

---

# **success<sub>b</sub>ackup<sub>c</sub>heck***Documentation*

***Release 0.2.0***

**infolinuxluigi.com**

**Nov 20, 2017**



---

## Contents

---

<b>1</b>	<b>About</b>	<b>3</b>
1.1	Installation instructions . . . . .	3
1.2	Config . . . . .	6
1.3	Usage . . . . .	8
1.4	API Reference . . . . .	8
1.5	Help . . . . .	9
1.6	Changelog . . . . .	10
1.7	License . . . . .	10
1.8	Indices and tables . . . . .	10
	<b>Python Module Index</b>	<b>11</b>



**Warning:** Beta software You are using a software that has not reached a stable version yet. Please beware that interfaces might change, APIs might disappear and general breakage can occur before *1.0*.

If you plan to use this software for something important, please read the roadmap, and the issue tracker in Github. If you are unsure about the future of this project, please talk to the developers, or (better yet) get involved with the development of success-backup-check!



Move user files on a server from a place where the user has write rights & move it to an archive or backup folder, where to user has no read or write access.

A use case example: Daily backup of a point-of-sale database. So that every point-of-sale device has only the current database in the storage. The 2 server check in the operation when was the last backup & send an email if the current database on the server is to old.

## 1.1 Installation instructions

### 1.1.1 Requirements

- Python 3.x
- <https://www.smartmontools.org> for hardware tests

#### Linux

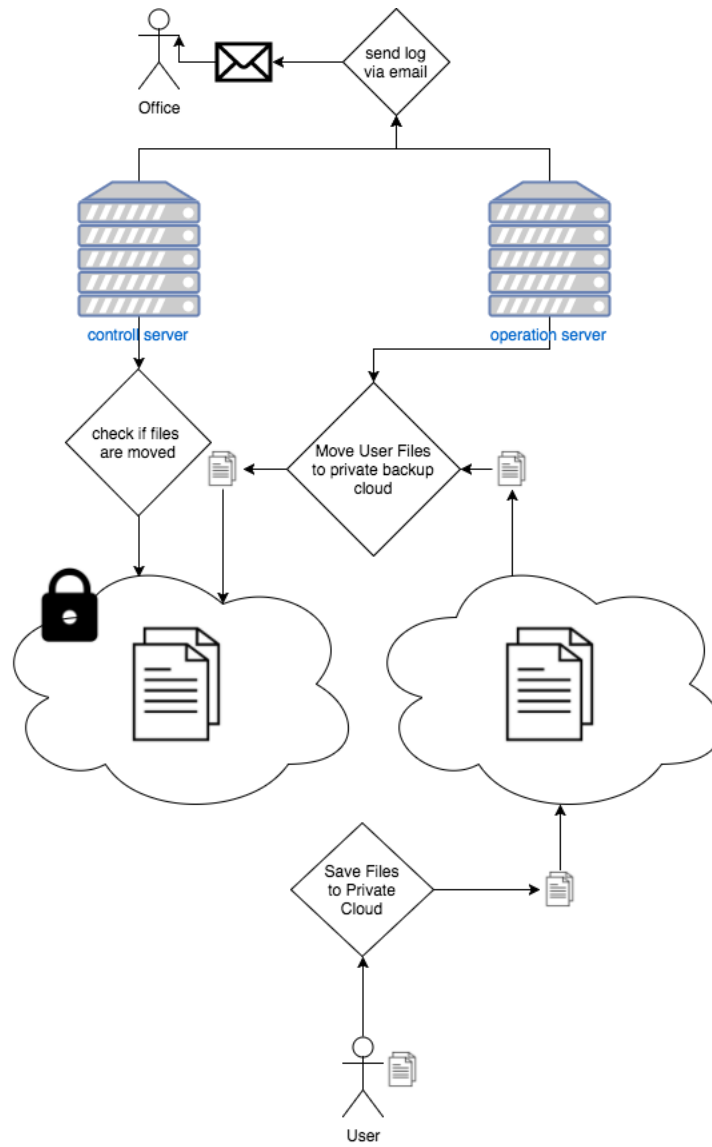
Debian / Ubuntu:

