Fermentrack Documentation

John Beeler

Apr 28, 2019

Contents:

FERMENTRACK

CHAPTER 1

What is Fermentrack?

Fermentrack is a web interface for controlling and monitoring fermentation temperature and progress. It interfaces with BrewPi controllers as well as specific gravity sensors like the Tilt and iSpindel in order to help you make better beer. Fermentrack is designed to be run on a Raspberry Pi, but can be used in most environments capable of running Python.

Fermentrack is a complete replacement for the web interface used by BrewPi written in Python using the Django web framework. In addition to the features bundled with brewpi-www, Fermentrack provides an easy install process, simple flashing of both Arduino-based and ESP8266-based BrewPi controllers, and the ability to control multiple chambers at once. Fermentrack also supports features such as WiFi and Bluetooth connectivity to help reduce clutter in your brewery.

To help track your wort's progress on it's journey to become beer, Fermentrack also supports multiple specific gravity sensors both on a standalone basis as well as in conjunction with BrewPi-based fermentation temperature controllers. Currently, Fermentrack supports the Tilt Hydrometer and iSpindel devices for automated gravity logging in addition to providing the ability to log manual gravity readings taken with traditional hydrometers or refractometers.

1.1 Included with Fermentrack

- **Fermentrack** - The Django-based replacement for brewpi-www. Licensed under MIT license.
- **brewpi-script** - Installed alongside Fermentrack is brewpi-script. Licensed under GPL v3.
- **BrewPi Firmware (Various)** - Fermentrack can be used to install various versions of the BrewPi firmware. Most of these are licensed under GPL v3, though other licenses may apply.
- **Other Firmware (Various)** - Fermentrack also can be used to install firmware such as iSpindel and BrewPiLess to compatible devices. Licenses may vary.

Other components used in or bundled with Fermentrack may have their own licensing requirements. These components can be referenced at *Included Components & Licensing*

1.2 Why Use Fermentrack? (New Features)

One of the key reasons to write Fermentrack was to incorporate features that are missing in the official BrewPi web interface. The following are just some of the features that have been added:

- Single Command Installation (See: *Installing Fermentrack*)
- Native multi-chamber support
- Cleaner, more intuitive controller setup
- Integrated support for ESP8266-based controllers
- Official support for "legacy" controllers
- Native support (including mDNS autodetection) for WiFi controllers
- Robust device detection for serial controllers
- Support for Tilt and iSpindel specific gravity sensors
- Support for Python 3

1.3 Requirements

All Fermentrack installations require the following:

- Raspberry Pi Zero, Zero W, 2 B, or 3 /w Internet Connection
- Fresh Raspbian install
- 1GB of free space available

Additionally, a Bluetooth receiver is required for Tilt Hydrometer support.

1.4 Getting Started with Fermentrack

Getting started with Fermentrack is incredibly easy! All you need to do is:

1. Install Raspbian on your Raspberry Pi
2. Install Fermentrack (one command!)
3. Configure Fermentrack from your web browser
4. Connect & configure your BrewPi temperature controllers or specific gravity sensors

It can be done from start to finish in a bit under an hour, assuming all your hardware is assembled & ready to go. To learn how, read *Getting Started with Fermentrack*.

1.5 Other Notes

Fermentrack is currently designed for "legacy" BrewPi firmware running on ESP8266 and Arduino hardware, and does not support "modern" firmware such as that included with current official BrewPi controllers.

A full table of controllers/expected hardware availability is available in *Supported Hardware*.

Warning: Fermentrack is currently intended to be installed on a fresh installation of Raspbian. It is **not** intended to be installed alongside brewpi-www and will conflict with the apache server brewpi-www installs. If you intend to use Fermentrack alongside an installation of Raspberry Pints or another PHP-based application, read *Legacy (PHP/Apache) Application Support (Optional)*.

CHAPTER 2

Getting Started with Fermentrack

2.1 Preparing a Raspberry Pi for Fermentrack

Prior to installing Fermentrack, you need to install Raspbian and set everything up. Click the link below to watch a video showing how to prepare the Raspberry Pi using a Mac, or read the linked instructions below for your operating system.

2.1.1 Prepare the Raspberry Pi - [Video]

1. Download the latest version of Raspbian from here. I recommend the Lite version as I prefer headless installations, but the full version works as well.
2. Burn Raspbian to your SD card using these instructions.
3. Enable SSH on your Raspberry Pi by writing an empty file named "ssh" to the root of the SD card.
4. *Optional* - Configure WiFi - See the note below if you want to configure WiFi now, thus preventing having to find an ethernet cable
5. Plug the SD card into your Raspberry Pi, connect the Pi to ethernet (if you did not configure WiFi), and plug in power.
6. Locate the IP address for your Raspberry Pi This can generally be done by executing `arp -a | grep raspberry` however you can also locate your Raspberry Pi by logging into your router and looking for the device.
7. Update the Raspberry Pi software by running `sudo apt-get update` and `sudo apt-get upgrade`.
8. Run `raspi-config` and configure the Pi. At a minimum, expand the filesystem (option 1).
9. Update the default password for the `pi` user using `passwd`
10. *Optional* - Configure WiFi on your Raspberry Pi (if needed)

2.1.2 Additional Info about Install-Time WiFi Configuration

In early 2016 there was an update made to Raspbian which allows for install-time configuration of WiFi. Buried deep within a blog post, there was the following note:

"If a wpa_supplicant.conf file is placed into the /boot/ directory, this will be moved to the /etc/wpa_supplicant/ directory the next time the system is booted, overwriting the network settings; this allows a Wifi configuration to be preloaded onto a card from a Windows or other machine that can only see the boot partition."

While not discussed in the official documentation, this greatly simplifies headless configuration (especially for Raspberry Pi zeros).

