
django-helpdesk Documentation

Release 0.3.0.dev2

Ross Poulton + django-helpdesk Contributors

Jun 12, 2022

Contents

1	Contents	3
1.1	Installation	3
1.2	Upgrading	6
1.3	Configuration	7
1.4	Settings	9
1.5	Spam Filtering	13
1.6	Custom Fields	14
1.7	API	14
1.8	Integrating django-helpdesk into your application	16
1.9	Working with teams and larger organizations	16
1.10	License	17
2	How Does It Look?	23
3	Quick Start	25
4	Key Features	27
4.1	Customer-facing Capabilities	27
4.2	Staff Capabilities	27
5	Licensing	29

django-helpdesk is a Django application to manage helpdesk tickets for your internal helpdesk. It was formerly known as Jutda Helpdesk.

1.1 Installation

`django-helpdesk` installation isn't difficult, but it requires you have a bit of existing know-how about Django.

1.1.1 Prerequisites

Before getting started, ensure your system meets the following recommended dependencies:

- Python 3.8+
- Django 2.2 LTS or 3.2 LTS (strongly recommend migrating to 3.2 LTS as soon as possible)

Ensure any extra Django modules you wish to use are compatible before continuing.

NOTE: Python 2.7 support was deprecated in `django-helpdesk` as of version 0.2.x and completely removed in version 0.3.0. Users that still need Python 2 support should remain on version 0.2.x.

1.1.2 Getting The Code

Installing using PIP

Try using `pip install django-helpdesk`. Go and have a beer to celebrate Python packaging.

Checkout `stable` from git (Cutting Edge)

If you're planning on editing the code or just want to get whatever is the latest and greatest, you can clone the official Git repository with `git clone git://github.com/django-helpdesk/django-helpdesk.git`. We use the `stable` branch as our development branch for the next major release of `django-helpdesk`.

Copy the `helpdesk` folder into your `PYTHONPATH`.

I just want a .tar.gz!

You can download the latest PyPi package from <http://pypi.python.org/pypi/django-helpdesk/>

Download, extract, and drop helpdesk into your PYTHONPATH

1.1.3 Adding To Your Django Project

If you're on a brand new Django installation, make sure you do a migrate **before** adding helpdesk to your `INSTALLED_APPS`. This will avoid errors with trying to create User settings.

1. Edit your `settings.py` file and add helpdesk to the `INSTALLED_APPS` setting. You also need `django.contrib.admin` in `INSTALLED_APPS` if you haven't already added it. eg:

```
INSTALLED_APPS = (
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites', # Required for determining domain url for use in
    ↪emails
    'django.contrib.admin', # Required for helpdesk admin/maintenance
    'django.contrib.humanize', # Required for elapsed time formatting
    'bootstrap4form', # Required for nicer formatting of forms with the default
    ↪templates
    'account', # Required by pinax-teams
    'pinax.invitations', # Required by pinax-teams
    'pinax.teams', # Team support
    'reversion', # Required by pinax-teams
    'rest_framework', # required for the API
    'helpdesk', # This is us!
)
```

Note: you do not need to use pinax-teams. To disable teams see the *Working with teams and larger organizations* section.

Your `settings.py` file should also define a `SITE_ID` that allows multiple projects to share a single database, and is required by `django.contrib.sites` in Django 1.9+. If you aren't running multiple sites, you can simply add a default `SITE_ID` to `settings.py`:

```
SITE_ID = 1
```

2. Make sure django-helpdesk is accessible via `urls.py`. Add the following line to `urls.py`:

```
path('helpdesk/', include('helpdesk.urls')),
```

Note that you can change 'helpdesk/' to anything you like, such as 'support/' or 'help/'. If you want django-helpdesk to be available at the root of your site (for example at <http://support.mysite.tld/>) then the line will be as follows:

```
path('', include('helpdesk.urls', namespace='helpdesk')),
```

This line will have to come *after* any other lines in your `urls.py` such as those used by the Django admin.

Note that the *helpdesk* namespace is no longer required for Django 1.9+ and you can use a different namespace. However, it is recommended to use the default namespace name for clarity.

3. Create the required database tables.

Migrate using Django migrations:


```
python manage.py migrate helpdesk
```

4. Include your static files in your public web path:

```
python manage.py collectstatic
```

5. Inside your `MEDIA_ROOT` folder, inside the `helpdesk` folder, is a folder called `attachments`. Ensure your web server software can write to this folder - something like this should do the trick:

```
chown www-data:www-data attachments/
chmod 700 attachments
```

(substitute `www-data` for the user / group that your web server runs as, eg ‘`apache`’ or ‘`httpd`’)

If all else fails, you could ensure all users can write to it:

```
chmod 777 attachments/
```

But this is NOT recommended, especially if you’re on a shared server.

6. Ensure that your `attachments` folder has directory listings turned off, to ensure users don’t download files that they are not specifically linked to from their tickets.

