
BorgWeb Documentation

Release 0.0.0






see "AUTHORS" file

2018-06-25

Contents

1	What is BorgWeb?	3
1.1	Main features	3
2	Links	5
2.1	Notes	5

 /home/user/github/borgweb/logs/localhost-2015-09-09-16:26:57

-  localhost-2015-09-09-17:08:23
-  localhost-2015-09-09-17:08:11
-  localhost-2015-09-09-16:26:57
-  localhost-2015-09-09-16:26:51
-  localhost-

```
borg: Error: Repository repo does not exist.
Traceback (most recent call last):
  File "/home/pguth/github/borg/borg/archiver.py", line 923, in main
    exit_code = archiver.run(sys.argv[1:])
  File "/home/pguth/github/borg/borg/archiver.py", line 879, in run
    return args.func(args)
  File "/home/pguth/github/borg/borg/archiver.py", line 106, in do_create
    repository = self.open_repository(args.archive, exclusive=True)
  File "/home/pguth/github/borg/borg/archiver.py", line 40, in open_repository
    repository = Repository(location.path, create=create, exclusive=exclusive)
  File "/home/pguth/github/borg/borg/repository.py", line 60, in __init__
    self.open(path, exclusive)
  File "/home/pguth/github/borg/borg/repository.py", line 133, in open
    raise self.DoesNotExist(path)
borg.repository.DoesNotExist: repo

borg: Exiting with failure status due to previous errors
```

« Beginning... End of file »

What is BorgWeb?

BorgWeb is a browser-based user interface for [Borg Backup](#).

The UI is intended to help with everyday tasks, it is not intended as a full UI to everything borg can do.

You'll need help of an admin to install and configure it, as well as to restore backups.

1.1 Main features

- BorgWeb (the web service) usually runs on the machine that is backed up with BorgBackup. You can use a web browser to access BorgWeb from the same machine or from another machine.
- BorgWeb can operate using a builtin web server or as WSGI app using an external web server (like e.g. apache + mod_wsgi).
- using the web browser you can:
 - review backup log files
 - start a backup

- [Main Web Site](#)
- [PyPI packages](#)
- [Github](#)
- [Issue Tracker](#)

2.1 Notes

NOT RELEASED DEVELOPMENT VERSIONS HAVE UNKNOWN COMPATIBILITY PROPERTIES.
THIS IS SOFTWARE IN DEVELOPMENT, DECIDE YOURSELF WHETHER IT FITS YOUR NEEDS.
Please also see the [LICENSE](#) for more information.

2.1.1 Installation

BorgWeb requires:

- [Python](#) `>= 3.4`
- some python dependencies, see `install_requires` in `setup.py`

Currently, only installation from git repo checkout is supported.

Use `pip install -e .` from the top-level `BorgWeb` directory to install it into same virtualenv as you use for `BorgBackup`.

To install the complete environment for development you can do the following:

```
# Install Python code and dependencies:
virtualenv --python=python3 borg-env
source borg-env/bin/activate
```

(continues on next page)

(continued from previous page)

```
pip install tox pytest
git clone https://github.com/borgbackup/borgweb.git
cd borgweb
pip install -e .

# Install JS code and dependencies:
cd js
npm install

# Start the local Flask webserver:
cd ../../
mkdir logs
./borg-env/bin/borgweb

# Start the watch process and Browsersync
# In another shell navigate to `borgweb/js` and enter:
gulp watch
```

2.1.2 Configuration

The builtin default configuration expects a “repo/” directory and a “logs/” directory in the current working directory. This is mostly development and testing, we do not expect normal setups to use the default configuration.

You can override this by pointing to a custom configuration file via the environment variable BORGWEB_CONFIG.

The configuration file must only have lines like this (and NO indentation):

```
KEY = value # KEY must be all-uppercase, valid python syntax
```

See borgweb/config.py for the currently supported keys and example values.

Borg does by default not create log files. In order to create logs, the output of Borg must be redirected to a file.