To utilize this, prior to the initial boot on a newly flashed Raspbian installation, create a `wpa_supplicant.conf` file in the `/boot` directory of the SD card with the following contents (adjusting to match your network configuration):

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
country=US
network={
    ssid="YOUR_SSID"
    psk="YOUR_PASSWORD"
    key_mgmt=WPA-PSK
}
```

Note - In the above, `ssid` is the name of your wireless network, `psk` is the password for your wireless network (if applicable), and `key_mgmt` is the password management protocol (which, for most home networks these days is WPA-PSK) You will also need to select the appropriate 2 letter country code for where you plan on using the Raspberry Pi.

Raspberry Pi 3 B+ WiFi Note

There is a race condition on the Raspberry Pi B+ where the WiFi is disabled at first boot if the above `wpa_supplicant.conf` file does not exist with a valid country code. Unfortunately, this process also can run simultaneously with the process that actually *copies* the `wpa_supplicant.conf` file to where it needs to be. This can result in the file being copied after the WiFi is disabled, permanently preventing headless configuration. If your Raspberry Pi 3 B+ doesn't appear on your network, reflash the SD card and try again.

2.2 Installing Fermentrack

There are three main ways to install Fermentrack:

- One-line fully automated
- Semi-automated
- Manual

Regardless of what method you choose, all of these expect that your Raspberry Pi has been properly set up with a working copy of Raspbian. If Raspbian is not yet installed, please read and complete *Preparing a Raspberry Pi for Fermentrack*

2.2.1 Fully Automated

The easiest way to install Fermentrack is via a single command executed on your Raspberry Pi. To install via this method, simply SSH into your Raspberry Pi and execute:

```
curl -L install.fermentrack.com | sudo bash
```

Follow the prompts as needed, and once the script completes, you're done!

You can also watch a video of this process on YouTube at: https://youtu.be/9hRH1dNygnk

2.2.2 Semi-Automated

If you prefer a slightly less automatic installation method, you can download the fermentrack-tools repo from git and use the install script contained therein. To install using this script, do the following:

1. Log into your Raspberry Pi
2. install `git` (Type `sudo apt-install git-core`)
3. Clone the repo (`git clone https://github.com/thorrak/fermentrack-tools.git`)
4. Run the install script (`sudo fermentrack-tools/install.sh`)
5. Follow the prompts on screen. Relaunch the install script if prompted.
6. Once installation completes, open a web browser and connect to Fermentrack to complete the installation process.

2.2.3 Manual

Unfortunately, the manual installation instructions haven't been fully written yet. That said, below is a general overview of what needs to happen. The steps below need to be run in order - from top to bottom. The command to run is in the right column, with a brief description of what the command seeks to accomplish in the left.

Todo: Add callout for version number here (once the next version of Fermentrack is released)

Log into your Raspberry Pi via SSH or pull up a terminal window	
Set the current user to *root*	`sudo su`
Update your apt package list	`apt-get update`
Upgrade all already installed packages via apt	`apt-get upgrade`
Install the system-wide packages (nginx, etc.)	`apt-get install -y git-core build-essential nginx redis-server avrdude bluez libcap2-bin libbluetooth3 python3-venv python3-dev python3-zmq python3-scipy python3-numpy`
Add the fermentrack user	`useradd -m -G dialout fermentrack -s /bin/bash`
Disable the fermentrack user's password	`passwd -d fermentrack`
Add the fermentrack user to the www-data group	`usermod -a -G www-data fermentrack`
Clone the Fermentrack repo	`sudo -u fermentrack -H git clone https://github.com/thorrak/fermentrack.git "/home/fermentrack/fermentrack"`
Create the virtualenv for launching fermentrack	`sudo -u fermentrack -H git clone https://github.com/thorrak/fermentrack.git "/home/fermentrack/fermentrack"`
Manually link numpy into the virtualenv	`sudo -u fermentrack -H git clone https://github.com/thorrak/fermentrack.git "/home/fermentrack/fermentrack"`
Manually link scipy into the virtualenv	`sudo -u fermentrack -H git clone https://github.com/thorrak/fermentrack.git "/home/fermentrack/fermentrack"`
Enable Python to interact with bluetooth	(See appropriate section below for instructions)
Change to the utils directory	`cd /home/fermentrack/fermentrack/utils/`
Create the secretsettings.py file	`sudo -u fermentrack -H bash "$installPath"/fermentrack/utils/make_secretsettings.sh`
Run the Fermentrack upgrade script	`sudo -u fermentrack -H bash /home/fermentrack/fermentrack/utils/upgrade3.sh`
Set up Nginx	(See appropriate section below for instructions)
Create the cron entries to launch Circus/Fermentrack	`sudo -u fermentrack -H bash /home/fermentrack/fermentrack/utils/updateCronCircus.sh add2cron`
Launch Circus/Fermentrack	`sudo -u fermentrack -H bash /home/fermentrack/fermentrack/utils/updateCronCircus.sh start`

Enabling Python to Interact with Bluetooth

Todo: Rewrite the Enabling Python to Interact with Bluetooth Section

Note: The below is the code from the automated install shell script. You will need to do something similar, man-

ually for bluetooth integration to work. This section will be rewritten at some point in the future to provide actual instructions.

```
PYTHON3_INTERPRETER="$(readlink -e $installPath/venv/bin/python)"
if [ -a ${PYTHON3_INTERPRETER} ]; then
  sudo setcap cap_net_raw+eip "$PYTHON3_INTERPRETER"
```

Setting Up Nginx

Although Fermentrack may be installed, without Nginx being configured, the app will not be accessible via a web browser.

The default-fermentrack configuration file can be found at: fermentrack-tools/nginx-configs/default-fermentrack. You will need to find and replace all instances of "brewpiuser" with "fermentrack".

Todo: Rewrite the Setting Up Nginx section

