

OpenMind: Know Your Customer

Contents

About OpenMind	3
Feedback	3
A Request	3
Installation	3
Install Ruby and Ruby on Rails	4
Get the Code	4
Create the Database Schema	4
Update database.yml	5
Update environment.rb	5
Install Image Libraries	5
Thinking Sphinx	5
Remaining Gems	6
Configure the mail server	6
Create Database Tables	7
Enable Observers	7
Populating Seed Data	7
Configure environment.yml	7
Configure the Web Service	8
Setup the Task Scheduler	8
Install the Daemons Gem	8
Start the Task Scheduler	8
Seed Data	9
Enterprises	9
Users	9
Configuration Options	10
Host	10
Port	10
Admin_email	10

Email_subject_prefix.....	10
Rescind_minutes.....	10
Allocation_expiration_days	11
Email_image_url	11
App_theme	11
Allocation_expiration_warning_days	11
Default_user_timezone	11
Max_file_upload_size	11
Minutes_to_send_vote_reminder.....	11
periodic_job_timeout	11
Themes.....	11
Periodic Jobs	12
Roles.....	12
Lookup Codes.....	13
Creating New Users.....	13

About OpenMind

OpenMind allows product companies to solicit feedback and input from users of their product.

Using the OpenMind application, user can:

- Create ideas that represents features they would like to see in the product
- View ideas that other users have created, and vote for those ideas which they would like to see added to the product
- Engage in online discussions with other users
- Mark ideas of interest to be watched

Using the OpenMind application, product managers can:

- See concrete information about which features users value most
- Move features to scheduled releases on the product roadmap
- Communicate roadmap decisions to the user community

If you'd like to see OpenMind in action, you can check it out at <http://openmind.scribesoftware.com>. You won't be able to sign up for an account if you are not a Scribe customer, but it should give you a good feeling for much of the functions and capabilities of OpenMind.

Feedback

We're interested in your feedback. If you have comments or suggestions for this manual or the OpenMind application, send email to bob@sturim.org.

If you use OpenMind, please consider allowing us to use your company's logo on our web page.

A Request

I wrote OpenMind primary to meet the needs of my company, [Scribe Software](#). I decided to make the code available as open source as a way to give back to the community. However, there is additional overhead in trying to keep the product suitable for a broader community than one company. I am happy to take on that burden if people find OpenMind useful.

If you find OpenMind useful, I ask that you send me an email to let me know at bob@sturim.org. And please, consider letting me add your logo to the openmind site: <http://openmind.sourceforge.net>.

Installation

Follow the steps outlined below to install and configure the OpenMind application. Unless otherwise noted, all commands are run from the Rails application root directory. For example, if you install the application in ~/openmind, then this directory is your rails application root directory. In examples below, we assume the directory is named "openmind".

Install Ruby and Ruby on Rails

You can install Ruby and Ruby gems by following the instructions at <http://rubyonrails.org/download>.

OpenMind has been tested to work with Rails 2.3.5. You can install Rails 2.3.5 with the following command:

```
gem install rails -v "2.3.5"
```

Get the Code

Starting with OpenMind 2.4, OpenMind is now hosting at GitHub (<http://www.github.com>) and uses git instead of subversion as its source code management system.

To install OpenMind, you will first need to install git, which you can do at <http://git-scm.com/>.

To download and install OpenMind, issue the following command:

```
git clone git://github.com/spob/openmind.git
```

This will install the OpenMind application in an application called "openmind" located within the directory from which you issued the git clone command.

Create the Database Schema

- Create a schema in a MySQL database for the OpenMind data.
- Create a user and password. Make note of the schema name, username and password, as well as the ip_address or machine name of the machine on which MySQL is running. Ensure that the user you created has read/write privileges to the schema and can create and drop tables, indices, etc.

Setup a database schema using the MySQL Databases icon in the Databases section of the cPanel. Create a schema for the openmind community. Note that Hosting Rails prefixes your account name to the database name you selected. For example, if you hostingrails account name is bobstur and you selected openmind as your database, then the schema name will be "bobstur_openmind".

Note your schema name here: _____

In the same screen in the database control panel, create a user. Again, Hosting Rails prepends the username. For example, I created a user called "openmind". The final username will end up being: bobstur_openmind.

Note the user information below:

DB Username: _____

DB Password: _____

In same screen , add the user to the schema you created. When prompted, grant your user “All Privileges” for your schema.