If you are using Apache, put a `.htaccess` file in the `attachments` folder with the following content:

```
Options -Indexes
```

You will also have to make sure that `.htaccess` files aren’t being ignored.

Ideally, accessing http://MEDIA_URL/helpdesk/attachments/ will give you a 403 access denied error.

7. If you already have a view handling your logins, then great! If not, add the following to `settings.py` to get your Django installation to use the login view included in `django-helpdesk`:

```
LOGIN_URL = '/helpdesk/login/'
```

Alter the URL to suit your installation path.

8. Load initial e-mail templates, otherwise you will not be able to send e-mail:

```
python manage.py loaddata emailtemplate.json
```

9. If you intend on using local mail directories for processing email into tickets, be sure to create the mail directory before adding it to the queue in the Django administrator interface. The default mail directory is `/var/lib/mail/helpdesk/`. Ensure that the directory has appropriate permissions so that your Django/web server instance may read and write files from this directory.

Note that by default, any mail files placed in your local directory will be permanently deleted after being successfully processed. It is strongly recommended that you take further steps to save emails if you wish to retain backups.

Also, be aware that if a disk error occurs and the local file is not deleted, the mail may be processed multiple times and generate duplicate tickets until the file is removed. It is recommended to monitor log files for `ERRORS` when a file is unable to be deleted.

1.1.4 Upgrading from previous versions

If you are upgrading from a previous version of `django-helpdesk` that used migrations, get an up to date version of the code base (eg by using `git pull` or `pip install --upgrade django-helpdesk`) then migrate the

database:

```
python manage.py migrate helpdesk --db-dry-run # DB untouched
python manage.py migrate helpdesk
```

Lastly, restart your web server software (eg Apache) or FastCGI instance, to ensure the latest changes are in use.

Unfortunately we are unable to assist if you are upgrading from a version of django-helpdesk prior to migrations (ie pre-2011).

You can continue to the ‘Initial Configuration’ area, if needed.

1.1.5 Notes on database backends

NOTE REGARDING SQLITE AND SEARCHING: If you use sqlite as your database, the search function will not work as effectively as it will with other databases due to its inability to do case-insensitive searches. It’s recommended that you use PostgreSQL or MySQL if possible. For more information, see this note in the Django documentation: <http://docs.djangoproject.com/en/dev/ref/databases/#sqlite-string-matching>

When you try to do a keyword search using sqlite, a message will be displayed to alert you to this shortcoming. There is no way around it, sorry.

NOTE REGARDING MySQL: If you use MySQL, with most default configurations you will receive an error when creating the database tables as we populate a number of default templates in languages other than English.

You must create the database the holds the django-helpdesk tables using the UTF-8 collation; see the MySQL manual for more information: <http://dev.mysql.com/doc/refman/5.1/en/charset-database.html>

You may be able to convert an existing MySQL database to use UTF-8 collation by using the following SQL commands:

```
ALTER DATABASE mydatabase CHARACTER SET utf8 COLLATE utf8_general_ci;
ALTER TABLE helpdesk_emailtemplate CONVERT TO CHARACTER SET utf8 COLLATE utf8_general_
→ci;
```

Both utf8_general_ci or utf16_general_ci have been reported to work.

If you do NOT do this step, and you only want to use English-language templates, you may be able to continue however you will receive a warning when running the ‘migrate’ commands.

1.2 Upgrading

Your django-helpdesk installation can be upgraded to the latest version using the release notes below.

1.2.1 Prerequisites

Please consult the Installation instructions for general instructions and tips. The tips below are based on modifications of the original installation instructions.

1.2.2 0.2 -> 0.3

- Under *INSTALLED_APPS*, *bootstrapform* needs to be replaced with *bootstrap4form*
- Unless turning off *pinax_teams*, add the following to *INSTALLED_APPS* for *pinax_teams*: `` "account", "pinax.invitations", "pinax.teams", "reversion", ``

- If using `send_templated_mail`, then it now needs to be imported from `helpdesk.templated_email`

1.3 Configuration

IMPORTANT NOTE: Any tickets created via POP3 or IMAP mailboxes will DELETE the original e-mail from the mail server.

Before django-helpdesk will be much use, you need to do some basic configuration. Most of this is done via the Django admin screens.

1. Visit `http://yoursite/admin/` and add a Helpdesk Queue. If you wish, enter your POP3 or IMAP server details.
2. Visit `http://yoursite/helpdesk/` (or whatever path as defined in your `urls.py`)
3. If you wish to automatically create tickets from the contents of an e-mail inbox, set up a cronjob to run the management command on a regular basis. (Or use Celery, see below)

Don't forget to set the relevant Django environment variables in your crontab:

```
*/5 * * * * /path/to/helpdesk/site/manage.py get_email
```

This will run the e-mail import every 5 minutes

You will need to create a support queue, and associated login/host values, in the Django admin interface, in order for mail to be picked-up from the mail server and placed in the tickets table of your database. The values in the settings file alone, will not create the necessary values to trigger the `get_email` function.