```
rm -f /etc/nginx/sites-available/default-fermentrack &> /dev/null
# Replace all instances of 'brewpiuser' with the fermentrackUser we set and save as␣
↪the nginx configuration
sed "s/brewpiuser/${fermentrackUser}/" "$myPath"/nginx-configs/default-fermentrack > /
↪etc/nginx/sites-available/default-fermentrack
rm -f /etc/nginx/sites-enabled/default &> /dev/null
ln -sf /etc/nginx/sites-available/default-fermentrack /etc/nginx/sites-enabled/
↪default-fermentrack
service nginx restart
```

2.3 Configuring Fermentrack

Once you have finished installing Fermentrack, you are ready to configure it. You will be guided through the configuration process when you first connect to Fermentrack. An overview of this configuration procedure is below.

2.3.1 User Setup

When you first access a new installation of Fermentrack you will be asked to set up a user account. This account will enable you to configure devices, configure the Fermentrack application, and view brew logs.

Setting up the user account is extremely straightforward:

1. From the root Fermentrack page, click "Continue to guided installation"
2. **On the next page, entering the following:**
    - **Username** - The username used to log into Fermentrack
    - **Password** - The password for the user account
    - **Email Address** - Currently unused, but may potentially be used later
3. Click "Create new user account"
4. Done!

You're now ready to move on to configuring the site settings.

2.3.2 Site Settings

After configuring a user account, the next step is to configure Fermentrack. The following are the available configuration options:

- **Brewery Name** - The name displayed in the upper left of each Fermentrack page
- **Date time format display** - The date format used on the dashboard of each device
- **Require login for dashboard** - *Currently Unused* - Should users be required to be logged in to *view* the dashboard/logs. Login will still be required to change temperature settings, configuration options, etc.
- **Temperature format** - The preferred (default) temperature format. Used when setting fermentation temperature profiles. Can be overridden per device.
- **Preferred timezone** - The preferred timezone you would like graphs displayed in. *Note* - All data is stored in UTC (GMT) - you are only selecting how the data will be *displayed*. Feel free to change this at any time with no issues.
- **Git Update Type** - Which releases you will be prompted to upgrade to. Can be set to "tagged versions" (which will generally be tested and stable), "all commits" which will include all new code releases, and "None".

All of these can be updated at any time by clicking on the "gear" icon in the upper right of any Fermentrack page.

2.3.3 Notes for Advanced Users

User Accounts

Currently, Fermentrack has limited access control implemented, and is not yet designed for multiple user installations. This feature is planned for a later release - once it becomes available, different levels of access will be able to be specified on a per-user basis.

In the mean time if you need multiple user accounts they can be configured using the Django admin interface (accessible via the "gear" icon). Each account will need "superuser" access to be able to use the full functionality of Fermentrack. Again - please keep in mind - multiple user access is not officially supported. When access control functionality is implemented, any users created previously through this method will retain full access/control of Fermentrack.

Advanced Site Settings

There are a handful of additional site settings which are intended for advanced users and developers only. These settings can only be accessed via the Django admin. These settings include:

- **Allow Git Branch Switching** - Allows switching to a different git branch from within Fermentrack
- **User Has Completed Configuration** - Determines if the user has completed the initial configuration workflow, or should be prompted to set Fermentrack up on next login.

Additionally, graph colors can be configured via the Django admin. The graph color options include:

- **Graph Beer Set Color** - The color of the "Beer Setting" line
- **Graph Beer Temp Color** - The color of the "Beer Temperature" line
- **Graph Fridge Set Color** - The color of the "Fridge Setting" line
- **Graph Fridge Temp Color** - The color of the "Fridge Temp" line
- **Graph Room Temp Color** - The color of the "Room Temp" line
- **Graph Gravity Color** - The color of the "Specific Gravity" line

2.4 Legacy (PHP/Apache) Application Support (Optional)

Unlike apps such as RaspberryPints and BrewPi-www which use Apache to serve webpages, Fermentrack uses nginx. If you wish to run applications other than Fermentrack on the same Raspberry Pi you will need to configure nginx to serve those applications instead of Apache.

Fermentrack-tools includes a script which can be used to install this support automatically.

> **Warning:** Support for php-5 was discontinued in the latest versions of Raspbian. You may need to use an older version of raspbian to obtain support for apps like RaspberryPints designed for php-5.

2.4.1 Understanding Legacy Support

To support legacy applications, the fermentrack-tools script does the following:

- Install `php5-common`, `php5-cli`, and `php5-fpm` to allow Nginx to serve php files
- Disable `apache2` from launching at startup
- Create a new nginx configuration file serving webpages from `/var/www/html` on port 81

Note - Due to the port change mentioned above, any apps that were previously running at `http://<your-ip>/` will now be running at `http://<your-ip>:81/`

2.4.2 Installation

Although fermentrack-tools offers a script to allow for fully automated installation of support for PHP/legacy apps, support can be installed manually as well.

Automated Installation

To run the fully automated installation script, simply SSH into your Raspberry Pi and execute:

```
curl -L install-legacy-support.fermentrack.com | sudo bash
```

Manual Installation

To manually install legacy app support, you will need to do the following as root:

1. Install PHP5 support - `apt-get install php5-common php5-cli php5-fpm`
2. Disable `apache2` from running at startup - `update-rc.d apache2 disable`
3. Disable any currently running instance of `apache2` - `service apache2 stop`
4. Install the appropriate configuration file to `nginx`. An example configuration file can be found here, and must be installed in `/etc/nginx/sites-enabled`
5. Restart PHP5-FPM - `service php5-fpm restart`
6. Restart Nginx - `service nginx restart`

2.4.3 Legacy BrewPi-www Installation Support

Although performing the above actions will allow brewpi-www to run alongside Fermentrack, doing so is not recommended. Attempting to run brewpi-www in this way can result in issues as Fermentrack and brewpi-www compete to access/manage your fermentation controller.

2.5 Installing Raspbian to a Raspberry Pi using Windows 7

here

2.5.1 Download and Install Raspbian

1. Download the latest version of Raspbian here. You can download either the Lite or Full version - the Lite version is good for "headless" setups where you won't have a monitor & keyboard hooked up to your Raspberry Pi, the full version includes a graphical interface for use with a monitor/keyboard.
2. Download and install Etcher as recommended here.

3. Burn Raspbian to your SD card using Etcher: 3.1 Connect the SD card you will be installing Raspbian onto to your Windows PC using a removable SD card adaptor 3.2 Select the proper Raspbian .zip file 3.3 Select the proper removable drive to flash (Etcher only allows you to select removable drives) 3.4 Flash to the SD Card 3.5 Navigate to and open the SD Card to verify files were flashed

4. Enable SSH on your Raspberry Pi by writing an empty file named "ssh" to the root of the SD card via the Notepad Windows Program: 4.1 Run Notepad 4.2 In a new file, put in one space and nothing else 4.3 Click File > Save As, and save the file to the root (lowest level) directory on the SD Card: 4.3.1 Name the file `ssh` 4.3.2 NOTE - Be certain to Save as type: All Types (* *) 4.4 Close the file

5. Configure WiFi (Optional, but required if running the Lite/Headless Version and you do not plan to connect the Raspberry Pi via Ethernet Cable): 5.1 Run Notepad 5.2 In a new file, paste the contents below. (Inserting the proper country code, network name, and network password) Network Names with some symbols may be problematic. If you have issues connecting, eliminate your SSID from having any unusual symbols. When entering your network name and password include the Quotes.

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
    ssid="NETWORK-NAME"
    psk="NETWORK-PASSWORD"
}
```

5.3 Click File > Save As, and save the file to the root (lowest level) directory on the SD Card: 5.3.1 Name the file `wpa_supplicant.conf` 5.3.2 Be certain to Save as type: All Types (* *) 5.4 Close the file

6. Verify that the `ssh` and `wpa_supplicant.conf` files were created in the Root Directory. 6.1 Open My Computer (File Explorer) 6.2 Navigate and open the removable SD Card containing Raspbian 6.3 Confirm that the ssh file and the wpa_supplicant.conf file can be seen 6.4 Navigate off the SD Card file. 6.5 Eject SD Card properly.