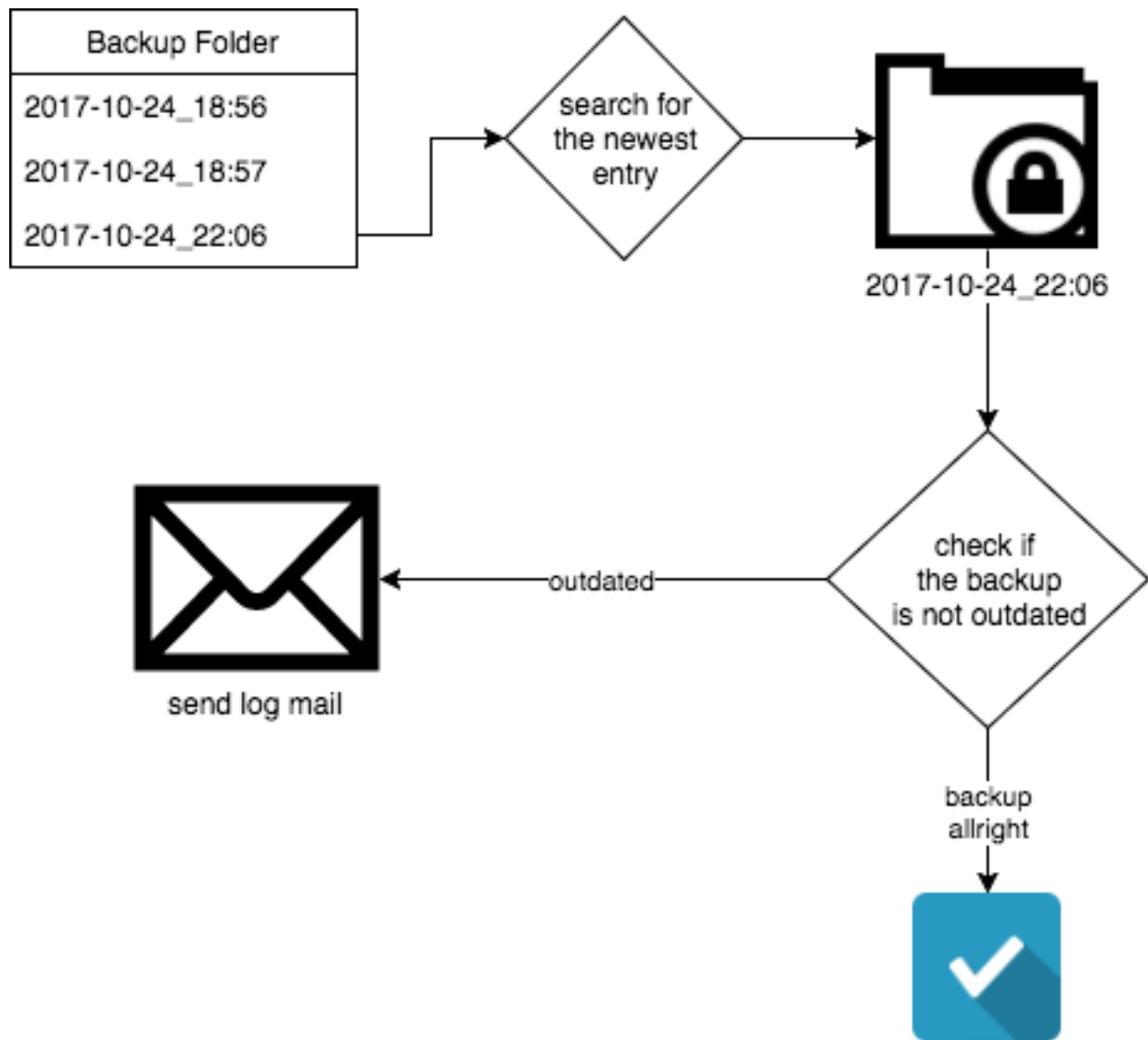
```
$ sudo apt-get install smartmontools
```

#### Mac

Brew:

```
$ brew install smartmontools
```





### 1.1.2 Install

success-backup-check can be installed using pip:

```
$ sudo python3 -m pip install git+git://github.com/linuxluigi/success-backup-check.git
```

This command will fetch the archive and its dependencies from the internet and install them.