Update database.yml

Edit the file: openmind/config/database.yml

Assuming you are configuring this environment for production purposes, edit the section:

```
production:
  adapter: mysql
  database: OpenMind_production
  username: root
  password:
  host: localhost
  encoding: utf8
```

If you are configuring for development or test, configure the appropriate section accordingly.

You can leave the value of the host as “localhost”

Update environment.rb

If you’re configuring a production environment, in your config/environment.rb file you'll just need to uncomment the following line to confirm you're in production mode:

```
ENV['RAILS_ENV'] ||= 'production'
```

Install Image Libraries

If you plan on using the “captcha” functionality in the registration screen, you will need to install image processing libraries:

- Install the ImageMagick libraries (<http://www.imagemagick.org>).
- Install the RMagic gem (<http://rmagick.rubyforge.org/index.html>).

Thinking Sphinx

OpenMind uses Thinking Sphinx for full text searching.

To install Thinking Sphinx, you will first need to install Sphinx: http://freelancing-god.github.com/ts/en/installing_sphinx.html

To install Thinking Sphinx, type the following command:

```
gem install thinking-sphinx
```

You can visit the Thinking Sphinx web site for further details: http://freelancing-god.github.com/ts/en/installing_thinking_sphinx.html

When running in production, it is strongly suggested that you use a monitoring tool such as `monit` (<http://mmonit.com/monit/>). Below is an excerpt from a sample monitoring script that you can use as a starting point.

```
##### Sphinx #####
check process sphinx with pidfile
/apphome/log/searchd.production.pid
  start program = "/usr/local/bin/searchd --config
/home/bsturim/openmind/config/production.sphinx.conf"
  stop program = "/usr/local/bin/searchd --stop --config
/home/bsturim/openmind/config/production.sphinx.conf"

  if mem > 100 Mb for 3 cycles then restart
  if cpu > 80% for 5 cycles then restart
  if 20 restarts within 20 cycles then timeout
```

Remaining Gems

You can install remaining gems by typing:

```
rake gems:install
```

from the application home directory.

Configure the mail server

If you are configuring a production environment, edit the file:

`config/environments/production.rb` (edit `development.rb` and `test.rb` if you are setting up a development or test environment). Add entries to configure the mail server. For example:

```
ActionMailer::Base.smtp_settings = {
  :address => 'themailserver',
  :port => 27,
  :authentication => :plain,
  :user_name => 'emailusername',
  :password => 'emailpassword'
}
```

For example, the configuration settings to send email via a gmail account are:

```
ActionMailer::Base.smtp_settings = {
  :address => 'smtp.gmail.com',
  :port => 587,
  :domain => 'sturim.org',
  :authentication => :plain,
  :user_name => 'user@gmail.com',
  :password => 'password'
}
```

You can use `sendmail` by appending the following values to the `production.rb` file:

```
config.action_mailer.raise_delivery_errors = true
config.action_mailer.delivery_method = :sendmail
config.action_mailer.perform_deliveries = true
config.action_mailer.default_charset = "utf-8"
config.action_mailer.default_content_type = "text/html"
```

Create Database Tables

Populate the database by running the following command:

```
rake db:migrate
```

If you run into any errors during installation, you can view the log file at `logs/production.log` for details.

Enable Observers

After you've performed the initial migration, you must enable observers in the system by copy the file:¹

```
config/observe.no.txt
```

to

```
config/observe.txt
```

Populating Seed Data

OpenMind requires a minimal set of seed data in order to run properly. You can populate the database with the required seed data by issuing the following command:

```
rake db:seed
```

Configure environment.yml

Edit the file `config/environment.yml`. Set the options as appropriate. For further explanation on each option, see "Configuration Options" on page 8.

¹ This is a bit of a hack. It is required because of a catch 22 – when performing the initial migration, rake will cause `environment.rb` to be read, which will normally cause observers to be enabled. Since some of the models being observed have `acts_as_solr` declarations, `acts_as_solr` will then look for the tables, which have not yet been created. If anyone has a better idea of how to resolve this issue, please let me know.

Configure the Web Service

If you are running in production, at this point the app is configured. What remains is to set up the server of your choice (fast_cgi, mongrel, etc.). What follows are steps specific to Hosting Rails to get the application up and running.

Create the symlink so the www directory is your Rails app public folder. Get to your application root directory and:

```
[~]# mv ~/public_html ~/public_html_backup  
[~]# ln -s ~/yourapp/public ~/public_html
```

Notify Hosting Rails support to configure your apache vhost to use mod_rails.

