. In order to start using libJournal 2.0.0, you must set it up as below in order to use any other features:

```
# Import libJournal
from libjournal.libj import libJournal

# Initialize libJournal object
libj = libJournal()

# Create, format, and setup the database (called test.db in this instance)
libj.create_db("test")
libj.format_db("test.db")
libj.set_db_location("test.db")
```

The *entry_new()* method is used to add entries into the database, which records title, date, time, content, and number. The significance of number will be explained later:

```
# Adds an entry with the name "Dogs", and with the content "Cats"
libj.entry_new("Dogs", "Cats")
```

Unlike previous versions of libJournal, titles of entries **can** be the same!

In order to print/return all entries stored in certain databse, use the *entries_all()* method, and for the most recent entry the *most_recent_entry()* method can be used, examples of both are below:

```
# Returns all entries in the database
libj.entries_all()

# Returns the most recent entry in the databse
libj.most_recent_entry()
```

Due to the usage of SQLite3, more advanced search options are available, such as by title, date, and number. Number is an automatically assigned integer which is given to each entry upon creation, and can be used to specifically reference a single entry. Examples of all forms of searching are below:

```
# Searches all entries for titles named "Cats" (str). It will return a list of␣
↪entries each assigned a number (in order of time) which can be used to read them.
libj.search_by_title("Cats")

# Searches all entries for the specific identifier number 4 (int) and tt will return␣
↪a list of entries each assigned a number (in order of time) which can be used to␣
↪read them.
libj.search_by_number(4)

# Searches all entries for the entries created on the date and it will return a list␣
↪of entries each assigned a number (in order of time) which can be used to read them.
libj.search_by_date("2018-04-01")
```

In order to read the contents of entries, 3 commands exist each searching by one of the parameters listed above: title, number, date:

```
# Reads the second entry titled cats and will return it's contents
libj.search_by_title("Cats", 2)

# Reads the entry wit the identifier 4 and return it's contents
```

(continues on next page)

(continued from previous page)

```
libj.search_by_number(4)

# Reads the third entry made on the date "2018-04-01" and returns it's contents
libj.search_by_date("2018-04-01", 3)
```

Finally, in order to delete an entry, you must delete it by it's identifier number using the *delete_by_number()* method:

```
# Deletes the entry with the identifier number 4
libj.delete_entry(4)
```

And this is all that is included in libJournal 2.0.0! Check the library roadmap to see upcoming features and fixes. Downloads:

- Github
- pypi

~~

2.3 libJournal v1.2.2

This version of libJournal is identical to v1.2.0, the only difference being a bug fix. Version 1.2.1 was skipped due to a pypi bug. In order to install this version, run this::

> pip install libJournal==1.2.2

Once installed, this must be completed before using the rest of the library. Yet this is slightly different as the create_json() method can be used if the JSON doesn't exist beforehand:

```
# Import libJournal
from libjournal.libj import libJournal

# Initialize the libJournal object & set the location of the JSON
libj = libJournal()

# Creates a json called "name.json" and then sets it as the JSON location
libj.create_json("name")
libj.set_json_location("name.json")
```

The entry_add() method is used to add entries and the entry_delete() method is used to delete entries:

```
# Adds an entry with the name "title", and with the content "Cats"
libj.add_entry("Title", "Cats")

# Deletes an entry with the name "title", and with the content "Cats"
libj.delete_entry("Title")
```

Please note that entry titles **cannot** be the same! Next in use the find_entry() method in order to search for an entry based on title and return true or false:

```
# Searches for the entry with the title "Title" and returns a boolean
libj.find_entry("Title")
```

Finally, you can use the read_entry() in order to read a certain entry:

```
# Opens the entry with the title "Title" and returns the content of that entry
libj.read_entry("Title")
```

That's all for this version of libJournal!

Downloads:

- Github
- pypi

2.4 libJournal v1.2.0

This version of libJournal is very similar to v1.1.0, and in-fact most of the documentation has been transfered over. The only difference is 1 new method. In order to install this version, run this:;

> pip install libJournal==1.2.0

Once installed, this must be completed before using the rest of the library. Yet this is slightly different as the create_json() method can be used if the JSON doesn't exist beforehand:

```
# Import libJournal
from libjournal.libj import libJournal

# Initialize the libJournal object & set the location of the JSON
libj = libJournal()

# Creates a json called "name.json" and then sets it as the JSON location
libj.create_json("name")
libj.set_json_location("name.json")
```

The entry_add() method is used to add entries and the entry_delete() method is used to delete entries:

```
# Adds an entry with the name "title", and with the content "Cats"
libj.add_entry("Title", "Cats")

# Deletes an entry with the name "title", and with the content "Cats"
libj.delete_entry("Title")
```

Please note that entry titles **cannot** be the same! Next in use the find_entry() method in order to search for an entry based on title and return true or false:

```
# Searches for the entry with the title "Title" and returns a boolean
libj.find_entry("Title")
```

Finally, you can use the read_entry() in order to read a certain entry:

```
# Opens the entry with the title "Title" and returns the content of that entry
libj.read_entry("Title")
```

That's all for this version of libJournal!