If you wish to use *celery* instead of cron, you must add 'django_celery_beat' to `INSTALLED_APPS` and add a periodic celery task through the Django admin.

You will need to create a support queue, and associated login/host values, in the Django admin interface, in order for mail to be picked-up from the mail server and placed in the tickets table of your database. The values in the settings file alone, will not create the necessary values to trigger the `get_email` function.

4. If you wish to automatically escalate tickets based on their age, set up a cronjob to run the escalation command on a regular basis:

```
0 * * * * /path/to/helpdesk/site/manage.py escalate_tickets
```

This will run the escalation process hourly, using the 'Escalation Days' setting for each queue to determine which tickets to escalate.

5. If you wish to exclude some days (eg, weekends) from escalation calculations, enter the dates manually via the Admin, or setup a cronjob to run a management command on a regular basis:

```
0 0 * * 0 /path/to/helpdesk/site/manage.py create_escalation_exclusions --days_
↪saturday,sunday --escalate-verbosely
```

This will, on a weekly basis, create exclusions for the coming weekend.

6. Log in to your Django admin screen, and go to the 'Sites' module. If the site `example.com` is listed, click it and update the details so they are relevant for your website.
7. If you do not send mail directly from your web server (eg, you need to use an SMTP server) then edit your `settings.py` file so it contains your mail server details:

```
EMAIL_HOST = 'XXXXX'
EMAIL_HOST_USER = 'YYYYYY@ZZZZ.PPP'
EMAIL_HOST_PASSWORD = '123456'
```

8. If you wish to use SOCKS4/5 proxy with Helpdesk Queue email operations, install PySocks manually. Please note that mixing both SOCKS and non-SOCKS email sources for different queues is only supported under Python 2; on Python 3, SOCKS proxy support is all-or-nothing: either all queue email sources must use SOCKS or none may use it. If you need this functionality on Python 3 please [let us know](#).

You're now up and running! Happy ticketing.

1.3.1 Queue settings via admin interface

Locale

The *Locale* value must match the value in the `locale` column in the `helpdesk_emailtemplate` table if you wish to use those templates. For default installations/templates those values are `cs`, `de`, `en`, `es`, `fi`, `fr`, `it`, `pl`, `ru` and `zh`.

If you want to use a different *Local* then you will need to generate/edit the necessary templates (and set the value in the `locale` column) for those locales. This includes when using language variants, such as `de-CH`, `en-GB` or `fr-CA` for example.

E-Mail Check Interval

This setting does not trigger e-mail collection, it merely throttles it. In order to trigger e-mail collection you must run a crontab to trigger `manage.py get_email`. The setting in *E-Mail Check Interval* prevents your crontab from running the e-mail trigger more often than the interval set.

For example, setting *E-Mail Check Interval* to 5 will limit the collection of e-mail to once every five minutes, even if your crontab is firing every five seconds. If your cron job is set to fire once every hour, then e-mail will only be collected once every hour.

The cron job triggers the collection of e-mail, *E-Mail Check Interval* restricts how often the trigger is effective.

To remove this limit, set *E-Mail Check Interval* to 0.

Potential problems

There is potential for a timing clash to prevent triggering of mail collection if *E-Mail Check Interval* and your crontab interval are identical. Because the crontab runs fractionally before, or at exactly the same time as *E-Mail Check Interval* is run, if the timings for both are identical then every second call by the crontab will be ignored by *E-Mail Check Interval* because its interval has yet to expire.

The result is that if both crontab and *E-Mail Check Interval* are set to run at five minute intervals, then mail may actually only be collected every ten minutes. You will see the evidence of this in the helpdesk mail log, or in the logs of your mail server.

To avoid this problem set the crontab and *E-Mail Check Interval* to marginally different values (or set *E-Mail Check Interval* to 0). *E-Mail Check Interval* will only take an integer value, in minutes, so if you want a five minute interval between mail checks, then you will either have to set *E-Mail Check Interval* to 4 and the crontab interval to 300 seconds, or the *E-Mail Check Interval* to 5 and the crontab interval to 305 seconds.

The crontab interval overrides the *E-Mail Check Interval*, and resets the *E-Mail Check Interval* each time it fires, as long as the crontab interval is greater than *E-Mail Check Interval*.

1.3.2 Custom Navigation Header

You may add your own site specific navigation header to be included inside the `<body>` tag and before django-helpdesk navbar.

1. Create an override template in your project's templates directory:

```
helpdesk/custom_navigation_header.html
```

2. Update the contents to display your custom navigation.

1.4 Settings

First, django-helpdesk needs `django.core.context_processors.request` activated, so you must add it to the `settings.py`. Add the following:

```
TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        ...
        'OPTIONS': {
            ...
            'context_processors': (
                # Default ones first
                ...
                # The one django-helpdesk requires:
                "django.template.context_processors.request",
            ),
        },
    ],
]
```

The following settings can be changed in your `settings.py` file to help change the way django-helpdesk operates. There are quite a few settings available to toggle functionality within django-helpdesk.