7. Remove the SD Card and plug the SD Card into your Raspberry Pi and plug in power to the Raspberry Pi. (If you did not configure WiFi, connect the Pi to Ethernet, plug in power) **NOTE - Give time to boot, this can take 60 seconds or longer**

2.5.2 Configure your Raspberry Pi

8. Download and install Putty (http://www.putty.org/). (This next step assumes that ssh is enabled on the Raspbian Image (Step 4) and that you properly created the wpa_supplicant.conf file.(Step 5))

Step 9. Login over WiFi/Ethernet using Putty. (Your Windows device MUST be on the same WiFi Network as you configured your Pi for) 9.1 Launch Putty 9.2 Set the Host Name (or IP address) Field to: `raspberrypi.local` (you can also log into your router, look for the device, and enter the correct IP address into Putty) 9.3 By default the Port should be set to 22 and the connection type should be set to SSH. 9.4 Click Open 9.5 If you see a Security Alert, select Yes. 9.6 A new terminal window will pop open prompting you for a user name. 9.7 The default user name is: `pi` 9.8 The default password is: `raspberry`

(You can now access your Pi via WiFi)

10. Update the Raspberry Pi Software. 10.1 Run the command `sudo apt-get update -y` on the Raspberry Pi using SSH (Putty) 10.2 Run the command `sudo apt-get upgrade -y`

11. Configure Raspberry Pi (RPi 3 B+ with Raspbian Stretch Lite shown) 11.1 Run the command `sudo Raspi-config` 11.2 Change User Password from default `raspberry` • Option #1 • Enter new password 11.3 (Optional) Change Hostname from default `raspberrypi`. • Option #2 • Option N1 • Enter new Hostname • (NOTE - Changing the Hostname will alter how you login via Putty. If Hostname is changed, in step 9.2 you will need to enter the new hostname similar to `newhostname.local` and on step 9.8 you would need to enter your new password. 11.4 (Optional, if needed & not done earlier using `wpa_supplicant.conf`) Configure WiFi • Option #2 • Option N2 • Follow Prompts 11.5 Reboot

Raspberry Pi is now ready for Fermentrack Install.

2.5.3 (Optional) Set up additional, optional networks

::todo Change this to the main Raspi Setup file

If you move your Raspberry Pi around often, or potentially need to connect to multiple networks, you can configure the `wpa_supplicant.conf` file to contain multiple network options. To do so, do the following:

1. Connect Raspberry Pi and Windows based hardware on the same network as original installation.
2. Launch Putty and login to Raspberry Pi.

3. Determine additional Networks to configure to. (If you don't know, run the following command to locate local networks) 3.1 Run the command `sudo iwlist wlan0 scan` 3.2 Record the ESSID you want to connect to (you will need to know the password)

4. Add the network details to the Raspberry Pi 4.1 open the `wpa_supplicant.conf` file • Run the command `sudo nano /etc/wpa_supplicant/wpa_supplicant.conf` 4.2 Find the following from Installation Step 5.2:

```
network={
    ssid="NETWORK-NAME"
    psk="NETWORK-PASSWORD"
}
```

4.3 Add priority and Network ID to original network configuration.

```
network={
    ssid="NETWORK-NAME"
    psk="NETWORK-PASSWORD"
    priority=1
}
```

4.4 Add additional Networks under your main and set priority.

::

network={ ssid=”additional-network-name” psk=”additional-network-password” priority=2

}

network={ ssid=”Secondary-Network-Name” psk=”Secondary-Network-Password” priority=3

}

4.5 Save your New Network Configuration. (Press the following) • Ctrl + x • Y • Enter 4.6 Reboot the Pi by running `sudo shutdown -r now`

5. Confirm that Raspberry Pi is on Priority 1 Network 5.1 Launch Putty and login in to Raspberry Pi • If connected successfully, congratulations! • If unsuccessful: • Make certain RaspberryPi and Windows hardware are on the same network. • Log Windows/Pi devices into the original network to see if connection can be made. • If Raspberry Pi is lost and can’t be connected to, wipe SD card and start the installation process over.

Fermentrack is now ready to be configured.

CHAPTER 3

User Guide

3.1 Flashing a Controller

Fermentrack includes software which allows you to easily download & flash firmware to controllers. This feature is currently supported for the ESP8266 and Arduino architectures, and supports multiple families of firmware including BrewPi, BrewPiLess, and iSpindel.

All devices will need be flashed via serial over USB - including devices you ultimately want to use over WiFi.

See also:

A video of this process can be seen on YouTube at https://youtu.be/vpm-8k8tnGk

3.1.1 Accessing the Flash Workflow

Flashing a new controller is accomplished through the controller flashing interface. This can be accessed directly or through the guided device setup workflow. From the device menu, choose "Flash Controller".

3.1.2 Flashing a Controller

1. Once in the flash workflow, click "Refresh Firmware List from Fermentrack.com". This downloads a list of availble, screened firmware from Fermentrack.com.
2. After the list downloads, select your device family and click "Submit".
3. On the next screen, select the board (hardware variant) your device is based on. For some families, there may only be one option.
4. Double check your device family & board on the following screen and then ensure that the device you want to flash is **not** connected to the computer/device running Fermentrack. Once this is done click "Scan Devices"
5. Review the "Preexisting Devices" list to ensure that the device you want to flash is *not* listed. Assuming this is the case, connect the device you want to flash to the device running Fermentrack via a USB cable. Click "Scan for New Devices"

6. Your device should now be detected and displayed in the list. If it isn't, return to an earlier step and restart the process. Click the "Set Up" button next to your device.
7. On the following screen, select the firmware you want to flash to the device. Click "Flash to Device".
8. Once the firmware has flashed successfully, your device is ready to use!

3.2 Adding a BrewPi Controller to Fermentrack

Setting up a controller that is running the BrewPi firmware is easy, and Fermentrack will guide you every step of the way. If you prefer to jump straight in and set the controller up manually, Fermentrack supports that too. Just jump ahead to *Setting up a controller using the Advanced (Manual) Workflow*.

All of these instructions assume that you have already flashed the relevant firmware to your controller. If you just built it and need to flash it, read *Flashing a Controller* and complete that process before continuing.

Warning: Prior to setting up a controller with Fermentrack, please read any notes specific to your controller's hardware on the :doc:hardware page.