Or download it from git and execute:

```
$ git clone git@github.com:linuxluigi/success-backup-check.git
$ cd success-backup-check
$ python setup.py install --user
```

You might prefer to install it system-wide. In this case, skip the `--user` option and execute as superuser by prepending the command with `sudo`.

### 1.1.3 Troubleshoot

Only tested on linux & mac, I don't know this will work correctly on windows machines.

Windows users may find that these command will only works if typed from Python's installation directory.

Some Linux distributions (e.g. Ubuntu) install Python without installing pip. Please install it before. If you don't have root privileges, download the `get-pip.py` script at <https://bootstrap.pypa.io/get-pip.py> and execute it as `python get-pip.py --user`.

## 1.2 Config

Config file path: ``/etc/success_backup_check.conf``

Complete Example:

```
[Mail]
From = from@example.com
To = to@example.com
ApiKey = YourSendGridApiKey

[Time]
days = 3

[Server]
ArchivDir = /srv/backup/daily_backup/
mode = active
file_typ = MDB

[BackupDirs]
MyDatabaseDir = /home/user/daily-db
UserWork = /home/user/done/work

[Logging]
log_level = WARNING
log_file = /var/log/succes_backup_check.log
```

### 1.2.1 Mail

Mails are send via <https://sendgrid.com> and need a From & To email address and also the sendgrid api key via ApiKey.:

```
[Mail]
From = from@example.com
To = to@example.com
ApiKey = YourSendGridApiKey
```

### 1.2.2 Time

Right now there is the section [Time] just one option. How many days one folder can be outdated.:

```
[Time]
days = 3
```

### 1.2.3 Server

[Server] is for selecting the master backup path on the server & set the server mode.

- ArchivDir is the master path in witch the backups are will be save to.
- mode has 2 values active -> move the files from original path to the backup folder & passive -> just check if the active server has done the work right. The default value is passive
- file\_typ set the typ of files witch should be backup. Examples all databases with the ending MDB.

```
[Server]
ArchivDir = /srv/backup/daily_backup/
mode = active
file_typ = MDB
```

### 1.2.4 BackupDirs

The [BackupDirs] Section set witch directory should be backed up. Every entry is a new directory. On the left side is the name of the new directory on the backup server & on the right side ios the full path of the to back up directory:

```
[BackupDirs]
MyDatabaseDir = /home/user/daily-db
UserWork = /home/user/done/work
```

### 1.2.5 Logging

[Logging] is for selecting the log\_level (WARNING, INFO, DEBUG) & where to save to the log\_file:

```
[Logging]
log_level = WARNING
log_file = /var/log/succes_backup_check.log
```

## 1.3 Usage

To run the program run:

```
$ success-backup-check
```

Or an example in crontab. (change your Python version):

```
$ python3 /usr/local/lib/python3.5/dist-packages/success_backup_check/__main__.py
```

### 1.3.1 Parameter

Param	Default Value	Function
<ul style="list-style-type: none"> <li>• <i>-c file</i></li> <li>• <i>-config file</i></li> </ul>	<i>/etc/success_backup_check.conf</i>	set the config file path
<ul style="list-style-type: none"> <li>• <i>-h</i></li> </ul>		show the help text
<ul style="list-style-type: none"> <li>• <i>-version</i></li> </ul>		show the version

## 1.4 API Reference

API documentation for the success-backup-check module.

`success_backup_check.__main__.get_parser()`

Creates a new argument parser.

