
Kegbot Documentation

Bevbot LLC

Aug 10, 2022

Contents

1	Kegbot Server Overview	3
1.1	Quick Install	3
1.2	Dependencies	3
1.3	Supported Operating Systems	4
1.4	License	4
2	Install Guide	5
2.1	Install dependencies	5
2.2	Set up an environment	5
2.3	Install Kegbot Server	6
2.4	Create the database	6
2.5	Run Redis	6
2.6	Run the Setup Wizard	7
2.7	Configure E-Mail	7
3	Run Kegbot Server	9
3.1	Run the built-in web server	9
3.2	Configure Kegbot	9
3.3	Allowing external access	10
3.4	Run the task queue workers (Celery)	10
3.5	Next steps	10
4	Production Setup with Gunicorn, Nginx, and Supervisor	11
4.1	Gunicorn	11
4.2	Nginx	11
4.3	supervisord	13
5	Extras and Add-Ons	15
5.1	Plugins	15
5.2	Developer Extras	16
6	Upgrade from a Previous Version	19
6.1	Upgrade notes	19
6.2	Upgrade procedure	19
7	Troubleshooting Kegbot Server	21
7.1	Images and/or javascript not served when <code>DEBUG = False</code>	21

7.2	Date errors while viewing sessions	21
7.3	Statistics are not regenerated	21
7.4	Tweets (or other plugin events) are not sent	22
8	Appendix: Extra Settings	23
9	Changelog	25
9.1	Version 1.2.3 (2015-01-12)	25
9.2	Version 1.2.2 (2015-01-03)	25
9.3	Version 1.2.1 (2014-12-02)	25
9.4	Version 1.2.0 (2014-12-01)	25
9.5	Version 1.1.1 (2014-11-11)	26
9.6	Version 1.1.0 (2014-09-19)	26
9.7	Version 1.0.2 (2014-08-21)	26
9.8	Version 1.0.1 (2014-07-21)	26
9.9	Version 1.0.0 (2014-06-24)	26

Kegbot Server is the brains of a Kegbot system. This guide will help you install, configure, and run a new Kegbot server.

Contents:

Kegbot Server Overview

Kegbot Server is the web service behind Kegbot. It serves as both a frontend for visual navigation of drinking activity, as well as the backend for all Kegbot data.

1.1 Quick Install

New: The easiest way to get going is by following the [Digital Ocean Install Instructions](#).

For those already familiar with Linux, these ultra-quick install instructions may suffice:

```
$ virtualenv ~/kb                # create a new home for kegbot
$ . ~/kb/bin/activate           # step into it
(kb) $ pip install kegbot        # install the latest kegbot release
(kb) $ setup-kegbot.py           # interactive configuration tool
(kb) $ kegbot runserver          # run the dev web server
Development server is running at http://127.0.0.1:8000/
```

Read on for detailed, step-by-step instructions.

1.2 Dependencies

Kegbot Server is built on a number of open source projects. The major dependencies are:

- [Python 2.7](#)
- [MySQL](#) or [PostgreSQL](#)
- [Redis 2.8](#) or newer

In addition, Kegbot Server requires several Python modules which are installed automatically by the Python package manager.

1.3 Supported Operating Systems

Kegbot Server should run on any UNIX-based operating system. Since there are many possible distributions, the examples in this document are written for Debian/Ubuntu and Mac OS X ([Homebrew](#)) package managers.

1.4 License

Kegbot Server is licensed under the [GNU General Public License v2.0](#). You must accept this license before installing or using Kegbot Server.

2.1 Install dependencies

Before we begin, be sure Kegbot's major dependencies are available on your your system.