Downloads:

- Github
- pypi

2.5 libJournal v1.1.0

This version of libJournal is very different from v1.0. There are a couple syntax chagnes, yet most changes take place in the source code. To install this version of libJournal use this:

```
pip install libjournal==1.1.0
```

Once installed, one must setup before using libJournal:

```
# Import libJournal
from libjournal.libj import libJournal

# Initialize the libJournal object & set the location of the JSON
libj = libJournal()
libj.set_json_location("example.json")
```

The entry_add() method is used to add entries and the entry_delete() method is used to delete entries:

```
# Adds an entry with the name "title", and with the content "Cats"
libj.add_entry("Title", "Cats")

# Deletes an entry with the name "title", and with the content "Cats"
libj.delete_entry("Title")
```

Please note that entry titles **cannot** be the same! Next in use the find_entry() method in order to search for an entry based on title and return true or false:

```
# Searches for the entry with the title "Title" and returns a boolean
libj.find_entry("Title")
```

Finally, you can use the read_entry() in order to read a certain entry:

```
# Opens the entry with the title "Title" and returns the content of that entry
libj.read_entry("Title")
```

That's all for this version of libJournal!

Downloads:

- Github
- pypi

2.6 libJournal v1.0

This version of libJournal is very limited and simple. We'd recommend you to use one of the later version but this may suit your needs in some way. Setup is very simple. To download the package use this command:

```
pip install libjournal==1.0
```

Once installed, the setup in order to use this library is very simple:

```
import libjournal

# Creates new object instance & sets the JSON location
```

(continues on next page)

(continued from previous page)

```
libj = libjournal()
libj.set_JSON_location("example.json")
```

Now you're all set to start using the library! In this version of libJournal you add & delete entries like this:

```
# Adds an entry, first argument is the 'entry_title' and the secong argument is
↪'entry_content'
libj.add_entry("Title", "Content")
libj.delete_entry("Title")
```

In order to search & read entries, you must use this method:

```
# Searches and reads an entry, the first argument is 'entry_title'
libj.entry_read("Title")
```

And that's everything to this version of libJournal!

Downloads:

- Github
- pypi

CHAPTER 3

Library Roadmap

The roadmap is everything that we are planning on adding for the next release of libJournal. Check here before the release to see what's coming up and after to see what we're adding next release.

3.1 libJournal v1.2.2 → v2.X.X

As of this date, no bugs have been reported and new features have not been announced. Please check back later for further updates.

Libjournal is a free and open-source library that is currently in development. It allows you to create a digital journal, add, delete, edit, and search entries that are organized by time, date, name, and unique identifier numbers. Many features are to come, and for a list of coming features visit the roadmap!

Here is a simple example of libJournal (2.0.0):

```
# Import the library
from libjournal.libj import libJournal

# Create new object
libj = libJournal()

# Create the database in which the data is stored and set it as the location
libj.setup_db("meow.db")
libj.set_db_location("meow.db")

# Creates an entry with the name "Meow" and the content "Cats". The time, date, and␣
↪identifier number are automatically added
libj.entry_new("Meow", "I like cats")
```

CHAPTER 4

Features

- Create a database, format, and set location *once*
- Add entries with titles, can be repeated, and content
- Search up entries using name, date, or identifier
- Read entries using name, date, or identifier
- Delete entries based on unique identifier

CHAPTER 5

Installation

Install libjournal through pip by running:

```
pip install libjournal
```

You can also download libJournal manually through Github releases, or through pypi:

- Github
- pypi

CHAPTER 6

Contribute

- Issue Tracker: https://github.com/undystopia/libjournal/issues
- Source Code: https://github.com/undystopia/libjournal

CHAPTER 7

License

libJournal is licensed under GPL 3.0

CHAPTER 8

Indices and tables

- genindex
- modindex
- search