leihs administration and installation guide

COLLABORATORS			
	TITLE : leihs administration and installation guide		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Ramon Cahenzli	October 11, 2010	

REVISION HISTORY			
NUMBER	DATE	DESCRIPTION	NAME

Contents

1 Introduction

leihs is web-based inventory handling and resource booking system. It allows users to view available equipment and place reservations through the frontend. Inventory managers and sysadmins use the backend to handle incoming reservations and manage items in the inventory.

This guide shows you how to install a leihs server. The guide is written from the perspective of a system administrator or developer. If you are interested in running leihs in your own organization but aren't a sysadmin, talk to your IT department. leihs is not intended to be installed on a client, so any software installation on your own machine isn't necessary. All you need is a web browser.

Consulting and installation services are also available from independent companies supporting Free Software all around the world. Ask around for a company or individual who knows Ruby on Rails applications, you will surely find someone who can help you install leihs.

If you are such a person yourself and would like to have your services listed on the leihs project website and in this document, please write me an e-mail.

2 Quick Install

This section is meant for admins experienced with installing Ruby on Rails applications. It lists the necessary installation steps in the briefest possible way, so you can get up and running quickly.

2.1 Installation on Mac OS X

The following steps were tested on Mac OS X 10.5 and 10.6 on an x86. They may also work for later versions.

1. Install Apple Xcode. This is required because some modules may need a C/C++ compiler. XCode is available from Apple after a free developer registration.
2. Install MySQL 5.0. Use the 32-bit version if you are running OS X Leopard (10.5) but use the 64-bit version if you're on Snow Leopard (10.6). **Make absolutely sure to install the 32-bit version on Leopard (10.5) even if you are running a 64-bit operating system.** Some bugs in Mac OS X's MySQL and Ruby integration make it impossible to continue otherwise.
3. Update the local gem system:

   ```
   $ sudo gem update --system
   ```

4. Install the native MySQL gem:

   If you are installing on Snow Leopard (OS X 10.6) use the following installation options:

   ```
    $ sudo env ARCHFLAGS='-arch x86_64' \
   gem install mysql -- --with-mysql-config=/usr/local/mysql/bin/mysql_config
   ```

   If you are using Leopard (OS X 10.5) or below, use the following string instead:

   ```
    $ sudo gem install mysql -- --with-mysql-config=/usr/local/mysql/bin/mysql_config \
   --with-mysql-dir=/usr/local/mysql --with-mysql-lib=/usr/local/mysql/lib
   \ --with-mysql-include=/usr/local/mysql/include
   ```

5. Download the latest version of leihs from our SourceForge project page. Unpack it to a convenient directory. We use the home directory of the *leihs* user (/Users/leihs) to install leihs in. Of course you can use any directory.

6. Configure database access for this installation of leihs. Copy the file config/database.yml.example to config/database.yml and set things up according to your needs. You will need a MySQL database for leihs. Here is an example of a development-mode database configuration:

```
production:
   adapter: mysql
   database: leihs2_production
   encoding: utf8
   username: root
   password:
   host: localhost
   port: 3306
```

7. Create and migrate the database:

```
$ RAILS_ENV='production' rake db:migrate
```

8. Create any temporary directories that are necessary for e.g. image uploads, temporary files etc. Make sure to create these directories so that the leihs user has write permission to them.

```
$ cd /home/leihs
$ mkdir -p public/images/attachments
$ mkdir -p tmp/sessions
$ mkdir tmp/cache
$ mkdir -p log
```

9. Download Sphinx (a fulltext search system) and install thinking-sphinx (a gem). In this example we also include libstemmer, a library that allows for word stem searching in various languages. We use version 0.9.9:

```
$ cd /tmp
$ wget http://sphinxsearch.com/downloads/sphinx-0.9.9.tar.gz
$ tar xvfz sphinx-0.9.9.tar.gz
$ cd sphinx-0.9.9
$ wget http://snowball.tartarus.org/dist/libstemmer_c.tgz
$ tar xvfz libstemmer_c.tgz
$ ./configure --with-libstemmer && make
$ sudo make install
$ sudo gem install thinking-sphinx --source http://gemcutter.org -v 1.3.15
```

10. Install the required version of Rails as well as a few gems that cannot be installed automatically:

```
$ cd /home/leihs
$ sudo gem install -v=2.3.5 rails
$ sudo gem install thinking-sphinx -v 1.3.15
$ sudo gem install prawn -v 0.8.4
$ sudo gem install rake
$ sudo rake gems:install
```

11. Configure and start the Sphinx server:

```
$ RAILS_ENV='production' rake ts:config
$ RAILS_ENV='production' rake ts:reindex
$ RAILS_ENV='production' rake ts:start
```

12. Start the server:

```
$ RAILS_ENV='production' ./script/server
```

Now you should see your local leihs server at http://localhost:3000. You can log in with username "super_user_1" and password "pass".

