

OpenMind: Know Your Customer

Contents

- About OpenMind.....3
- Feedback3
- Installation.....3
 - Create the Database Schema3
 - Update database.yml4
 - Update environment.rb4
 - Installing ruby-debug.....5
 - Native Code Gems5
 - MySQL5
 - Configure the mail server.....5
 - Populate the Database6
 - Configure environment.yml6
 - Configure the Web Service.....6
 - Install Image Libraries6
 - Setup SOLR7
 - Setup the Task Scheduler8
 - Install the Daemons Gem8
 - Start the Task Scheduler.....8
- Seed Data9
 - Enterprises.....9
 - Users.....9
- Configuration Options10
 - Host10
 - Port.....10
 - Admin_email10
 - Email_subject_prefix.....10
 - Rescind_minutes10
 - Allocation_expiration_days.....10

Email_image_url.....	10
App_theme.....	10
Allocation_expiration_warning_days.....	11
Default_user_timezone.....	11
Themes.....	11
Periodic Jobs.....	11
Roles.....	12
Lookup Codes.....	12
Creating New Users.....	12

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

Feedback

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

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

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 unzip the openmind.zip file to ~/openmind, then this directory is your rails application root directory. In examples below, we assume the directory is named "openmind".

There are a wide variety of rails hosting companies available for you to choose from if you opt to use a third party service to host OpenMind. We have used Hosting Rails (<http://www.hostingrails.com>) and have found them to be generally reliable. For your convenience, we have included setup steps specific to the Hosting Rails service highlighted in the instructions below.

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'
```

Installing ruby-debug

If you plan on using your development environment, then you will need to install the ruby debugger:

```
gem install ruby-debug
```

Native Code Gems

OpenMind depends upon several gems that rely upon native code. Because of this, such gems cannot be frozen with the OpenMind installation and must be installed on your particular server.

MySQL

OpenMind requires the driver my MySQL. To install it, type:

```
gem install mysql
```

The mysql gem is already installed on HostingRails servers

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"
```

Note: The migration performed in step 0 below creates record in the database which will cause observes to send email. Therefore, you must configure the mail server as outlined in this step before you attempt to run the migration. Alternatively, you can uncomment the following line in production.rb (though you will probably want to re-comment the line at a later point):

```
# config.action_mailer.raise_delivery_errors = false
```

Populate the Database

Populate the database by running the following command:

```
rake db:migrate
```

You will need to ensure that the SOLR server is running before performing the migration. See “Setup SOLR” on page 7.

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

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 6.

Configure the Web Service

At this point, the app is configured, and wait 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.

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>).

Both ImageMagick and RMagic are already installed on HostingRails servers

Setup SOLR

You must setup OpenMind to enable full text searching using the Solr library.

Solr requires java version 1.5 or later be installed.

Solr requires a solr server be running to automatically update search indexes when records change. Use the following rake command to start the server on windows:

```
rake solr:startwin
```

and

```
rake solr:start
```

for all other environments.

You can stop the server with:

```
rake solr:stop
```

You will need to ensure that the SOLR server is running in order to perform any tasks that touch comments or topics, including performing the initial database migration.

If you're upgrading from an earlier version of OpenMind, you'll need to manually regenerate the search indexes to start from within the ruby on rails console:

```
ruby script/console
Loading development environment (Rails 2.1.1)
Requiring REXML
>> Topic.rebuild_solr_index
=> "Index for Topic has been rebuilt\n"
>> TopicComment.rebuild_solr_index
=> "Index for TopicComment has been rebuilt\n"
```

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 solr with pidfile
/home/openmin/openmind/tmp/pids/production_pid
  start program = "/home/openmin/bin/solr_start"
  stop program = "/home/openmin/bin/solr_stop"
group solr
```

Setup the Task Scheduler

Install the Daemons Gem¹

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
```

to run as a background task.

To stop the daemon, type:

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

¹ 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.

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
```

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 10.

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 “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>).

Both ImageMagick and RMagic are already installed on HostingRails servers

Setup SOLR

You must setup OpenMind to enable full text searching using the Solr library.

Solr requires java version 1.5 or later be installed.

Solr requires a solr server be running to automatically update search indexes when records change. Use the following rake command to start the server on windows:

```
rake solr:startwin
```

and

```
rake solr:start
```

for all other environments.

You can stop the server with:

```
rake solr:stop
```

You will need to ensure that the SOLR server is running in order to perform any tasks that touch comments or topics, including performing the initial database migration.

If you're upgrading from an earlier version of OpenMind, you'll need to manually regenerate the search indexes to start from within the ruby on rails console:

```
ruby script/console
Loading development environment (Rails 2.1.1)
Requiring REXML
>> Topic.rebuild_solr_index
=> "Index for Topic has been rebuilt\n"
>> TopicComment.rebuild_solr_index
=> "Index for TopicComment has been rebuilt\n"
```

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 solr with pidfile
/home/openmin/openmind/tmp/pids/production_pid
    start program = "/home/openmin/bin/solr_start"
    stop program = "/home/openmin/bin/solr_stop"
    group solr
```

Setup the Task Scheduler" on page 6.

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. Currently OpenMind only uses a single type of lookup code – Release Status – which defines the lifecycle states of a product release. The system administrator can modify these values from the “Lookup Codes” menu.

OpenMind may add additional lookup code types at a future date.

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.