Airshare

Release 0.1.6

Jun 07, 2020

Contents

CHAPTER 1

What is Airshare?

Airshare is a Python-based CLI tool and module that lets you transfer data between two machines in a local network, P2P, using Multicast-DNS. It also opens an HTTP gateway for other non-CLI external interfaces. It works completely offline! Built with aiohttp and zeroconf.

CHAPTER 2

Features

- Blazing fast content transfer within a local network.
- Lets you transfer plain text, send from or receive into your clipboard.
- Supports transfer of multiple files, directories and large files - content is sent chunk by chunk and never read into memory entirely.
- Lets you send files whose paths have been copied into the clipboard (more details in the docs).
- Cross-platform, works on Linux, Windows and Mac (CLI and Web Interface), and also supports mobile (Web Interface).
- Uses Multicast-DNS service registration and discovery - so you can access content with human-readable code words.
- Can be used as a module in other Python programs.

CHAPTER 3

Important Links

- Source Code: https://github.com/KuroLabs/Airshare
- Bug Reports: https://github.com/KuroLabs/Airshare/issues
- Documentation: https://airshare.rtfd.io
- PyPI: https://pypi.org/project/Airshare

CHAPTER 4

Installation

- Installation using `pip`

```
$ pip install Airshare
```

- Installation using `pipx`

```
$ pipx install Airshare
```

- Installation using Homebrew

```
$ brew install airshare
```

CHAPTER 5

CLI Tool Reference

5.1 Airshare CLI Tool

5.1.1 Usage

```
Usage: airshare [OPTIONS] CODE [FILES]

  Airshare - an easy way to share content in a local network.

  CODE - An identifying code for Airshare.

  FILES - File(s) or directories to send.

Options:

  -p, --port INTEGER   Specify the port number to host a sending or receiving
                       server (default 8000).

  -t, --text TEXT      Send (serve) text content. For multiple words, enclose
                       within quotes.

  -u, --upload         Host a receiving server or upload file(s) to one.

  -cs, --clip-send     Send (serve) clipboard contents as text.

  -cr, --clip-receive  Receive served content and also copy into clipboard (if
                       possible).

  -fp, --file-path     Send files whose paths have been copied to the
                       clipoard.

  --help               Show this message and exit.

  --version            Show the version and exit.
```

5.1.2 Flags

- The `-t` flag can be used to send text using airshare, except to a receiving server.

```
$ airshare noobmaster -t "I'm still worthy!"
```

- The `-u` flag opens an upload endpoint to receive files from multiple users who initiate a send with the same flag. This is useful to receive files from devices without CLI support - they may simply visit the endpoint URL from any browser.

  At the receiver,

```
$ airshare -u noobmaster
```

  At the sender,

```
$ airshare -u noobmaster file.txt
```

- The `-fp` flag allows users to copy file or directory paths from the Finder or File Explorer and send them. Useful for selecting multiple files instead of typing file paths.

  Select required files and use the following shortcuts to copy file paths.

  For Mac,

  For Windows,

  For Linux,

  To send the files,

```
$ airshare -fp noobmaster
```

- The `-cs` flag allows users to directly send the clipboard contents as text.

  To send,

```
$ airshare -cs noobmaster
```

- The `-cr` flag allows users to copy the data received if clipboard compatible.

  To receive,

```
$ airshare -cr noobmaster
```

CHAPTER 6

Module Reference

6.1 Airshare Module

6.1.1 airshare.cli module

Command Line Interface for Airshare.

6.1.2 airshare.exception module

Exceptions defined for Airshare.

exception airshare.exception.**CodeExistsError**(*code*)
Bases: airshare.exception.AirshareError

Exception to be raised when Airshare code already exists.

To be raised when trying to create or serve an Airshare server, but one already exists with that identifying code.

exception airshare.exception.**CodeNotFoundError**(*code*)
Bases: airshare.exception.AirshareError

Exception to be raised when the Airshare code is not found.

To be raised when trying to download from or upload to an Airshare, but none exists with that identifying code.

exception airshare.exception.**IsNotReceiverError**(*code*)
Bases: airshare.exception.AirshareError

Exception to be raised when trying to upload to a non-receiver.

To be raised when trying to upload to an Airshare when it exists but is not a receiver.

exception airshare.exception.**IsNotSenderError**(*code*)
Bases: airshare.exception.AirshareError

Exception to be raised when trying to receive from a non-sender.