3.2.1 Setting up a WiFi-connected controller using the Guided Workflow

1. After the controller is powered on and connected to your network, launch guided setup and select "Add New Device (Guided)" from the "Select Device to Control" dropdown
2. Select the correct board type (ESP8266) from the dropdown and click "Submit"
3. If your device is already flashed, choose "Yes - Already Flashed". If it isn't, read *Flashing a Controller* before continuing.
4. Select "WiFi" on the left, and then click "Scan WiFi via mDNS"
5. Select the appropriate device from the "Available (Uninstalled) Devices" list, and click "Set Up"
6. Enter a name for the device, adjust settings as needed, and click "Submit"

3.2.2 Setting up a serial-connected controller using the Guided Workflow

Note: When setting up a controller to connect via serial, the selected "port" is incredibly important, but is prone to change on reboot or as other devices are connected. Please read *About Serial Port Autodetection* for information on how Fermentrack handles this issue.

1. With the controller disconnected from the Raspberry Pi, launch guided setup and select "Add New Device (Guided)" from the "Select Device to Control" dropdown
2. Select the correct board type from the dropdown and click "Submit"
3. If your device is already flashed, choose "Yes - Already Flashed". If it isn't, read *Flashing a Controller* before continuing.
4. If setting up any device other than an ESP8266 click "Begin Serial Autodetection". If setting up an ESP8266, select "Serial" on the left, and then click "Begin Serial Autodetection"
5. Ensure that the controller **is not** connected to the Raspberry Pi, and click "Scan Devices"

6. Connect the controller to the Raspberry Pi
7. Click "Scan for New Devices"
8. Choose the device that corresponds to your Arduino and click "Set Up"
9. Enter a name for the device, adjust settings as needed, and click "Submit"

About Serial Port Autodetection

By default, Linux assigns serial ports to devices like BrewPi controllers based on the order they are connected. If you have multiple devices connected to your Raspberry Pi this can mean that Fermentrack could potentially mistake one device for another. To prevent this, Fermentrack includes a feature which will disregard the specified serial port when connecting to the controller and will instead connect to the device with the USB serial number that matches the device you set up.

If you wish to disable this feature and instead only connect to the specified serial port, uncheck the "Prefer Connecting Via Udev" option in the "Manage Device" screen.

This feature only works on Linux-based operating systems (including Raspbian for Raspberry Pi), and may not work if Fermentrack is installed on a Mac or Windows-based computer.

3.2.3 Setting up a controller using the Advanced (Manual) Workflow

Note: When setting up a controller to connect via serial, the selected "port" is incredibly important, but is prone to change on reboot or as other devices are connected. Please read *About Serial Port Autodetection* for information on how Fermentrack handles this issue.

Setting up a controller using Advanced Workflow

1. Connect the controller to the Raspberry Pi
2. Launch guided setup and select "Add New Device (Advanced)"
3. Enter the configuration options associated with the device.

Todo: Rewrite the Manual Workflow section to document the available options, etc.

3.3 BrewPi Controller Configuration

Once your BrewPi controller is set up within Fermentrack it's ready for configuration. If you've already configured your controller on either another installation of Fermentrack or an installation of brewpi-www, then you may already be ready to go. Otherwise, just follow these instructions to get on your way.

1. Go to your Fermentrack installation's main page, and click "View Dashboard" underneath the device you wish to configure.
2. From the menu at the top with your device's name, choose "Configure Sensors/Pins/Devices"
3. A screen will appear showing you the configuration status of your device. At a minimum, you need to have a heating pin, cooling pin, and chamber sensor set for your device to operate.
4. Configure each of the devices listed under "Available Devices" - so long as

3.3.1 Heating/Cooling/Door Sensor Pin

Heating/cooling are configured by pin number. In most cases the correct pin will be prescribed, with the function listed out in the name. (On ESP8266 devices, for example, you will see a header similar to "Pin 16 (D0 (Heat))". This is generally the "heat" pin.)

To configure, simply choose the device function from the drop down menu (Heating Relay, Cooling Relay, etc.), set whether or not your relay requires the pin to be inverted (for most mechanical relays, you want Invert Pin to be "Inverted") and click "Assign"

Warning: While the door sensor pin will generally appear as available to configure, you should not set this pin unless you have an actual door sensor connected to your controller. If you set this up incorrectly your controller may think the door to your fermentation chamber is always open, and fail to trigger heating/cooling as expected.

3.3.2 Temperature Sensors

OneWire temperature sensors are configured by device address rather than the pin they are connected to. Connect your OneWire sensors to your BrewPi controller, wait a few seconds, and refresh the page. You should see all connected sensors listed under either Available Devices or Installed Devices.

To install an available temperature sensor, simply choose the device function (Chamber Temp, Room Temp, Beer Temp, etc.) and click "Assign". Your BrewPi controller will configure the device appropriately.

3.3.3 Other Options (Chamber fan, chamber light, etc.)

While the BrewPi controller firmware allows for other options to be selected such as chamber fan, chamber light, etc. these options don't actually do anything and should be ignored.

3.4 Gravity Sensor Setup

Todo: Write the Gravity Sensor Setup page

3.4.1 Tilt Hydrometer

3.4.2 iSpindel

3.4.3 Manual Sensor

CHAPTER 4

Changelog, Licensing, and Development

4.1 Fermentrack Architecture

Todo: Rewrite/fix architecture.rst to match the mkdocs/markdown version

The Fermentrack stack is based on a front end application, a controller, and a firmware running on the device that handles reading temperatures, switching cooling and heating etc. Everything but the firmware part is running under a process manager which takes care of launching the front end and brewpi.py controller scripts.

![Fermentrack Architecture](img/fermentrack.png)

See [components](components.md) documentation for links and licenses.

The webserver nginx and chaussette WSGI server

Used to proxy http requests to chaussette over WSGI to the Fermentrack django application.

cron

Used to start the Fermentrack stack, it starts the Circus process manager via a @reboot job, it also checks the status of circus every 10 seconds, if it not running it will start it. All this is handled by a script: *updateCronCircus.sh*

Supports the following arguments: *{start|stop|status|startifstopped|add2cron}* where:

- *start* - will start circusd and all the services
- *stop* - will quit circusd and all processes (note it would be started again in 10 minutes)
- *status* - will output a status of all processes running (see below)
- *startifstopped* - will start the process manager if stopped (called from cron every 10 minutes)
- *add2cron* - if crontab entries are missing, it will add them back.

