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”.

1. 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 privledges 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 pre-pepends 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.

1. 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

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

If you’re using a shared machine, you can leave the value of the host as “localhost”

1. 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'

1. 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 hostingrails, 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 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

1. Populating the database by running the following command:

rake db:migrate

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

1. Edit the file config/environment.yml. Set the options as appropriate. Each option is documented in line.

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

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

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.