`success_backup_check.__main__.main(args=None)`

Main entry point

**Parameters** `args` – list A of arguments as if they were input in the command line. Leave it None to use `sys.argv`.

`success_backup_check.archiv_files.archiv_files(directory, archive_dir, extension='MDB')`

Move Files with the default ending extension from `dir` to `archive_dir`:  
:param `directory`: original dir, where the database is right now  
:param `archive_dir`: archiv dir, werhe the database will move to  
:param `extension`: file ending name, default “MDB”

Returns:

`success_backup_check.archiv_files.search_dir(path, extension)`

extension with leading point, for example: “.MDB”

**Parameters**

- `path` –
- `extension` –

`success_backup_check.check_backup.check_backup(directory, days)`

### Check a Directory if the last modify date is older than n days

**Args:** directory: Directory witch will be checked days: modify time in days

**Returns:** False -> the dir is out of date True -> everything is fine

`success_backup_check.read_config.main(config_path)`

Read the config from the file at config\_path and return it's content :param config\_path: str complete path of the config file :type config\_path: object

**Returns** config content

`success_backup_check.set_logging.get_logging_level(logging_level)`

Change string into logging level. Example "DEBUG" -> logging.DEBUG

**Returns**

**logging level** "DEBUG" -> logging.DEBUG "INFO" -> logging.INFO "WARNING" -> logging.WARNING default -> logging.WARNING

**Return type** object

`success_backup_check.set_logging.set_logging(logging_level, log_file)`

Setup logging config to log into terminal & log file. Just execute set\_logging(logging\_level, log\_file) and start of the script & every time when logging will use this config.

**Parameters**

- **logging\_level** – string "DEBUG", "INFO" or "WARNING"
- **log\_file** – full path of the log file example: "/var/log/mylog.log"

`success_backup_check.set_logging.set_logging_minimal(logging_level)`

Set logging minimal logging, just writing log to console without saving it into a file. :param logging\_level: string "DEBUG", "INFO" or "WARNING"

`success_backup_check.hdd_smart_test.main(config)`

HDD SMART test

**Parameters** **config** (object) – config object from read\_config

`success_backup_check.hdd_smart_test.send_alert_sendmail(TEXT, failed_drives, config)`

HDD SMART test

**Parameters**

- **TEXT** (object) – Message witch should be send
- **config** (object) – config object from read\_config

## 1.5 Help

Need some help? Write me [info@linuxluigi.com](mailto:info@linuxluigi.com)

## 1.6 Changelog

Version	Content
0.2.0	<ul style="list-style-type: none"> <li>• Add documentation</li> <li>• Refactor Sourcecode</li> <li>• add HDD SMART test</li> <li>• add logging</li> <li>• add more options to config</li> <li>• add <code>-c file_path</code> &amp; <code>--config file_path</code> as program parameter</li> </ul>
0.1	<ul style="list-style-type: none"> <li>• Init project</li> </ul>

## 1.7 License

MIT License

Copyright (c) 2017 Steffen Exler

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the “Software”), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED “AS IS”, WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

## 1.8 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)

### S

- `success_backup_check.__main__`, 8
- `success_backup_check.archiv_files`, 8
- `success_backup_check.check_backup`, 8
- `success_backup_check.hdd_smart_test`, 9
- `success_backup_check.read_config`, 9
- `success_backup_check.set_logging`, 9



## A

archiv\_files() (in module success\_backup\_check.archiv\_files), 8

## C

check\_backup() (in module success\_backup\_check.check\_backup), 8

## G

get\_logging\_level() (in module success\_backup\_check.set\_logging), 9

get\_parser() (in module success\_backup\_check.\_\_main\_\_), 8

## M

main() (in module success\_backup\_check.\_\_main\_\_), 8

main() (in module success\_backup\_check.hdd\_smart\_test), 9

main() (in module success\_backup\_check.read\_config), 9

## S

search\_dir() (in module success\_backup\_check.archiv\_files), 8

send\_alert\_sendmail() (in module success\_backup\_check.hdd\_smart\_test), 9

set\_logging() (in module success\_backup\_check.set\_logging), 9

set\_logging\_minimal() (in module success\_backup\_check.set\_logging), 9

success\_backup\_check.\_\_main\_\_ (module), 8

success\_backup\_check.archiv\_files (module), 8

success\_backup\_check.check\_backup (module), 8

success\_backup\_check.hdd\_smart\_test (module), 9

success\_backup\_check.read\_config (module), 9

success\_backup\_check.set\_logging (module), 9