On Ubuntu or Debian, use `apt-get`:

```
$ sudo apt-get install \  
python-dev \  
python-pip \  
libjpeg-dev \  
libmysqlclient-dev \  
redis-server \  
mysql-client \  
mysql-server \  
redis-server \  
supervisor \  
nginx
```

On Mac OS X, we recommend using [Homebrew](#):

```
$ brew install \  
python \  
mysql \  
libpng \  
jpeg \  
redis \  
nginx
```

2.2 Set up an environment

The first thing you'll need is the Python *virtualenv* package. You probably already have this, but if not, you can install it with:

```
$ sudo pip install virtualenv
```

Once that's done, chose a directory for the Kegbot Server environment and create it with *virtualenv*. In our examples we'll use */data/kb*:

```
$ virtualenv /data/kb
```

Finally, activate your virtualenv:

```
$ source /data/kb/bin/activate
(kb) $
```

Tip: You must activate your virtualenv before updating Kegbot, as well as before running any command-line Kegbot programs.

2.3 Install Kegbot Server

Once you have the environment set up, you're ready to install Kegbot Server:

```
(kb) $ pip install -U kegbot
```

Sit back and relax; this command will download and install the [latest release](#), along with all of its Python dependencies.

2.4 Create the database

Create a new database to store your Kegbot data. On MySQL, the command to use is:

```
$ mysqladmin -u root create kegbot
```

Tip: If your MySQL server requires a password, add *-p* to these commands.

We also recommend creating a new MySQL user solely for Kegbot access. Run the following command, replacing “pw” with a password of your choice:

```
$ mysql -u root -e '
GRANT ALL PRIVILEGES ON kegbot.* TO kegbot@localhost
IDENTIFIED BY "pw"; flush privileges;'
```

2.5 Run Redis

If you installed Redis, it's probably already running. Verify by pinging it:

```
$ redis-cli ping PONG
```

Warning: Kegbot uses Redis databases *0* and *1* by default. If you are using the same Redis server for other programs, we recommend using separate databases.

2.6 Run the Setup Wizard

The program `setup-kegbot.py` will help you:

- Configure Kegbot for the database you set up in the previous step;
- Create Kegbot's Media and Static Files directories;
- Install defaults into your new database.

Run the setup wizard:

```
(kb) $ setup-kegbot.py
```

When finished, you should have a settings file in `~/ .kegbot/local_settings.py` that you can examine.

2.6.1 About Media and Static Files

After running the wizard, two important settings should have been configured: the *media* and *static files* directories.

MEDIA_ROOT This variable controls where Kegbot stores uploaded media: pictures added during account registration or pours, for example.

STATIC_ROOT This variable controls where Kegbot's static media is stored, such as built-in style sheets and images shown on the web page.

Warning: Never place other content in `STATIC_ROOT`, and always be sure it is set to a directory that Kegbot can completely overwrite. For more information, see [Django's documentation for managing static files](#).

2.7 Configure E-Mail

Kegbot can send e-mail in several circumstances. These include:

- Initial account registration.
- Password recovery.
- Configurable notifications.

Before it can send e-mail, Kegbot must be configured with an e-mail backend. To use an SMTP server, add the following lines to your `local_settings.py` file and configure them as appropriate:

```
# Tell Kegbot use the SMTP e-mail backend.
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'

# SMTP server hostname (default: 'localhost') and port (default: 25).
EMAIL_HOST = 'email.example.com'
EMAIL_PORT = 25
```

(continues on next page)

(continued from previous page)

```
# Credentials for SMTP server.  
EMAIL_HOST_USER = 'username'  
EMAIL_HOST_PASSWORD = 'password'  
EMAIL_USE_SSL = False  
EMAIL_USE_TLS = False  
  
# "From" address for e-mails.  
EMAIL_FROM_ADDRESS = 'me@example.com'
```

Run Kegbot Server

By now you have Kegbot installed; it's now time to run the server!

3.1 Run the built-in web server

Kegbot Server includes a simple built-in web server (part of Django). This is not a high performance web server, but works well during testing.