To be raised when trying to download from an Airshare when it exists but is not a sender.

6.1.3 airshare.receiver module

Module for receiving data and hosting receiving servers.

airshare.receiver.**receive**(*, *code*, *decompress=False*)
Receive file(s) from a sending server.

Parameters

- **code** (*str*) – Identifying code for the Airshare sending server.
- **decompress** (*boolean, default=False*) – Flag to enable or disable decompression (Zip).

Returns **text (or) file_path** – Returns the text or path of the file received, if successful.

Return type str

airshare.receiver.**receive_server**(*, *code*, *decompress=False*, *port=8000*)
Serves a file receiver and registers it as a Multicast-DNS service.

Parameters

- **code** (*str*) – Identifying code for the Airshare service and server.
- **decompress** (*boolean, default=False*) – Flag to enable or disable decompression (Zip).
- **port** (*int, default=8000*) – Port number at which the server is hosted on the device.

airshare.receiver.**receive_server_proc**(*, *code*, *decompress=False*, *port=8000*)
Creates a process with 'receive_server' as the target.

Parameters

- **code** (*str*) – Identifying code for the Airshare service and server.
- **decompress** (*boolean, default=False*) – Flag to enable or disable decompression (Zip).
- **port** (*int, default=8000*) – Port number at which the server is hosted on the device.

Returns **process** – A multiprocessing.Process object with 'receive_server' as target.

Return type multiprocessing.Process

6.1.4 airshare.sender module

Module for sending data and hosting sending servers.

airshare.sender.**send**(*, *code*, *file*, *compress=False*)
Send file(s) or directories to a receiving server.

Parameters

- **code** (*str*) – Identifying code for the Airshare receiving server.
- **file** (*str or list or None*) – Relative path or list of paths of the files or directories to serve. For multiple files or directories, contents are automatically zipped.
- **compress** (*boolean, default=False*) – Flag to enable or disable compression (Zip). Effective when only one file is given.

Returns **status_code** – Status code of upload POST request.

Return type int

airshare.sender.**send_server**(*, *code*, *text=None*, *file=None*, *compress=False*, *port=8000*)
Serves a file or text and registers it as a Multicast-DNS service.

Parameters

- **code** (*str*) – Identifying code for the Airshare service and server.
- **text** (*str or None*) – String value to be shared. If both *text* and *files* are given, *text* will be shared. Must be given if *files* is not given.
- **file** (*str or list or None*) – Relative path or list of paths of the files or directories to serve. If multiple files or directories are given, the contents are automatically zipped. If not given or both *files* and *text* are given, *text* will be shared. Must be given if *text* is not given.
- **compress** (*boolean, default=False*) – Flag to enable or disable compression (Zip). Effective when only one file is given.
- **port** (*int, default=8000*) – Port number at which the server is hosted on the device.

airshare.sender.**send_server_proc**(*, *code*, *text=None*, *file=None*, *compress=False*, *port=8000*)
Creates a process with 'send_server' as the target.

Parameters

- **code** (*str*) – Identifying code for the Airshare service and server.
- **text** (*str or None*) – String value to be shared. If both *text* and *files* are given, *text* will be shared. Must be given if *files* is not given.
- **file** (*str or list or None*) – Relative path or list of paths of the files or directories to serve. If multiple files or directories are given, the contents are automatically zipped. If not given or both *files* and *text* are given, *text* will be shared. Must be given if *text* is not given.
- **compress** (*boolean, default=False*) – Flag to enable or disable compression (Zip). Effective when only one file is given.
- **port** (*int, default=8000*) – Port number at which the server is hosted on the device.

Returns **process** – A multiprocessing.Process object with 'send_server' as target.

Return type multiprocessing.Process

6.1.5 airshare.utils module

Utility functions for Airshare.

airshare.utils.**get_local_ip_address**()
Obtains the device's local network IP address.

Returns **ip** – Packed 32-bit representation of the device's local IP Address.

Return type bytes

airshare.utils.**qr_code**(*url*)
Generate QR Code from URL and print it.

Parameters **url** (*str*) – URL to create the QR Code for.

airshare.utils.**get_service_info**(*code*)
Get service information for an Airshare service.

Parameters **code** (*str*) – Identifying code for the Airshare service.

