

Email helpdesk built using Ruby on Rails and Zurb Foundation <http://getbrimir.com>

774 commits

11 branches

9 releases

25 contributors

<> Code

Issues 7


Pull requests 2


Wiki



Pulse

Graphs


HTTPS clone URL

<https://github.com/ivaldi/brimir> 

You can clone with [HTTPS](#), [SSH](#),
or [Subversion](#). 

 Clone in Desktop Download ZIP branch: master **brimir** / + 

Add notice about MySQL collation and charset

 frenkel authored 3 hours agolatest commit bd4b81cf42 

app	Prefer HTML part because most email clients send that anyway.	a month ago
bin	Updates by rake rails:update	4 months ago
config	improving brazilian portuguese locale	25 days ago
db	Make sure MySQL has same limits as PostgreSQL. Fixes #181.	a month ago
lib	Initial Rails setup.	3 years ago
log	Initial Rails setup.	3 years ago
public	Further improve popup and styling. Include a gem to precompile non-di...	9 months ago
script	Fix output of curl command	a month ago
test	Whitespace removed	3 months ago
vendor	Initial Rails setup.	3 years ago
.gitignore	explicitly add tzinfo-data to Gemfile	4 months ago
.travis.yml	Test sqlite3 as well	3 months ago
Gemfile	Use single quotes	2 months ago
Gemfile.lock	Add devise_idap_authenticatable gem	2 months ago
LICENSE	Add AGPL license info.	3 years ago
README.md	Add notice about MySQL collation and charset	3 hours ago
Rakefile	Rename the project to a better working title.	3 years ago
config.ru	Rename the project to a better working title.	3 years ago

 README.md

Brimir

build passing

coverage 94%

[Brimir](#) is a simple helpdesk system that can be used to handle support requests via incoming email. Brimir is currently used in production at [Ivaldi](#).

Installation

Brimir is a rather simple Ruby on Rails application. The only difficulty in setting things up is how to get incoming email to work. See the next section for details.

Any Rails application needs a web server with Ruby support first. We use Phusion Passenger (`mod_rails`) ourselves, but you can also use Thin, Puma or Unicorn. Phusion Passenger can be installed for Nginx or Apache, you can chose whichever you like best. The installation differs depending on your distribution, so have a look at their [Nginx installation manual](#) or their [Apache installation manual](#).

After setting up a webserver, you have to create a database for Brimir and modify the config file in `config/database.yml` to reflect the details. Set your details under the production section. We advise to

use `adapter: postgresql` OR `adapter: mysql2` for production usage, because those are the only two adapters and database servers we test. *If you plan to use MySQL, make sure you use `utf8` as your charset and collation.*

Next up: configuring your outgoing email address and url. This can be set in `config/environments/production.rb` by adding the following lines *before* the keyword `end`:

```
config.action_mailer.default_options = { from: 'brimir@yoururl.com' }

config.action_mailer.default_url_options = { host: 'brimir.yoururl.com' }
```

Now install the required gems by running the following command if you want **PostgreSQL support**:

```
bundle install --without sqlite mysql development test --deployment
```

Run the following command to install gems if you want **MySQL support**:

```
bundle install --without sqlite postgresql development test --deployment
```

Generate a `secret_key_base` in the `secrets.yml` file:

```
sed -i "s/<%= ENV\[\"SECRET_KEY_BASE\"%\] %>/'bin/rake secret`/g" config/secrets.yml
```

Next, load the database schema and precompile assets:

```
rake db:schema:load RAILS_ENV=production
rake assets:precompile RAILS_ENV=production
```

If you want to use LDAP, configure `config/ldap.yml` accordingly, then change the auth strategy in your application config in file `config/application.rb`:

```
config.devise_authentication_strategy = :ldap_authenticatable
```

(Optional for LDAP) Last thing left to do before logging in is making a user and adding some statuses. You can do this by running:

```
bin/rails console production
u = User.new({ email: 'your@email.address', password: 'somepassword', password_confirmation: 's
```

Customization

Some applicant level configuration can be set through `config/settings.yml`

Brimir is available in several languages. By default, it will use the locale corresponding to the user browser agent, if it was among the supported locales. If you want to change this and force certain locale, you can do that by setting: `ignore_user_agent_locale: true` in `config/settings.yml`

Incoming email

Incoming emails can be posted to the tickets url by using the script found in `scripts/post-mail`. Create an alias in your `/etc/aliases` file like this:

```
brimir: "|/bin/bash /path/to/your/brimir/repo/script/post-mail http://yoururl.com/tickets.json"
```

Now sending an email to brimir@yoururl.com should start curl and post the email to your brimir installation.

Text

ChangeFileRegex

RootFile

RootFile

RootFile

RootFile

ChangeFileRegex

ChangeFileRegex
missing information
on file

ChangeFileRegex

ChangeFileLines
missing information on lines

Contributing

We appreciate all contributions! If you would like to contribute, please follow these steps:

- Fork the repo.
- Create a branch with a name that describes the change.
- Make your changes in the branch.
- Submit a pull-request to merge your feature-branch in our master branch.

Requested features

Some users have made requests for the following features. If you would like to contribute, you could add any of these.

- Allowing customers to update ticket status, with correct email notifications.
- Switchable property to support threads by using special tags in the subject line instead of relying on mail headers.
- Support for hosted incoming mail services (Sendgrid, Mandrill), possibly using griddler gem.
- Ability to sign in using a Single Sign On functionality based on Shared Token or JWT.
- Queue sorting by column header.
- Private note addition to tickets.
- Automated replies based on the current rule system.
- Closing issues from the overview page.
- Remove user functionality, without losing ticket and reply information.
- Adding knowledge base functionality.
- Welcome mail for new users (after mailing a ticket for example) with their password.
- Set priority, assignee and labels on the create ticket form.
- Assign tickets to groups of users
- When replying, select a response from pre-defined canned responses and modify to your needs
- TicketsController#create should limit access to IP and be user/pass protected
- TicketsController#new should be configurable as open-to-the-world or not
- Integration with OpsWeekly
- Social media integration such as FreshDesk and Zoho have (reply to requests via social media)
- Ticket creation api (and improving existing api)
- Unread ticket status per user.
- Ticket search that also searches in from field and replies.
- Mark tickets as duplicate, linking it to the duplicated ticket.
- CC or BCC people from the reply form.

License

Brimir is licensed under the GNU Affero General Public License Version 3.