Launch the web server with the following command:

```
(kb) $ kegbot runserver
Validating models...

0 errors found
Django version 1.6.2, using settings 'pykeg.settings'
Development server is running at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

Now try it out! Navigate to <http://localhost:8000/>.

The first time you start Kegbot, you will be prompted to finish installation by running the Setup Wizard. Follow the on-screen prompts to create your admin account and finish installation.

Kegbot Server is 100% functional when running under this development server, but for better performance you'll want to run Kegbot with a "real" HTTP server. When you're ready, see *Production Setup with Gunicorn, Nginx, and Supervisor* for instructions.

3.2 Configure Kegbot

The admin account you just created has access to an additional tab in the navigation bar: "Admin".

Head over to "Taps". Taps define what beer is currently available in the system. By default, two taps are created, but you can add more if you need them. If you're only using one tap, don't worry about the second one.

Before you can pour a drink, you should add a new Keg to the tap.

3.3 Allowing external access

By default, the built-in web server only accepts connections from the local machine. To allow external computers to reach this server, specify the bind address when running it:

```
$ kegbot runserver 0.0.0.0:8000
```

3.4 Run the task queue workers (Celery)

Certain features, such as stats computation, Twitter checkins and web hook support, require a non-webserver process in order to run in the background. Kegbot uses [Celery](#) as its task queue.

Celery is automatically installed with kegbot. You can try running it in the foreground:

```
(kb) $ kegbot run_workers
```

Be sure Celery is always running when your server is running. If it isn't, some features will not work. See [Production Setup with Gunicorn, Nginx, and Supervisor](#) for instructions on automatically launching Celery in the background.

3.5 Next steps

Congratulations! Your Kegbot Server should be basically functional: ready to accept new user registrations and drink reports. In the next chapters, we'll go over tweaking your setup for a long-running production environment.

Production Setup with Gunicorn, Nginx, and Supervisor

Production setup of the Kegbot server consists of the following components:

- [Nginx](#) (public facing web server)
- [Gunicorn](#) (internal HTTP application server)
- [supervisord](#) (process control and monitor)

This chapter will guide you through running these two components.

Note: It is possible to use other components in place of the ones recommended here; you'll need to figure out those steps yourself.

4.1 Gunicorn

Starting with version 0.9.4, Gunicorn is automatically installed with Kegbot Server. You can test it with the built-in command, `run_gunicorn`:

```
(kb) $ kegbot run_gunicorn --debug
```

If this works, you're ready to fire up nginx.

4.2 Nginx

4.2.1 Install Nginx

Nginx is a fast and flexible HTTP server written in C. You can install it with the `nginx` package on Ubuntu:

```
$ sudo apt-get install nginx
```

After nginx is installed and started, a default HTTP server will be running on port 80.

4.2.2 Create Kegbot nginx config

Kegbot includes a sample nginx configuration file in the `deploy/` directory which needs to be tailored to your setup. The contents are also included below:

```
# Kegbot nginx.conf file
#
# Instructions:
#   - Replace "kegbot.example.com" with your external host name
#   - Replace "/data/kegbot/www/media" with the path you configured for
#     MEDIA_ROOT
#   - Replace "/data/kegbot/www/static" with the path you configured for
#     STATIC_ROOT
#   - Replace "/data/kegbot/www" with that parent directory name.
#   - Replace ":8000" with the port number of the Kegbot application server, if
#     changed.

upstream kegbot {
    server 127.0.0.1:8000;
}

server {
    listen 80;
    tcp_nopush on;
    tcp_nodelay on;

    gzip on;
    gzip_disable "msie6";
    gzip_types text/plain text/css application/x-javascript text/xml application/xml_
    ↪application/xml+rss text/javascript;
    gzip_vary on;

    keepalive_timeout 0;
    client_max_body_size 10m;

    location / {
        proxy_redirect      off;
        proxy_set_header    Host                $host:$server_port;
        proxy_set_header    X-Real-IP           $remote_addr;
        proxy_set_header    X-Forwarded-For     $proxy_add_x_forwarded_for;
        proxy_set_header    X-Forwarded-Protocol $scheme;
        proxy_pass            http://kegbot;
    }

    location /media/ {
        alias                /data/kegbot/www/media/;
        access_log           off;
        log_not_found        off;
        expires              7d;
        add_header           pragma public;
        add_header           cache-control "public";
    }
}
```

(continues on next page)

(continued from previous page)