Crontab entries added with *add2cron*:

> @reboot ~/fermentrack/brewpi-script/utils/updateCronCircus.sh start **/10 * * * * ~/fermentrack/brewpi-script/utils/updateCronCircus.sh startifstopped

Example *status* output:

> $ ~/fermentrack/brewpi-script/utils/updateCronCircus.sh status Fermentrack: active brewpi-spawner: active circusd-stats: active dev-brewpi1: active

The process manager *circus*

Fermentrack is started at boot with the help of cron (see *cron*), the process manager handles all the different processes needed by Fermentrack.

- **Fermentrack** - The django application (web interface) runs under chaussette
- **brewpi-spawner** - An internal Fermentrack process for spawning controller scripts for controlling controllers like brewpi-esp8266.
- **circusd-stats** - An Internal circus process for stats, not used yet.
- **dev-brewpi1** - Is a controller script spawned by brewpi-spawner, handing a controller.

Circus documentation can be found [here](https://circus.readthedocs.io/en/latest/).

Logging

- **Circus process manager logs:**
  - */home/fermentrack/fermentrack/log/circusd.log*
- **Controller script (brewpi.py) log:**
  - */home/fermentrack/fermentrack/log/dev-[name]-stdout.log*
- **Controller script (brewpi.py) error/info log:**
  - */home/fermentrack/fermentrack/log/dev-[name]-stderr.log*
- **Controller script spawner:**
  - */home/fermentrack/fermentrack/log/fermentrack-brewpi-spawner.log*
- **Fermentrack django application:**
  - */home/fermentrack/fermentrack/log/fermentrack.log*

Logs are rotated every 2MB and the last 5 are saved with a number suffix.

4.2 Changelog

All notable changes to this project will be documented in this file.

The format is based on [Keep a Changelog](http://keepachangelog.com/) because it was the first relatively standard format to pop up when I googled "changelog formats".

4.2.1 [2019-03-31] - TiltBridge Support

Added

- Added support for TiltBridge Tilt-to-WiFi devices

Changed

- Removed Hex SHA display on GitHub update
- Tweaked backup count for log files to reduce clutter

Fixed

- Fixed hostname lookup in connection debug when running on a nonstandard port
- Fixed multipart firmware flashing
- Remove Git branch switching prompt during initial setup
- Remove links to defunct Tilt logs
- Fixed OneWire address display on BrewPi "Assign Pin/Device" page
- Fix link to "load beer log" modal on device dashboard when no beer is loaded

4.2.2 [2019-03-17] - Firmware Flash Changes

Added

- Added support for flashing multi-part firmware (eg partition tables)

Changed

- Updated firmware_flash models to support additional device families
- Changed to version 2 of firmware_flash models

4.2.3 [2019-02-17] - External Push (Remote Logging) Support

Added

- Fermentrack can now periodically "push" readings out to an external device/app
- Added "new control constants" support for "modern" controllers

Fixed

- Explicitly linked Favicon from template
- Fixed BrewPi-Script error when attempting to use feature not available in Python 3.4
- Properly catch error in BrewPi-Script when pidfile already exists
- Added filesize check for gravity sensor & brewpi-device logfiles
- Add support for temperature calibration offsets

4.2.4 [2019-02-17] - External Push (Remote Logging) Support

Added

- Fermentrack can now periodically "push" readings out to an external device/app
- Added "new control constants" support for "modern" controllers

Fixed

- Explicitly linked Favicon from template
- Fixed BrewPi-Script error when attempting to use feature not available in Python 3.4
- Properly catch error in BrewPi-Script when pidfile already exists
- Added filesize check for gravity sensor & brewpi-device logfiles
- Add support for temperature calibration offsets

4.2.5 [2018-10-24] - Tilt Monitor Refactoring

Changed

- The Tilt Hydrometer monitor now uses aioblescan instead of beacontools for better reliability
- Added support for smaller screen sizes

Fixed

- Tilt Hydrometers will now properly record temperatures measured in Celsius

4.2.6 [2018-08-05] - Gravity Refactoring

Added

- DS18b20 sensors can now have temperature offsets added to each reading to correct for calibration errors
- ESP8266 controllers can now have their WiFi settings reset via the "manage sensor" web interface
- Control constants form now supports both "new" (OEM BrewPi) and "old" ("Legacy" branch) control constants
- Tilt hydrometers can now have their specific gravity readings calibrated
- "Heat/Cool State" will now be shown on temperature graphs
- Fermentrack logo added as favicon

Changed

- The iSpindel endpoint can now be accessed at either /ispindel or /ispindle
- Specific gravity will now be shown on graphs with 3 decimal places
- Beer log format has been changed to add state information

Fixed

- Removed constant LCD polling for "modern" controllers
- Gravity support will now be properly disabled when the correct flag is set at setup
- iSpindel devices that do not report all 'extras' will no longer throw errors when reporting gravity

4.2.7 [2018-04-27] - "v1.0 release"

Added

- Added fermentation controller "Manage Device" page
- Upgrades are now logged to upgrade.log
- Controller "stdout" and "stderr" logs are now saved/accessible
- Support for serial devices
- Support for Arduino-based devices
- Support for in-app git branch switching
- Autodetection of serial devices
- Huey (delayed/scheduled task) support (currently unused)
- Controllers connected via serial can now have their serial port autodetected using the udev serial number
- Beer profiles are now displayed in graph form
- Firmware can now be flashed to new Arduino & ESP8266-based controllers from within the app
- Preferred timezone can now be selected for use throughout Fermentrack
- Beer log management (deletion/downloading)
- Added configuration options for graph line colors
- Graph lines can be toggled by clicking the icon in the legend
- Added support for specific gravity sensors
- Added support for Tilt Hydrometers
- Added support for iSpindel specific gravity sensors

Changed

- Inversion flag for installed devices is now shown on the "configure pins/sensors" page
- Form errors are now displayed on "configure pins/sensors" page
- Beer logs are no longer deleted along with the parent device (but they will become inaccessible from within Fermentrack)
- GitHub updates are no longer triggered automatically by visiting the update page, and must now be manually triggered by clicking a button
- The IP address of a BrewPiDevice is now cached, and can be used if mDNS stops working
- At end of a fermentation profile the controller will now be switched to beer constant mode
- All data points are now explicitly recorded in UTC

- Added icon to graph legend to display line color
- Updated to Django v1.11 (Long term support version)
- Changed from supporting Python 2 to Python 3

Fixed

- Inversion state no longer improperly defaults
- Minimum graph size adjusted to account for smaller displays
- Changed on_delete behavior to allow deletion of fermentation controllers
- Git update check will now properly wait between checks if up to date
- GIT_UPDATE_TYPE of 'none' will now properly disable update checks
- BrewPi controllers now accept unicode names
- "View Room Temp" link on Dashboard now functions
- Room temp now included in legend for graphs

4.2.8 [2017-03-17] - "v0.1 release"

Added

- First release!