1.4.1 HELPDESK_DEFAULT_SETTINGS

django-helpdesk has a built in `UserSettings` entity with per-user options that they will want to configure themselves. When you create a new user, a set of options is automatically created for them which they can then change themselves.

If you want to override the default settings for your users, create `HELPDESK_DEFAULT_SETTINGS` as a dictionary in `settings.py`. The default is below:

```
HELPDESK_DEFAULT_SETTINGS = {
    'use_email_as_submitter': True,
    'email_on_ticket_assign': True,
    'email_on_ticket_change': True,
    'login_view_ticketlist': True,
    'tickets_per_page': 25
}
```

1.4.2 Generic Options

These changes are visible throughout django-helpdesk

- **HELPDESK_REDIRECT_TO_LOGIN_BY_DEFAULT** When a user visits “/”, should we redirect to the login page instead of the default homepage?
Default: `HELPDESK_REDIRECT_TO_LOGIN_BY_DEFAULT = False`
- **HELPDESK_KB_ENABLED** show knowledgebase links?
Default: `HELPDESK_KB_ENABLED = True`
- **HELPDESK_NAVIGATION_ENABLED** Show extended navigation by default, to all users, irrespective of staff status?
Default: `HELPDESK_NAVIGATION_ENABLED = False`
- **HELPDESK_TRANSLATE_TICKET_COMMENTS** Show dropdown list of languages that ticket comments can be translated into via Google Translate?
Default: `HELPDESK_TRANSLATE_TICKET_COMMENTS = False`
- **HELPDESK_TRANSLATE_TICKET_COMMENTS_LANG** List of languages to offer. If set to false, all default google translate languages will be shown.
Default: `HELPDESK_TRANSLATE_TICKET_COMMENTS_LANG = ["en", "de", "fr", "it", "ru"]`
- **HELPDESK_SHOW_CHANGE_PASSWORD** Show link to ‘change password’ on ‘User Settings’ page?
Default: `HELPDESK_SHOW_CHANGE_PASSWORD = False`
- **HELPDESK_FOLLOWUP_MOD** Allow user to override default layout for ‘followups’ (work in progress)
Default: `HELPDESK_FOLLOWUP_MOD = False`
- **HELPDESK_AUTO_SUBSCRIBE_ON_TICKET_RESPONSE** Auto-subscribe user to ticket as a ‘CC’ if (s)he responds to a ticket?
Default: `HELPDESK_AUTO_SUBSCRIBE_ON_TICKET_RESPONSE = False`
- **HELPDESK_EMAIL_SUBJECT_TEMPLATE** Subject template for templated emails. `%(subject)s` represents the subject wording from the email template (e.g. “(Closed)”). *Warning:* your subject template should always include a `{{ ticket.ticket }}` somewhere as many django-helpdesk features rely on the ticket ID in the subject line in order to correctly route mail to the corresponding ticket. If you leave out the ticket ID, your helpdesk may not work correctly!
Default: `HELPDESK_EMAIL_SUBJECT_TEMPLATE = "{{ ticket.ticket }}" {{ ticket.title|safe }} %(subject)s`
- **HELPDESK_EMAIL_FALLBACK_LOCALE** Fallback locale for templated emails when queue locale not found
Default: `HELPDESK_EMAIL_FALLBACK_LOCALE = "en"`
- **HELPDESK_MAX_EMAIL_ATTACHMENT_SIZE** Maximum size, in bytes, of file attachments that will be sent via email
Default: `HELPDESK_MAX_EMAIL_ATTACHMENT_SIZE = 512000`
- **QUEUE_EMAIL_BOX_UPDATE_ONLY** Only process mail with a valid tracking ID; all other mail will be ignored instead of creating a new ticket.
Default: `QUEUE_EMAIL_BOX_UPDATE_ONLY = False`

- **HELPDESK_ANON_ACCESS_RAISES_404** If True, redirects user to a 404 page when attempting to reach ticket pages while not logged in, rather than redirecting to a login screen.

Default: `HELPDESK_ANON_ACCESS_RAISES_404 = False`

- **HELPDESK_ENABLE_DEPENDENCIES_ON_TICKET** If False, disable the dependencies fields on ticket.

Default: `HELPDESK_ENABLE_DEPENDENCIES_ON_TICKET = True`

- **HELPDESK_ENABLE_TIME_SPENT_ON_TICKET** If False, disable the time spent fields on ticket.

Default: `HELPDESK_ENABLE_TIME_SPENT_ON_TICKET = True`

- **HELPDESK_TICKETS_TIMELINE_ENABLED** If False, remove from the dashboard the Timeline view for tickets.

Default: `HELPDESK_TICKETS_TIMELINE_ENABLED = True`

1.4.3 Options shown on public pages

These options only change display of items on public-facing pages, not staff pages.

- **HELPDESK_VIEW_A_TICKET_PUBLIC** Show 'View a Ticket' section on public page?