Setup the Task Scheduler

Install the Daemons Gem²

The OpenMind task scheduler requires that the daemon's gem be installed. To verify that it is installed on your system, type:

```
gem list
```

and search for an entry for daemons.

To install the daemons gem, type:

```
gem install daemons
```

Start the Task Scheduler

OpenMind includes a task scheduler that runs background tasks – for example, checking for new entries in discussion forums once a day and sending out emails. You can start the task scheduler by issuing the following command:

```
ruby script/task_server_control.rb run -- -e production
```

to run in interactive mode, or

```
ruby script/task_server_control.rb start -- -e production
```

^{2 2} The application has frozen all of its required gems and plugins into the vendor directory of the application, and thus does not require any gems installed globally. This is important in hosted environments in which you may not have the permissions required to install gems globally. The daemons gem is an exception to this statement. The daemons gem must be installed in your ruby installation via the “gem install daemons” command. We will work to resolve this, and would appreciate any guidance if you have further ideas. For those of you using the hostingrails.com service, the daemons gem is installed on their servers and thus this is not an issue.

to run as a background task.

To stop the daemon, type:

```
ruby script/task_server_control.rb stop -- -e production
```

Note that if you make any changes to your environment.yml file, you'll need to stop and start the task server for those changes to be picked up for background tasks such as certain email alerts.

It is strongly suggested that you set up this script to be started automatically via cron or monit.d or another mechanism, so that it will start up automatically in the event of a server reboot.

When running in production, it is strongly suggested that you use a monitoring tool such as monit (<http://mmonit.com/monit/>). Below is a sample monitoring script that you can use as a starting point.

```
set daemon 60
set httpd port 2812 and
    use address localhost
    allow localhost
set logfile /home/openmin/openmind/log/monit.log
check process taskserver with pidfile
/home/openmin/openmind/tmp/pids/task_server.rb.pid
    start program = "/usr/local/bin/ruby
/home/openmin/openmind/script/task_server_control.rb start -f --
-e production"
    stop program = "/usr/local/bin/ruby
/home/openmin/openmind/script/task_server_control.rb stop -- -e
production"
group taskserver
```

Seed Data

When the application is first set up, it contains some basic data that you can modify to suit your needs.

Enterprises

The system will create a single enterprise, the Main Company. You can update this record to reflect your own business name.

Users

The system will create the following users:

- admin@openmind.org: This user is granted the admin role.
- prodmgr@openmind.org: This user is granted the product manager role.

- voter@openmind.org: This user is granted the voter role.
- readonly@openmind.org: This user is granted no roles.
- allocmgr@openmind.org: This user is granted the allocation manager role.
- all@openmind.org: This user is granted all roles

The password for each default user is “changeme” (without the quotes”).

These data exist to allow you to test your deployment, and to provide you a means to bootstrap your own users. It is strongly suggested that you create your own users and disable or delete the seeded users in any production environment.

Configuration Options

OpenMind provides a number of configuration options that you can specify in the config/environment.yml file.

Host

OpenMind sends email notifications when certain events occur. Many of those emails contains links to the application. The **host** and **port** parameters are used to construct the url for those links.

OpenMind constructs the url as <http://{host}:{port}/...>

Specify a value that will allow the url to resolve to your OpenMind instance.

Port

See “Host” above.

Admin_email

The email address of the administrator. This email will be used as the "from" address in emails sent by OpenMind. It is also appears in the page footers.

Email_subject_prefix

Email_subject_prefix holds the subject prefix. This will be the prefix in the subject for emails sent by OpenMind. For example, "OpenMind: Your new account has been created..."

Rescind_minutes

When a user votes, that user has a limited amount of time to rescind their vote before the vote becomes permanent. This parameter specifies, in minutes, the length of time before which a vote becomes permanent.

Allocation_expiration_days

Allocations expire after a specified number of days. When creating new allocations, the date on which the allocation is set to expire will default to this number of days from today.

Email_image_url

Html-formatted emails contain images. Such emails reference those images via a url. Specify the publically-accessible url to the directory in which those images are stores. Those images are shipped with the product beneath the public/themes/openmind/images/email directory.

App_theme

OpenMind is skinnable, allowing companies to tailor the look and feel to their own specific needs. By default, OpenMind ships with an OpenMind theme, beneath the public/themes directory. Companies wishing to create a new theme can create their own directory beneath the themes directory. This parameter should map to the name of that directory.