```

location /static/ {
    alias            /data/kegbot/www/static/;
    access_log       off;
    log_not_found    off;
    expires          7d;
    add_header       pragma public;
    add_header       cache-control "public";
}

location /robots.txt {
    root             /data/kegbot/www/static/;
    access_log       off;
    log_not_found    off;
}

location /favicon.ico {
    root             /data/kegbot/www/static/;
    access_log       off;
    log_not_found    off;
}
}

```

Create a copy of this file, editing the paths noted in the comments at the top. Finally, install it:

```

$ sudo cp kegbot-nginx.conf /etc/nginx/sites-available/
$ sudo rm /etc/nginx/sites-enabled/default
$ sudo ln -s /etc/nginx/sites-available/kegbot-nginx.conf /etc/nginx/sites-enabled/
$ sudo service nginx restart

```

4.3 supervisord

Install supervisor using apt-get:

```
$ sudo apt-get install supervisor
```

Supervisor manages programs according to its configuration files. Once again, there is a template in the `deploy/` directory:

```

# Supervisor configuration for Kegbot server.
#
# Instructions:
#   - Replace "/data/kegbot/kb/bin" with the installed path of the kegbot
#     programs.
#   - Replace "user=ubuntu" with the username you wish to run the programs.
#   - Edit paths.
#   - Copy to /etc/supervisor/conf.d/kegbot.conf

[group:kegbot]
programs=gunicorn,workers

[program:gunicorn]

```

(continues on next page)

(continued from previous page)

```
command=/data/kegbot/kb/bin/gunicorn pykeg.web.wsgi:application -w 3
directory=/data/kegbot
user=ubuntu
autostart=true
autorestart=true
redirect_stderr=true

[program:workers]
command=/data/kegbot/kb/bin/kegbot run_workers
directory=/data/kegbot
stopasgroup=true
user=ubuntu
autostart=true
autorestart=true
redirect_stderr=true
```

Make a copy of this file, editing the fields as noted, and install it:

```
$ sudo cp kegbot-supervisor.conf /etc/supervisor/conf.d/kegbot.conf
```

Restart supervisor so that it reads the Kegbot config:

```
$ sudo service supervisor restart
```

Finally, launch the web server and workers:

```
$ sudo supervisorctl start kegbot:gunicorn
$ sudo supervisorctl start kegbot:workers
```

By now you've got a default Kegbot server running. There are several more features you can enable by installing and configuring optional components.

5.1 Plugins

Kegbot Server ships with a few built-in plugins. Most plugins require configuration before they can be used. To configure, see the *Extras* section of the admin tab.

5.1.1 Twitter

When enabled, the Twitter plugin can:

- Tweet when a keg is started or finished.
- Tweet when a new drinking session is started.
- Tweet when a new drink is poured.

Tweets can originate from a system account, per-user accounts, or both.

To configure the plugin, get a [Twitter developer account](#) and set the *Consumer Key* and *Consumer Secret* through the Kegbot admin page for Twitter.

After configuring, click *Link Account* to configure a system account.

5.1.2 Foursquare

The Foursquare plugins allows your users to automatically check-in to your own Foursquare venue the first time they pour in a session.

To configure, get a [Foursquare developer account](#) and input the *Client ID*, *Client Secret*, and *Venue ID* through the Kegbot admin configuration page for Foursquare.

Users will also need to link their own Foursquare accounts (under the *Accounts* tab) before the system will check in on their behalf.

5.1.3 Untappd

The Untappd plugin is similar to the Foursquare plugin: when the system is configured and a drinker with a linked account pours, the plugin will check the user in to that beer type.

The beer type must have an *Untappd ID* saved in your Kegbot beer database; otherwise, Kegbot does not know which Untappd beer to log.

To configure, get [Untappd developer access](#) and configure Kegbot with the credentials. Users will need to link their own Untappd accounts (under the *Account* tab) before the system can check them in.

5.1.4 Web Hooks

You may wish to write your own service to respond to Kegbot events. The Web Hooks plugin is an excellent way to integrate with Kegbot, without being tightly coupled to the server itself.

When activated, the Kegbot server will issue an HTTP `POST` to every configured URL whenever a system event occurs. The body of the post data contains a single JSON message, which includes the following fields:

- `type`: The type of the message. Currently, this is always `'event'`.
- `data`: The payload of the message. This will be the system event object.

The system event JSON structure will be identical to those events seen at the `/api/events/` endpoint.

5.2 Developer Extras

Certain add-ons provide enhanced developer/debugging functionality. You probably won't be interested in these unless you're working on Kegbot Server code.

5.2.1 Remote Error Logs (Sentry)

Your Kegbot server can log internal errors and exceptions to a [Sentry](#) web server. This is mostly useful for monitoring a production site. You can [read the Sentry docs](#) to run your own server, or you can [pay for a hosted Sentry server](#).

Configuring Kegbot to log to your Sentry server is easy:

1. Be sure you have Raven, the sentry client, installed:

```
(kb) pip install raven
```

2. In your `local_settings.py`, add your Sentry URL (shown in your Sentry dashboard):

```
RAVEN_CONFIG = {  
    'dsn': 'http://foo:bar@localhost:9000/2',  
}
```

5.2.2 Debug Toolbar

Developers can get extra information while Kegbot is running by installing [Django Debug Toolbar](#).

To install:

```
### Required: Install the base package.
(kb) $ pip install django-debug-toolbar

### Optional: Additional memcache stats panel.
(kb) $ pip install django-debug-toolbar-memcache
```

When this package is installed and `settings.DEBUG` is `True`, Kegbot will automatically enable it; you don't need to do any additional configuration.

5.2.3 Stats Aggregation (StatsD)

Kegbot can post server-related counters and other statistics to [StatsD](#). This is primarily of interest to developers. To install:

```
(kb) $ pip install django-statsd-mozilla
```

Once installed, you may optionally change the default settings by adding entries to `local_settings.py`:

- `STATSD_CLIENT` (default is `statsd.client`)
- `STATSD_HOST` (default is `localhost`)
- `STATSD_PORT` (default is `8125`)
- `STATSD_PREFIX` (default is empty)

If you have the debug toolbar installed, you may instead route StatsD pings to it by setting `KEGBOT_STATSD_TO_TOOLBAR = True`.

Consult the [django-statsd configuration docs](#) for more details.

Upgrade from a Previous Version

6.1 Upgrade notes

Occasionally we make changes to Kegbot that require special steps or attention when upgrading. Though the section below covers the most commonly-needed upgrade steps, always read the upgrade notes in *the changelog* first.

6.2 Upgrade procedure

Warning: Always make a backup of your data prior to upgrading Kegbot.

1. If running from git, do a `git pull`.
2. Step in to your virtualenv and upgrade to the new version.

If you used `pip` last time:

```
(kb) $ pip install --upgrade kegbot
```

If you used `setup.py` last time:

```
(kb) $ ./setup.py develop
```

3. Run the upgrade script:

```
(kb) $ kegbot upgrade
```

4. Restart the Kegbot web server and Celery.

Troubleshooting Kegbot Server

7.1 Images and/or javascript not served when `DEBUG = False`

The built-in web servers (`kegbot runserver` and `kegbot run_gunicorn`) do not serve static files – any URLs beginning with `/media` and `/static` – when `DEBUG = False`.

This behavior is intentional: serving static files this way is “grossly inefficient” and “probably insecure” according to the [Django developers](#). As a result, Kegbot will only do it when `DEBUG = True`.

To fix, follow the [production setup guide](#), which configures these files to be served directly by the fronting web server Nginx. Alternatively, accept the consequences of running a non-production config and set `DEBUG = True`.

7.2 Date errors while viewing sessions

An error like this indicates a problem with MySQL:

```
ValueError: Database returned an invalid value in QuerySet.dates(). Are time
zone definitions and pytz installed?
```

This issue arises when MySQL does not have complete time zone information, and is discussed in [Django ticket 21629](#) and in the [MySQL documentation](#).

For most systems, the following command will fix the issue:

```
$ mysql_tzinfo_to_sql /usr/share/zoneinfo | mysql -u root -p mysql
```

7.3 Statistics are not regenerated

Statistics are recomputed in a background task. In order for this to work, the task queue (Celery) must be running. You can check queue status through the *Worker Status* section in the admin console.

7.4 Tweets (or other plugin events) are not sent

Plugins that report to external services, such as the Twitter plugin, require the task queue to be running. See *Statistics are not regenerated*.

On Twitter specifically, identical tweets – those with the exact same message content – *may* be suppressed by Twitter. You can work around this by ensuring the message is unique, for example by always including the drink URL.

Appendix: Extra Settings

This section lists settings that may be added to `local_settings.py`. Most of these options serve uncommon needs.

In addition to settings described here, any of [Django's built-in settings](#) may be listed in `local_settings.py`.

ALLOWED_HOSTS

Lists allowed hostnames that the server will respond to. See the [Django documentation for ALLOWED_HOSTS](#) for more information.

Default is `['*']` (any hostname accepted).

```
ALLOWED_HOSTS = ['kegbot.example.com']
```

KEGBOT_BASE_URL

Ordinarily, Kegbot will infer your site's hostname and base URL from incoming requests. This hostname is used to compose full URLs, for example when generating the links in outgoing e-mails and plugin posts.

In some situations you may want to set this value explicitly, for example if your server is available on multiple hostnames.

Default is unset (automatic mode).

```
KEGBOT_BASE_URL = 'http://mykegbot.example:8001/'
```

KEGBOT_ENABLE_ADMIN

When set to `true`, the [Django Admin Site](#) will be enabled, allowing you to browse and edit raw Kegbot data through a web interface.

Warning: Editing raw Kegbot data through the admin interface may leave your system in a corrupt or inconsistent state.

Default is `False`.

```
KEGBOT_ENABLE_ADMIN = True
```

`LOGGING['handlers']['redis']['url']`

When specified, this setting gives the URL of a redis instance to use for temporary log data. The URL is parsed by *redis.from_url*.

Default is `redis://localhost:6379`.

Upgrade Procedure: Please follow *Upgrade from a Previous Version* for general upgrade steps.

9.1 Version 1.2.3 (2015-01-12)

- Allow users to change e-mail addresses.
- Added “bugreport” admin page.
- Fix invitation email footer.

9.2 Version 1.2.2 (2015-01-03)

- New command *kegbot bugreport* collects various system information.
- Bugfix: Crash on end keg button (#326).
- Bugfix: Unicode error during *kegbot upgrade* (#328).

9.3 Version 1.2.1 (2014-12-02)

- Fixed *run_unicorn* launcher.

9.4 Version 1.2.0 (2014-12-01)

- Keg management improvements: The new “Keg Room” view shows kegs by status, and allows kegs to be manually moved between “available” and “finished” states.
- Fancy keg graphics.

- Backup file format has changed. Downgrade to v1.1 to restore from an earlier file format.
- Django 1.7 update.
- Flow sensing and multiuser features can be hidden.
- Statistics now properly consider local timezone (#199).
- Some new keg sizes are supported (#318).
- Keg full volume and beverage type can be edited (#279).

9.5 Version 1.1.1 (2014-11-11)

- API: New endpoint: *drinks/last*.
- Newly-created meters now default to FT330-RJ calibration values.
- Kegadmin: Kegs can be deleted from the “Edit Keg” screen.
- The *kegbot restore* command can run against an unzipped directory.

9.6 Version 1.1.0 (2014-09-19)

- Fullscreen mode.
- New keg artwork.
- New internal beverage fields: IBU, SRM, star rating, and color.

9.7 Version 1.0.2 (2014-08-21)

- Bugfix: Issue #309 (cannot reset password on private sites).
- Redis logging backend is configurable; see [Appendix: Extra Settings](#) (thanks Jared).
- Bugfix: Issue #313 ([link](#)/ matching on usernames).

9.8 Version 1.0.1 (2014-07-21)

- Bugfix: Issue #302 (*api status*/ endpoint).

9.9 Version 1.0.0 (2014-06-24)

- Initial 1.0 release.
- See *upgrade_pre_10* for upgrade instructions.