Default: `HELPDESK_VIEW_A_TICKET_PUBLIC = True`

- **HELPDESK_SUBMIT_A_TICKET_PUBLIC** Show 'submit a ticket' section & form on public page?

Default: `HELPDESK_SUBMIT_A_TICKET_PUBLIC = True`

- **HELPDESK_PUBLIC_TICKET_FORM_CLASS** Define custom form class to show on public pages for anon users. You can use it for adding custom fields and validation, captcha and so on.

Default: `HELPDESK_PUBLIC_TICKET_FORM_CLASS = "helpdesk.forms.PublicTicketForm"`

1.4.4 Options for public ticket submission form

- **HELPDESK_PUBLIC_TICKET_QUEUE** Sets the queue for tickets submitted through the public form. If defined, the matching form field will be hidden. This cannot be *None* but must be set to a valid queue slug.

Default: Not defined

- **HELPDESK_PUBLIC_TICKET_PRIORITY** Sets the priority for tickets submitted through the public form. If defined, the matching form field will be hidden. Must be set to a valid integer priority.

Default: Not defined

- **HELPDESK_PUBLIC_TICKET_DUE_DATE** Sets the due date for tickets submitted through the public form. If defined, the matching form field will be hidden. Set to *None* if you want to hide the form field but do not want to define a value.

Default: Not defined

1.4.5 Options that change ticket updates

- **HELPDESK_ALLOW_NON_STAFF_TICKET_UPDATE** Allow non-staff users to interact with tickets? Set to True to allow any authenticated user to manage tickets. You can also apply a custom authorisation logic for identifying helpdesk staff members, by setting this to a callable. In that case, the value should be a function accepting the active user as a parameter and returning True if the user is considered helpdesk staff, e.g.

`lambda u: u.is_authenticated() and u.is_active and u.groups.filter(name='helpdesk_staff').exists()))`

Default: `HELPDESK_ALLOW_NON_STAFF_TICKET_UPDATE = False`

- **HELPDESK_SHOW_EDIT_BUTTON_FOLLOW_UP** Show edit buttons in ticket follow ups?

Default: `HELPDESK_SHOW_EDIT_BUTTON_FOLLOW_UP = True`

- **HELPDESK_SHOW_DELETE_BUTTON_SUPERUSER_FOLLOW_UP** Show delete buttons in ticket follow ups if user is 'superuser'?

Default: `HELPDESK_SHOW_DELETE_BUTTON_SUPERUSER_FOLLOW_UP = False`

- **HELPDESK_UPDATE_PUBLIC_DEFAULT** Make all updates public by default? This will hide the 'is this update public' checkbox.

Default: `HELPDESK_UPDATE_PUBLIC_DEFAULT = False`

- **HELPDESK_STAFF_ONLY_TICKET_OWNERS** Only show staff users in ticket owner drop-downs?

Default: `HELPDESK_STAFF_ONLY_TICKET_OWNERS = False`

- **HELPDESK_STAFF_ONLY_TICKET_CC** Only show staff users in ticket cc drop-down?

Default: `HELPDESK_STAFF_ONLY_TICKET_CC = False`

1.4.6 Staff Ticket Creation Settings

- **HELPDESK_CREATE_TICKET_HIDE_ASSIGNED_TO** Hide the 'assigned to' / 'Case owner' field from the 'create_ticket' view? It'll still show on the ticket detail/edit form.

Default: `HELPDESK_CREATE_TICKET_HIDE_ASSIGNED_TO = False`

- **HELPDESK_ACTIVATE_API_ENDPOINT** Activate the API endpoint to manage tickets thanks to Django REST Framework. See the API section in documentation for more information.

Default: `HELPDESK_ACTIVATE_API_ENDPOINT = False`

1.4.7 Staff Ticket View Settings

- **HELPDESK_ENABLE_PER_QUEUE_STAFF_PERMISSION** If `True`, logged in staff users only see queues and tickets to which they have specifically been granted access - this holds for the dashboard, ticket query, and ticket report views. User assignment is done through the standard `django.admin.admin` permissions. *Note:* Staff with access to admin interface will be able to see the full list of tickets, but won't have access to details and could not modify them. This setting does not prevent staff users from creating tickets for all queues. Also, superuser accounts have full access to all queues, regardless of whatever queue memberships they have been granted.

Default: `HELPDESK_ENABLE_PER_QUEUE_STAFF_PERMISSION = False`

1.4.8 Default E-Mail Settings

The following settings default to `None` but can be set as defaults, rather than setting them per-queue.

- `QUEUE_EMAIL_BOX_TYPE`
- `QUEUE_EMAIL_BOX_SSL`
- `QUEUE_EMAIL_BOX_HOST` ``
- `QUEUE_EMAIL_BOX_USER`

- `QUEUE_EMAIL_BOX_PASSWORD`

1.4.9 Discontinued Settings

The following settings were defined in previous versions and are no longer supported.

