PiCloud Documentation

Release

Paul Crook

December 02, 2015

Contents

Table of Contents:

CHAPTER 1

Introduction

PiCloud is intended as a multi-platform, client/server software syncing suite. Files may be synced between multiple computers over the internet.

CHAPTER 2

Installation

Although subject to change, current installation is relatively simple. Well, at least for linux users. Windows support may arrive in the future.

2.1 Prerequisites

First, if you wish to truly use the program, you need at least two devices. One to setup as a server, the other as the client. Next, download the source files in your desired directory:

```
git clone https://github.com/paulcrook726/PiCloud.git
```

Once all that is done, do it with your other devices. At this point, all source files are "installed". Unfortunately, due to it still being in development, PiCloud has no installation scripts or fully-functioning cli or gui interfaces. Thus, please consider following the Usage section below.

CHAPTER 3

Usage

Follow these steps to set up and use the PiCloud source files.

3.1 Setting up the Server

Now for installing the server software. Starting from the location of the previously made command:

```
cd PiCloud/src
nano server.py
```

Of course, you may use which every editor you prefer instead of `nano`. These commands creates a file called `server.py` in the same directory of pycloud.py, which is the main module you need to import. Start by typing this into `server.py`:

```
from pycloud import *


def main():
        logging.basicConfig(format='%(asctime)s %(message)s',
                            filename='picloud.log',
                            level=logging.INFO)
        server = ServerSocket(46000)
        server.activate()



if __name__ == '__main__':
        main()
```

What this does is (1) creates a logging file for logging certain important information, (2) creates an instance of `ServerSocket`, (3) activates this socket by putting it into the main event loop. Next, save this file and exit. You may run this file now or later, your choice. Once you do, you have fully functioning server listening on port `46000`.

3.2 Setting up the Client

Setting up the client works quite similarly. Clone the repository, `cd` to `src`, and then this time make a file called `client.py`, or really whatever you feel like calling it. Put this in the file:

```
from pycloud import *


def main():
        logging.basicConfig(format='%(asctime)s %(message)s',
                            filename='picloud.log',
                            level=logging.INFO)
        c = ClientSocket(server_ip, 46000)
        f = pre_proc('your_file_here')
        send_file(c, f)
        while evaluate(c) == 0:
                pass


if __name__ == '__main__':
        main()
```

Add all this into the file except for `server_ip` and `'your_file_here'`. Instead of these, put in the actual ip of your device running the server, and the name of a file you would like to transfer in quotes and including the file extension. Once finished, simply make sure your server is running, and then run `client.py` and viola! your file has magically been copied to the other computer.

CHAPTER 4

Logs

After running your first time, you may notice that a `picloud.log` file has been created. If you open it up and view it in a text editor, it has documented the important parts of the interaction between server and client. If you ever run into problems, it is a good idea to check out the logs here.

CHAPTER 5

User Accounts

While still completely in development, a user account management system is sort of in place already. To create a "user", simply create an ID file in your client-side directory, and send it using the above method. Open an empty file, and name it whatever username you want to have, then with a .id file extension. Then write in the file your password. If you send this file to the server, the server automatically creates your user account.

CHAPTER 6

Index & Search

- genindex
- modindex
- search