Allocation_expiration_warning_days

Users receive a warning when they log on if they have unused allocations that are about to expire. This parameter specifies, in days, how soon before an allocation is set to expire that user should be warned.

Default_user_timezone

This configuration option defines the default time zone when creating new users. To determine a list of valid time zone values local to your location, type:

```
rake time:zones:local
```

Max_file_upload_size

Users can add images when commenting on ideas and topics. This setting defines a maximum filesize allowed for attachments, in KB. Note this filesize does not apply to users who have privileges to add attachments in the attachments screen.

Minutes_to_send_vote_reminder

The number of minutes to wait before, when an idea is created, sending a reminder to the author of an idea to vote if they have not already done so. A value of -1 signifies no reminder should be sent.

periodic_job_timeout

The number of minutes to wait for a periodic job to complete. If the job does not complete in this timeframe, OpenMind will assume it is dead and initiate a recovery.

Themes

OpenMind is skinnable. This feature is intended to allow you to tailor the appearance of OpenMind to match your corporate style. Each “skin” is called a theme.

Themes are stored beneath the public/themes folder. OpenMind ships with a default theme called “openmind”. If you’d like to customize the appearance of OpenMind, perform the following steps:

1. Copy the openmind directory to another directory with a different name within the themes directory. For example, copy public/themes/openmind to public/themes/acme.
2. Customize the icons, images and stylesheets within your new directory to suit your needs. Note that you cannot delete or rename files without risking causing problems with the screen layout, but you can edit them as much as you’d like.
3. Edit the app_theme property in the environment.yml file to point to your new directory. See “App_theme” on page 11.

Periodic Jobs

OpenMind includes the ability to run jobs in the background at specified intervals. In order for those jobs to run, the Task Scheduler must be running. See “Setup the Task Scheduler” on page 8.

There are three types of periodic jobs:

- A RunOncePeriodicJob executes one-time only. These jobs are automatically purged on a regularly basis after they’ve executed.
- A RunIntervalPeriodJob runs every x seconds, where x is defined by the value in the interval column. These jobs are never purged.
- A RunAtPeriodicJob runs at the same time every day. That time is determined by the value of the run_at_minutes column, where run_at_minutes represents the number of minutes after minute the job should execute. These jobs are never purged.

Periodic jobs are recorded in the “periodic_jobs” table in the database. By modifying the values in the columns, you can change the behavior of OpenMind.

OpenMind currently uses three periodic jobs:

- A RunAtPeriodicJob by default runs at 3AM each morning, checking for new topics and topic comments in the forum area and notifying individuals watching those topics via email of those new comments.
- When a user creates a comment against an idea, a RunOncePeriodicJob is created to check for watchers in the background, and notify those users about the new comment.
- Once an hour a RunIntervalPeriodJob deletes RunOncePeriodicJobs older than 7 days.

Roles

OpenMind provides the following roles:

- System Administrator creates and edit enterprises, users and lookup codes
- Product Manager manages products, releases, ideas, announcements and polls

- Voter can vote for ideas. Most users in the system will probably be voters.
- Allocations Manager manages the granting of allocations
- Moderator can moderate forum discussions, editing or deleting comments they deem in appropriate.

Each user can be granted zero, one or multiple roles to determine which functions they can perform.

Lookup Codes

Lookup codes provide a mechanism to allow users to easily change pick list values. The system administrator can modify these values from the “Lookup Codes” menu. Currently OpenMind uses the following lookup codes:

1. Release Status defines the lifecycle states of a product release.
2. Forum Group defines the groupings that appear in the forums page. Forum mediators can assign forums to a forum group. Forum groups are optional.
3. Enterprise Types define classifications that can be assigned to Enterprises. Enterprise types are optional.

Creating New Users

The creation of users follows the following steps:

1. Create the enterprise using the “Enterprises” menu for the user you intend to create if it does not already exist. Each user must belong to an enterprise.
2. Create the user using the “Users” menu.
3. OpenMind will send an email to the user with the user’s temporary password and activation code.
4. The user follows the link provided in the email and enters their activation code.
5. The users account is enabled and the user is sent an email informing them that activation was successful.
6. The user logs in with their temporary password. They will be prompted to change their password upon login.

OpenMind also provides a bulk load feature for creating enterprises and users. The bulk load feature is accessible from the “Users” menu entry.