- **HELPDESK_CUSTOM_WELCOME**
- **HELPDESK_KB_ENABLED_STAFF** Now always True
- **HELPDESK_NAVIGATION_STATS_ENABLED** Now always True
- **HELPDESK_PREPEND_ORG_NAME** Please customise your local *helpdesk/base.html* template if needed
- **HELPDESK_SHOW_DELETE_BUTTON_TICKET_TOP** Button is always shown
- **HELPDESK_SHOW_EDIT_BUTTON_TICKET_TOP** Button is always shown
- **HELPDESK_SHOW_HOLD_BUTTON_TICKET_TOP** Button is always shown
- **HELPDESK_SHOW_KB_ON_HOMEPAGE** KB categories are always shown on the homepage
- **HELPDESK_SUPPORT_PERSON** Please customise your local *helpdesk/attribution.html* template if needed
- **HELPDESK_DASHBOARD_SHOW_DELETE_UNASSIGNED** Button is always shown
- **HELPDESK_DASHBOARD_HIDE_EMPTY_QUEUES** Empty queues are always hidden
- **HELPDESK_DASHBOARD_BASIC_TICKET_STATS** Stats are always shown
- **HELPDESK_FOOTER_SHOW_API_LINK** Link to API documentation is always shown. Edit your local *helpdesk/base.html* template if needed.
- **HELPDESK_FOOTER_SHOW_CHANGE_LANGUAGE_LINK** Is never shown. Use your own template if required.
- **HELPDESK_ENABLE_PER_QUEUE_MEMBERSHIP** Discontinued in favor of **HELPDESK_ENABLE_PER_QUEUE_STAFF_PERMISSION**.
- **HELPDESK_FULL_FIRST_MESSAGE_FROM_EMAIL** Do not ignore forwarded and replied text from the email messages which create a new ticket; useful for cases when customer forwards some email (error from service or something) and wants support to see that
- **HELPDESK_ALWAYS_SAVE_INCOMING_EMAIL_MESSAGE** Any incoming .eml message is saved and available, helps when customer spent some time doing fancy markup which has been corrupted during the email-to-ticket-comment translate process

1.5 Spam Filtering

django-helpdesk includes a copy of `akismet.py` by [Michael Foord](#), which lets incoming ticket submissions be automatically checked against either the [Akismet](#) or [TypePad Anti-Spam](#) services.

To enable this functionality, sign up for an API key with one of these two services.

1.5.1 Akismet

- Sign up at <http://akismet.com/>
- Save your API key in `settings.py` as `AKISMET_API_KEY`

Note: Akismet is only free for personal use. Paid commercial accounts are available.

1.5.2 TypePad AntiSpam

- Sign up at <http://antispam.typepad.com/>
- Save your API key in `settings.py` as `TYPEPAD_ANTISPAM_API_KEY`

This service is free to use, within their terms and conditions.

If you have either of these settings enabled, the spam filtering will be done automatically. If you have *both* settings configured, TypePad will be used instead of Akismet.

1.5.3 Example

A sample configuration in `settings.py` may be:

```
TYPEPAD_ANTISPAM_API_KEY = 'abc123'
```

1.6 Custom Fields

django-helpdesk supports custom fields on the `Ticket` model. These fields are created by using the Django administration tool, and are shown on both the public and staff submission forms. You can use most Django field types including text, integer, boolean, and list.

The demo at <http://django-helpdesk-demo.herokuapp.com> contains an example of each type of custom field, including a mix of mandatory and optional fields.

Custom fields are relatively inefficient; you can search them, but this might degrade performance of your installation if you make use of custom fields. They can be useful for tracking extra information that your organisation needs but that isn't supported out of the box.

1.7 API

A REST API (built with `django-rest-framework`) is available in order to list, create, update and delete tickets from other tools thanks to HTTP requests.

If you wish to use it, you have to add this line in your settings:

```
HELPDESK_ACTIVATE_API_ENDPOINT = True
```

You must be authenticated to access the API, the URL endpoint is `/api/tickets/`. You can configure how you wish to authenticate to the API by customizing the `DEFAULT_AUTHENTICATION_CLASSES` key in the `REST_FRAMEWORK` setting (more information on this page : <https://www.django-rest-framework.org/api-guide/authentication/>)

1.7.1 GET

Accessing the endpoint `/api/tickets/` with a **GET** request will return you the complete list of tickets.

Accessing the endpoint `/api/tickets/<ticket-id>` with a **GET** request will return you the data of the ticket you provided the ID.

1.7.2 POST

Accessing the endpoint `/api/tickets/` with a **POST** request will let you create a new tickets.