4.3 Included Components & Licensing

Fermentrack is licensed under the MIT License, the terms of which can be read here: *License*

Fermentrack contains a number of JavaScript, Python packages, CSS, and other components to help provide additional functionality which are provided under their own licenses. Some of these packages are listed below.

4.3.1 Python Packages

In addition to Django, this app utilizes a number of Python packages. These packages include:

Package	License
Django	BSD 3 Clause
configobj	BSD 3 Clause
pyserial	BSD 3 Clause
huey	MIT (Expat)
raven	BSD 3 Clause
django-constance	BSD 3 Clause
GitPython	BSD 3 Clause
pytz	MIT (Expat)
redis	MIT (Expat)
zeroconf	LGPL v2
pyudev	LGPL v2.1
circus	Apache Public License v2
circus-web	Apache Public License v2
chaussette	Apache Public License v2
pid	Apache Public License v2
aioblescan	MIT (Expat)

4.3.2 JavaScript Packages

Fermentrack provides some of its functionality using JavaScript. Some of the third party JavaScript packages used within Fermentrack include:

Package	License
jQuery	MIT (Expat)
vue.js	MIT (Expat)
Dygraph	MIT (Expat)
Moment	MIT (Expat)
Moment Timezone	MIT (Expat)
Respond.js	MIT (Expat)
html5shiv.js	MIT (Expat)

4.3.3 UI (CSS, Fonts, etc.) Packages

In addition to Python and JavaScript packages, Fermentrack utilizes a handful of third party images, icon packs, CSS files, and fonts in rendering the user interface. Some of these include:

Package	License
FontAwesome (Font)	SIL OFL 1.1
FontAwesome (CSS)	MIT (Expat)
Bootstrap	MIT (Expat)
FlatUI	MIT (Expat)
5x8 LCD Font	SIL OFL 1.1

4.3.4 Separate Applications

Fermentrack is designed to help install or manage communications with certain key applications. These applications are not incorporated into Fermentrack but may be bundled with the software as a convenience.

Package	License
BrewPi-Script	GPL v3
BrewPi Firmware	GPL v3 (and potentially others)
Avrdude	GPL v2
Fuscus	GPL v3

4.4 Contributing to Fermentrack

Want to help? Awesome! There are a number of ways you can get involved in this project. To help you get started, some areas in which you can contribute are listed below along with additional resources for getting started.

4.4.1 Reporting Bugs/Issues

Found a bug or issue (or have a suggestion for how to improve)? Awesome!

The best way to reach out is to either open an issue on GitHub or discuss your issue or idea in the HomeBrewTalk thread. Both of these are actively monitored, and will help to keep track of progress towards a resolution.

4.4.2 Developing Fermentrack

Interested in helping develop Fermentrack (or improving its look & feel)? Fantastic - Fermentrack is an open source project, and all help is welcome.

Fermentrack is a Python-based application which uses the Django framework. It is open source and is managed on GitHub. To help you get started, take a look at the developer documentation located within *Changelog, Licensing, and Development*. After reading, if you have questions don't hesitate to reach out.

4.4.3 Documenting Fermentrack

A project like Fermentrack is only as good as its documentation. Documentation for Fermentrack is (currently) written in reStructuredText with an eye towards Sphinx. The source for the documentation is located in the `docs/source` folder in the Fermentrack repo on GitHub.

If you are familiar with GitHub, pull requests that include documentation fixes are always welcome. If not, reach out on GitHub or the HomeBrewTalk forums and we'll be happy to help get things updated.

4.5 "Push" Support

Although Fermentrack is focused on the "fermentation" phase of your brewing operation, Fermentrack is designed to integrate with your brewing operation as a whole. To support the use of data collected by Fermentrack in other applications, Fermentrack allows for data to be "pushed" on a periodic basis via HTTP requests (and will - in the future - support pushing via TCP (sockets)).

4.5.1 Supported "Push" Targets

Fermentrack currently supports one push target, though more are likely to be added in the near future:

- "Generic" Push Target - Fermentrack's "native" push format - Pushes both specific gravity & temperature data

Generic Push Target Format

The "generic" push target format is the one recommended for use by developers who are adding native Fermentrack support to their apps. This format contains temperature/gravity data collected by Fermentrack for each available specific gravity sensor and BrewPi controller.

This format is supported by the Fermentrack Push Target app for testing/development purposes.

```
{'api_key': 'abcde',
 'brewpi_devices': [{'control_mode': 'f',
                     'internal_id': 1,
                     'name': 'Legacy 2',
                     'temp_format': 'F'},
                    {'beer_temp': 31.97,
                     'control_mode': 'f',
                     'fridge_temp': 36.26,
                     'internal_id': 2,
                     'name': 'Kegerator',
                     'temp_format': 'F'}],
 'gravity_sensors': [{'gravity': 1.247,
                      'internal_id': 1,
                      'name': 'Pinky',
                      'sensor_type': 'tilt',
                      'temp': 78.0,
                      'temp_format': 'F'},
                     {'internal_id': 3,
                      'name': 'Spindly',
                      'sensor_type': 'ispindel',
                      'temp': 86.225,
                      'temp_format': 'F'}],
 'version': '1.0'}
```

4.5.2 Implementation Notes

Push support within Fermentrack is implemented through the use of a "helper script" which currently is launched once every minute. The helper script polls the defined push targets to determine which (if any) are overdue for data to be pushed, and - for those targets - then executes the push based on how those targets are configured. Fermentrack polls Redis for the latest available data point for specific gravity sensors, and polls BrewPi controllers for the latest data point directly. This data is then encoded based on the selected push format and sent downstream to the requested target.

Push requests are handled asynchronously. Due to the way that the polling script is implemented, push "frequencies" may be up to one polling cycle (currently 1 minute) later than expected. For 1 minute push cycles, this means that the actual frequency could be as high as 2 minutes.

4.5.3 Feedback

Push support was designed to support future applications that do not yet exist, and as such, may not be perfect for *your* application. That said, feedback is always appreciated and welcome. Feel free to reach out (HBT forums, GitHub, Reddit...) if you have something in mind that you'd like to integrate Fermentrack into, and don't think the existing push options will quite work.