Returns **info** – Details of the Airshare service.

Return type zeroconf.ServiceInfo

airshare.utils.**register_service**(*code*, *addresses*, *port*)
Registers an Airshare Multicast-DNS service based in the local network.

Parameters

- **code** (*str*) – Identifying code for the Airshare service.
- **addresses** (*list*) – List of local network IP Addresses for the service.
- **port** (*int*) – Port number for the Airshare service's server.

Returns **info** – Details of the Airshare service.

Return type zeroconf.ServiceInfo

airshare.utils.**get_zip_file**(*files*)
Creates a temporary Zip Archive of files and directories.

Parameters **files** (*list*) – List of paths of files and directories to compress.

Returns

- **zip_file_path** (*str*) – Canonical file path of the temporary Zip Archive file.
- **zip_file_name** (*str*) – File name to be assigned to the Zip Archive (during sending).

airshare.utils.**unzip_file**(*zip_file_path*)
Unzips a Zip Archive file into a new directory.

Parameters **zip_file_path** (*str*) – Path of the Zip Archive file to unzip.

Returns **zip_dir** – Canonical path of the unzipped directory.

Return type str

airshare.utils.**get_clipboard_paths**()
Extract file paths from the clipboard.

Returns **file_paths** – List of canonical paths extracted from the clipboard.

Return type list

airshare.utils.**is_file_copyable**(*file_path*)
Check if a file can be copied to the clipboard or not.

Parameters **file_path** (*str*) – Path of the file to check.

Returns **copyable** – True if the file can be copied to the clipboard, False otherwise.

Return type boolean

CHAPTER 7

Examples

7.1 Examples

7.1.1 CLI

Serving a file,

```
$ airshare noobmaster file.ext
```

Serving multiple files or directories,

```
$ airshare noobmaster file1.ext file2.ext dir1 ../dir2
```

Serving text,

```
$ airshare noobmaster -t "Some text here."
```

Serving clipboard text,

```
$ airshare -cs noobmaster
```

Serving files whose paths have been copied to the clipboard,

```
$ airshare -fp noobmaster
```

Uploading files,

```
$ airshare -u noobmaster file1.ext dir2 file2.ext
```

Uploading files whose paths have been copied to the clipboard,

```
$ airshare -u -fp noobmaster
```

Receiving a file or text,

```
$ airshare noobmaster
```

Receiving file uploads,

```
$ airshare -u noobmaster
```

Receiving a file or text and copying content to clipboard,

```
$ airshare -cr noobmaster
```

7.1.2 Module

```
import airshare

# Host a sending server (as a process, non-blocking code)
process = airshare.sender.send_server_process(code="noobmaster", text="Hi!")
# You can later terminate this process using `process.terminate()`

# Receive from an Airshare server
text = airshare.receiver.receive(code="noobmaster")

# Refer to the module documentation for more details
```

CHAPTER 8

Known Issues

- Link-local Name Resolution (for the `.local` addresses) on non-Apple devices requires Avahi (on Linux) or Bonjour (on Windows). Chances are you already have them, but if you don't, do check the web on how to install them.
- Android browsers do not have inbuilt Multicast-DNS service discovery, and cannot resolve the `.local` addresses. For this reason, we included QR Code support, for you to visit the URLs easily.
- Windows users with Python < 3.8, use `Ctrl` + `Break` to quit, as `Ctrl` + `C` will not work. This is a known issue with `asyncio`, which has been fixed in Python 3.8. If you do not have a `Break` key, try using `Ctrl` + `Fn` + `B`, or check the web for other alternatives (depending on your PC).

CHAPTER 9

Contributing

Pull requests are welcome. Read our Contribution Guide for more details.

CHAPTER 10

License

Airshare is licensed under the terms of the MIT License.

Airshare is the joint work of Kandavel A, Mohanasundar M and Nanda H Krishna.

```
MIT License

Copyright (c) 2020

Kandavel A <kanduarul@gmail.com>,
Mohanasundar M <itsmohanpierce@gmail.com> and
Nanda H Krishna <nanda.harishankar@gmail.com>

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.
```

CHAPTER 11

Acknowledgements

The Airshare logo was designed by Siddique.

The Airshare GIF was designed by Anam Saatvik.

CHAPTER 12

Indices and tables

- genindex
- modindex
- search

Python Module Index

a

Index