You need to provide a JSON body with the following data :

- **queue**: ID of the queue
- **title**: the title (subject) of the ticket
- **description**: the description of the ticket
- **resolution**: an optional text for the resoluion of the ticket
- **submitter_email**: the email of the ticket submitter
- **assigned_to**: ID of the ticket's assigned user
- **status**: integer corresponding to the status (OPEN=1, REOPENED=2, RESOLVED=3, CLOSED=4, DUPLICATE=5). It is OPEN by default.
- **on_hold**: boolean to indicates if the ticket is on hold
- **priority**: integer corresponding to different degrees of priority 1 to 5 (1 is Critical and 5 is Very Low)
- **due_date**: date representation for when the ticket is due
- **merged_to**: ID of the ticket to which it is merged

Note that `status` will automatically be set to OPEN. Also, some fields are not configurable during creation: `resolution`, `on_hold` and `merged_to`.

Moreover, if you created custom fields, you can add them into the body with the key `custom_<custom-field-slug>`.

Here is an example of a cURL request to create a ticket (using Basic authentication)

```
curl --location --request POST 'http://127.0.0.1:8000/api/tickets/' \
--header 'Authorization: Basic YWRtaW46YWRtaW4=' \
--header 'Content-Type: application/json' \
--data-raw '{"queue": 1, "title": "Test Ticket API", "description": "Test create_
ticket from API", "submitter_email": "test@mail.com", "priority": 4}'
```

Accessing the endpoint `/api/users/` with a **POST** request will let you create a new user.

You need to provide a JSON body with the following data :

- **first_name**: first name
- **last_name**: last name
- **username**: username
- **email**: user email
- **password**: user password

1.7.3 PUT

Accessing the endpoint `/api/tickets/<ticket-id>` with a **PUT** request will let you update the data of the ticket you provided the ID.

You must include all fields in the JSON body.

1.7.4 PATCH

Accessing the endpoint `/api/tickets/<ticket-id>` with a **PATCH** request will let you do a partial update of the data of the ticket you provided the ID.

You can include only the fields you need to update in the JSON body.

1.7.5 DELETE

Accessing the endpoint `/api/tickets/<ticket-id>` with a **DELETE** request will let you delete the ticket you provided the ID.

1.8 Integrating django-helpdesk into your application

Django-helpdesk associates an email address with each submitted ticket. If you integrate django-helpdesk directly into your django application, logged in users will automatically have their email address set when they visit the `/tickets/submit/` form. If you wish to pre-fill fields in this form, you can do so simply by setting the following query parameters:

- *queue*
- *title*
- *body*
- *submitter_email*
- *custom_<custom-field-slug>*

There is also a page under the url `/tickets/submit_iframe/` with the same behavior.

Fields may be hidden by adding them to a comma separated `_hide_fields_` query parameter.

Here is an example url to get you started: `http://localhost:8000/desk/tickets/submit_iframe/?queue=1&custom_dpnk-user=http://lol.cz;submitter_email=foo@bar.cz&title=lol&_hide_fields_=title,queue,submitter_email`. This url sets the queue to 1, sets the custom field `dpnk-url` to `http://lol.cz` and `submitter_email` to `lol@baz.cz` and hides the title, queue, and submitter_email fields. Note that hidden fields should be set to a default.

Note that these fields will continue to be user-editable despite being pre-filled.

1.9 Working with teams and larger organizations

If you only have one or two people working on tickets, basic Queue setup is enough to get you going. You can now assign tickets to teams for better ticket filtering, reducing noise and improving organization efficiency.

Rather than assigning tickets to teams directly, django-helpdesk allows you assign tickets to knowledge-base items and then assign knowledge base items to teams.

Knowledge-base items can be in either public or private knowledge-base categories, so this organizational structure need not have any influence on the external appearance of your public helpdesk web portal.

You can visit the ‘Pinax Teams’ page in your django admin in order to create a team and add team members.

You can assign a knowledge-base item to a team on the Helpdesk admin page.

Once you have set up teams. Unassigned tickets which are associated with a knowledge-base item will only be shown on the dashboard to those users who are members of the team which is associated with that knowledge-base item.

Note: It is possible that pinax-teams will interfere with other packages that you already use in your project. If you do not wish to use team functionality, you can dissable teams by setting the following settings: `HELPDESK_TEAMS_MODEL` to any random model, `HELPDESK_TEAMS_MIGRATION_DEPENDENCIES` to `[]`, and `HELPDESK_KBITEM_TEAM_GETTER` to `lambda _: None`. You can also use a different library in place of pinax teams by setting those settings appropriately. `HELPDESK_KBITEM_TEAM_GETTER` should take a `kbitem` and return a team object with a `name` property and a method `is_member(self, user)` which returns `true` if user is a member of the team.

1.10 License

django-helpdesk is released under the terms of the BSD license. You must agree to these terms before installing or using django-helpdesk.:

```
Copyright (c) 2008, Ross Poulton (Trading as Jutda)
All rights reserved.

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

    1. Redistributions of source code must retain the above copyright
       notice, this list of conditions and the following disclaimer.

    2. Redistributions in binary form must reproduce the above copyright
       notice, this list of conditions and the following disclaimer in the
       documentation and/or other materials provided with the distribution.

    3. Neither the name of Ross Poulton, Jutda, nor the names of any
       of its contributors may be used to endorse or promote products
       derived from this software without specific prior written permission.

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

EXCEPTIONS

This software is distributed with some third-party software which is not distributed
under the above license. See LICENSE.3RDPARTY for further details.
```

django-helpdesk includes 3rd party software. The licenses for these applications are included below.