4.6 License

```
MIT License

Copyright (c) 2016-2018 John Beeler
Copyright (c) 2016-2018 Fredrik Steen

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 5

Supported Hardware

Fermentrack supports both BrewPi-based temperature controllers as well as various specific gravity sensors. Support for each family of hardware varies, but is expanding with each release.

5.1 BrewPi Controllers

5.1.1 ESP8266-Based Controllers

Fermentrack was explicitly built to support ESP8266-based controllers regardless of connection method (WiFi or Serial).

For more information on ESP8266-based firmware, please refer to one of the following:

- BrewPi-ESP8266 GitHub
- BrewPi-ESP8266 HomeBrewTalk Thread

If connecting an ESP8266-based controller via serial, please note that by default Fermentrack will detect the USB serial number associated with your ESP8266 when initially configured, and will use that *instead of the specified serial port* to connect. For more information, read *About Serial Port Autodetection*.

5.1.2 Arduino-Based Controllers

Fermentrack currently has native support for Arduino-based controllers connected via Serial (USB).

By default, Fermentrack will detect the USB serial number associated with your Arduino when initially configured, and will use that - instead of the specified serial port - to connect. For more information, read *About Serial Port Autodetection*.

Bluetooth Support

There is a third party project which looks to add bluetooth support for Arduino (and similar) controllers to BrewPi. While this project is not natively supported from within Fermentrack, at the end the project presents the BrewPi controller as a device connected via serial which allows it to be set up within Fermentrack via the Manual Setup workflow.

5.1.3 Native Python (Fuscus)

Fuscus is a project which implements the legacy (Arduino) BrewPi featureset directly on the Raspberry Pi with no need for an external controller.

As of now, serial connections are supported by Fermentrack, and therefore it is expected that Fuscus should be Fermentrack compatible. Fuscus support *has not been tested* and should be considered experimental.

Due to the nature of the serial ports used by Fuscus, the serial autodetection process cannot be used to set up a Fuscus-based controller. To set up, please follow the instructions in *Setting up a controller using the Advanced (Manual) Workflow*.

Note: When setting up a Fuscus-based controller in the manual workflow, make sure to set "prefer_connecting_via_udev" to False. If it is set to true, BrewPi-Script may either not connect or connect to the wrong device.

Further down the development path are other features involving Fuscus such as direct installation and configuration management - though these are expected in v3 and beyond.

5.1.4 Spark Core/OEM Controllers

Currently, Fermentrack does not support Spark-based controllers. Support for Spark based controllers is planned, but will be implemented at a much later date. Once implemented, Fermentrack will support controlling both legacy (Arduino/Fuscus) and modern (Spark) controllers from the same installation.

5.2 Specific Gravity Sensors

5.2.1 Tilt Hydrometer

The Tilt Hydrometer is supported natively by Fermentrack which will assist with device setup. Tilt Hydrometers are easy to use, and can be installed either alongside or independent of a temperature controller. Natively, Tilt Hydrometers communicate via Bluetooth, however they can also be connected via the TiltBridge Bluetooth-to-WiFi adaptor.

5.2.2 iSpindel

iSpindel devices are directly supported by Fermentrack which can assist with device configuration & calibration, as well as the initial flashing of the iSpindel firmware.

CHAPTER 6

Frequently Asked Questions

6.1 Can I change the temperature format on a beer I've started logging?

No. To prevent inconsistency the log format is permanently set when logging begins to the temperature format associated with the device. If you would like to change the format and restart logging, do the following:

1. Update the temperature format in control constants to the desired format
2. Stop logging the existing beer
3. Start logging a new beer

6.2 Help - I forgot my Fermentrack login/password!

Thankfully, this is a pretty easy issue to overcome. Django provides the `manage.py` command line script which contains the `createsuperuser` command. To leverage this, do the following (assuming the standard install locations):

1. Log into your Raspberry Pi via ssh and switch to the user you installed Fermentrack to (generally this can be done with the command `sudo -u fermentrack -i` assuming you installed to the `fermentrack` user)
2. Change to the user's home directory (`cd ~`)
3. Enable the virtualenv (`source venv/bin/activate`)
4. Change to the Fermentrack directory (`cd fermentrack`)
5. Run the createsuperuser command (`./manage.py createsuperuser`)
6. Follow the prompts to create a new superuser account
7. Log into the Fermentrack admin panel and delete/modify the old account. The Fermentrack admin panel can be accessed through the `Settings` page (the gear in the upper right) and clicking the "Django Admin" button.

6.3 What happens to my beer logs/active profiles/other data if I change the Fermentrack "Preferred Timezone"?

Not much. To prevent this being an issue Fermentrack uses UTC (GMT) internally and converts times to your local timezone on the fly. Feel free to update your preferred timezone as you move, travel, or are otherwise inclined without worrying about how this might impact your existing logs or active profiles.

CHAPTER 7

Documentation To-Do List

This file is automatically generated, and contains a list of everything that has been noted as a "To Do" in the documentation.

If you're looking for a way to help with the Fermentrack documentation, this is a great place to start. You can help by either:

1. Directly addresing & clearing items from the list below
2. Adding new items (using the `..todo ::` directive) where additional documentation is needed.

Below are the items which need to be addressed:

Todo: Rewrite/fix architecture.rst to match the mkdocs/markdown version

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/fermentrack/checkouts/master/docs/source/develop/a line 4.)

Todo: Add callout for version number here (once the next version of Fermentrack is released)

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/fermentrack/checkouts/master/docs/source/getting started/install.rst, line 47.)

Todo: Rewrite the Enabling Python to Interact with Bluetooth Section

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/fermentrack/checkouts/master/docs/source/getting started/install.rst, line 100.)

Todo: Rewrite the Setting Up Nginx section

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/fermentrack/checkouts/master/docs/source/getting started/install.rst, line 120.)

Todo: Figure out how to include the text in about.rst on this page in the HTML version only

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/fermentrack/checkouts/master/docs/source/index.rst, line 52.)

Todo: Rewrite the Manual Workflow section to document the available options, etc.

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/fermentrack/checkouts/master/docs/source/user guide/controller adding.rst, line 76.)

Todo: Write the Gravity Sensor Setup page

(The original entry is located in /home/docs/checkouts/readthedocs.org/user_builds/fermentrack/checkouts/master/docs/source/user guide/gravity sensor setup.rst, line 6.)

Todo: Figure out how to include the text in about.rst on this page in the HTML version only