2.1.3 Usage

Start the server process

For using BorgWeb, you need to start its web service first.

To start the builtin server, run “borgweb”.

You should see the builtin server starting and announcing the URL it serves. It will continue running, outputting log information until you stop it.

To stop the builtin server, type Ctrl-C or close the window.

Alternatively, experienced python web administrators can also use BorgWeb as a WSGI app, see the borgweb.wsgi python module.

Point your browser at the service URL

Use a web browser like Firefox and visit the web service URL.

For the builtin server it usually is: <http://127.0.0.1:5000/>

BorgWeb requires Javascript, so make sure it is not disabled.

2.1.4 Frequently asked questions

Which platforms are supported? For the BorgWeb web service, we try to support the same platforms as for BorgBackup.

Additionally, you will need some sane browser to access the web service, like Firefox or Chrome/Chromium. Using MS Internet Explorer [IE] (or other other browsers based on it) is discouraged and unsupported. If you run a sane browser, accessing the service should work from desktop and mobile platforms, we try to adapt to your screen resolution.

Why a web-based / browser-based approach? We didn't implement a "normal" desktop application, but a web app because:

- too many different desktop and mobile platforms to support (Linux, *BSD, Windows, Mac OS X, Android, iOS + a ton of different options per platform)
- html5, css and js works (almost) everywhere and we even have same code / similar UI everywhere.
- you can run the browser on the same machine as the backup software (typical desktop backup scenario), but you can also run it on another machine (server-without-GUI scenario) - more flexibility!

2.1.5 Support

Please first read the docs, the existing issue tracker issues and mailing list posts – a lot of stuff is already documented / explained / discussed / filed there.

Issue Tracker

If you've found a bug or have a concrete feature request, please create a new ticket on the project's [issue tracker](#).

For more general questions or discussions, IRC or mailing list are preferred.

IRC

Join us on channel #borgbackup on chat.freenode.net.

As usual on IRC, just ask or tell directly and then patiently wait for replies. Stay connected.

Mailing list

To find out about the mailing list, its topic, how to subscribe, how to unsubscribe and where you can find the archives of the list, see the [mailing list homepage](#).

Bounties and Fundraisers

We use [BountySource](#) to allow monetary contributions to the project and the developers, who push it forward.

There, you can give general funds to the borgbackup members (the developers will then spend the funds as they deem fit). If you do not have some specific bounty (see below), you can use this as a general way to say "Thank You!" and support the software / project you like.

If you want to encourage developers to fix some specific issue or implement some specific feature suggestion, you can post a new bounty or back an existing one (they always refer to an issue in our [issue tracker](#)).

As a developer, you can become a Bounty Hunter and win bounties (earn money) by contributing to BorgWeb, a free and open source software project.

We might also use BountySource to fund raise for some bigger goals.

2.1.6 Internals

This page documents the internal workings of BorgWeb.

What we use

- Flask and Werkzeug - Python web micro-framework and http toolbox
- Bootstrap and jQuery - CSS framework, Javascript library
- BorgBackup - for doing the backups

Develop JS

1. Have NodeJS/io.js and NPM installed.
2. `git clone https://github.com/borgbackup/borgweb.git`
3. `cd borgweb/js`
4. `npm install`
5. `gulp watch`
6. Edit JS files within `js/`; files will automatically be bundle into `borgweb/static/bundle.js`

2.1.7 BorgWeb Developers / Contributors (“The Borg Collective”)

- Thomas Waldmann <tw@waldmann-edv.de>
- Per Guth <mail@perguth.de> (<http://perguth.de>)

License

```
Copyright (C) 2015 The Borg Collective (see AUTHORS file)
All rights reserved.
```

```
Redistribution and use in source and binary forms, with or without
modification, are permitted provided that the following conditions
are met:
```

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The name of the author may not be used to endorse or promote products derived from this software without specific prior

(continues on next page)

(continued from previous page)

written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.