1.10.1 License for jQuery & jQuery UI

:: Copyright (c) 2007 John Resig, <http://jquery.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.

1.10.2 License for jQuery UI ‘Smoothness’ theme

::

/*

- jQuery UI screen structure and presentation
- This CSS file was generated by ThemeRoller, a Filament Group Project for jQuery UI
- Author: Scott Jehl, scott@filamentgroup.com, <http://www.filamentgroup.com>
- Visit ThemeRoller.com

*/

1.10.3 License for akismet.py

:: Copyright (c) 2003-2009, Michael Foord All rights reserved. E-mail : fuzzyman AT voidspace DOT org DOT uk

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

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- Neither the name of Michael Foord nor the name of Voidspace may be used to endorse or promote products derived from this software without specific prior written permission.

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

1.10.4 License for jqPlot

:: Copyright (c) 2009 - 2010 Chris Leonello

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.

1.10.5 License for StartBootstrap SB Admin v2.0 theme

The MIT License (MIT)

Copyright (c) 2013-2016 Blackrock Digital LLC.

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.

1.10.6 License for Raphael

The MIT License

Copyright (c) 2008-2010 Dmitry Baranovskiy

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**

(continues on next page)

(continued from previous page)

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.

1.10.7 License for Morris.js

Copyright (c) 2013, Olly Smith
All rights reserved.

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

1. Redistributions of source code must retain the above copyright notice, this list of conditions **and** the following disclaimer.
2. Redistributions **in** binary form must reproduce the above copyright notice, this list of conditions **and** the following disclaimer **in** the documentation **and/or** other materials provided **with** the distribution.

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

1.10.8 License for DataTables

Copyright (C) 2008-2016, SpryMedia Ltd.

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.

(continues on next page)

(continued from previous page)

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.

1.10.9 License for Flot

Copyright (c) 2007-2014 IOLA and Ole Laursen

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.

1.10.10 License for Metis Menu

Copyright (C) 2016, Osman Nuri Okumuş

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.

1.10.11 License for Bootstrap CSS

Copyright (c) 2011–2016 Twitter, Inc.

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.

1.10.12 License for Font Awesome

Font License

Applies to **all** desktop **and** webfont files **in** the following directory:
font-awesome/fonts/.
License: SIL OFL 1.1
URL: <http://scripts.sil.org/OFL>

Code License

Applies to **all** CSS **and** LESS files **in** the following directories:
font-awesome/css/, font-awesome/less/, **and** font-awesome/scss/.
License: MIT License
URL: <http://opensource.org/licenses/mit-license.html>

Documentation License

Applies to **all** Font Awesome project files that are **not** a part of the Font **or** Code licenses.
License: CC BY 3.0
URL: <http://creativecommons.org/licenses/by/3.0/>

Brand Icons

All brand icons are trademarks of their respective owners.
The use of these trademarks does **not** indicate endorsement of the trademark holder by Font Awesome, nor vice versa.
Brand icons should only be used to represent the company **or** product to which they refer.

CHAPTER 2

How Does It Look?

You can see a demo installation at <http://django-helpdesk-demo.herokuapp.com/>

CHAPTER 3

Quick Start

django-helpdesk is just a Django application with models, views, templates, and some media. If you're comfortable with Django just try `pip install django-helpdesk`. If not, continue to read the Installation document.

CHAPTER 4

Key Features

django-helpdesk has been designed for small businesses who need to receive, manage and respond to requests for help from customers. In this context ‘customers’ may be external users, or other people within your company.

- Tickets can be opened via email
- Multiple queues / categories of tickets
- Integrated FAQ / knowledgebase
- API to manage tickets

4.1 Customer-facing Capabilities

Customers (who are not ‘staff’ users in Django) can:

1. Browse your knowledgebase / FAQ
2. Submit support requests via web/email
3. Review open and closed requests they submitted

4.2 Staff Capabilities

If a user is a staff member, they get general helpdesk access, including:

1. See the ticket dashboard showing unassigned tickets and basic status of the helpdesk
2. Review the tickets assigned to them
3. Search through all tickets, open and closed
4. Save their searches for future use
5. Follow up or respond to tickets

6. Assign tickets to themselves or other staff members
7. Resolve tickets
8. Merge multiple tickets into one
9. Create, Read, Update and Delete tickets through a REST API

Optionally, their access to view tickets, both on the dashboard and through searches and reports, may be restricted by a list of queues to which they have been granted membership. Create and update permissions for individual tickets are not limited by this optional restriction.

CHAPTER 5

Licensing

django-helpdesk is released under the BSD license, however it packages 3rd party applications which may be using a different license. More details can be found in the [License](#) documentation.