This gives you a test setup using the pure Ruby WebRick web server. For production setups, we recommend mod_passenger. See the "Installing a production environment" section of this guide for more information.

13. Set up a system cronjob that sends nightly e-mail reminders and, more importantly, updates all the models' availability counts. There are many ways to schedule repeating tasks on Mac OS X. Please see Apple's documentation on launchd for how to do this.

    In crontab format, an example job for a production environment would look like this:

    ```
    1 00    * * *   cd /home/leihs && RAILS_ENV='production' rake leihs:cron
    ```

    The important bit here is to run the "leihs:cron" rake task. How you do this exactly is irrelevant.

14. Optional: If you want to use images of inventory items, install ImageMagick. On Mac OS X, we recommend using MacPorts and then installing the ImageMagick port:

    ```
    $ sudo port install imagemagick
    ```

    Make sure that the ImageMagick binaries (e.g. convert) are in your $PATH. Please consult the Mac OS X documentation to find out how to add paths to your $PATH environment variable. It can be done e.g. by editing /etc/profile and adding the following lines:

    ```
    PATH=$PATH:/opt/local/bin
    export PATH
    ```

    But Apple may at any point recommend a different approach for configuring your PATH, so please don't take our word for it.

    Finally, install the rmagick gem:

    ```
    $ sudo gem install rmagick
    ```

2.2 Installation on Debian GNU/Linux

These instructions were tested on a minimal install of Debian GNU/Linux 5.0 (Lenny).

1. Install Ruby and irb

   ```
   # apt-get install ruby irb rdoc libopenssl-ruby ruby-dev
   ```

2. Install RubyGems from the RubyGems website. Make sure **not to install** the edition of RubyGems that is available from Debian's package archives. RubyGem development moves so quickly that we need to use the one from upstream.

   ```
   # cd /tmp
   # wget http://rubyforge.org/frs/download.php/56227/rubygems-1.3.3.tgz
   # tar xvfz rubygems-1.3.3.tgz
   # cd rubygems-1.3.3
   # ruby setup.rb
   # ln -s /usr/bin/gem1.8 /usr/bin/gem
   ```

   Note that the URL above might change! Please visit the RubyGems site to find the exact URL under "Downloads".

3. Install the MySQL header files and the MySQL gem:

```
# apt-get install libmysqlclient15-dev make
# gem install mysql
```

4. Download the latest version of leihs from our SourceForge project page. Unpack it to a convenient directory. We use the home directory of the *leihs* user (/home/leihs) to install leihs in. Of course you can use any directory.

5. Configure database access for this installation of leihs. Copy the file config/database.yml.example to config/database.yml and set things up according to your needs. You will need a MySQL database for leihs. Here is an example of a development-mode database configuration:

```
production:
   adapter: mysql
   database: leihs2_production
   encoding: utf8
   username: root
   password:
   host: localhost
   port: 3306
```

6. Create and migrate the database:

```
# su - leihs
$ RAILS_ENV='production' rake db:migrate
```

7. Create any temporary directories that are necessary for e.g. image uploads, temporary files etc. Make sure to create these directories so that the leihs user has write permission to them.

```
$ cd /home/leihs
$ mkdir -p public/images/attachments
$ mkdir -p tmp/sessions
$ mkdir tmp/cache
$ mkdir -p log
```

8. Download Sphinx (a fulltext search system) and install thinking-sphinx (a gem). In this example we also include libstemmer, a library that allows for word stem searching in various languages. We use version 0.9.9:

```
$ cd /tmp
$ curl -C - -O http://sphinxsearch.com/downloads/sphinx-0.9.9.tar.gz
$ tar xvfz sphinx-0.9.9.tar.gz
$ cd sphinx-0.9.9
$ curl -C - -O http://snowball.tartarus.org/dist/libstemmer_c.tgz
$ tar xvfz libstemmer_c.tgz
$ ./configure --with-libstemmer && make
$ su
# make install
# gem install thinking-sphinx --source http://gemcutter.org -v 1.3.15
```

9. Install the required version of Rails as well as a few gems that cannot be installed automatically:

```
# cd /home/leihs
# gem install -v=2.3.5 rails
# gem install thinking-sphinx -v 1.3.15
# gem install prawn -v 0.8.4
# gem install rake
# rake gems:install
```

10. Configure and start the Sphinx server:

```
$ cd /home/leihs
$ RAILS_ENV='production' rake ts:config
$ RAILS_ENV='production' rake ts:reindex
$ RAILS_ENV='production' rake ts:start
```

11. Start the leihs server:

```
$ cd /home/leihs
$ RAILS_ENV='production' ./script/server
```

    Now you should see your local leihs server at http://localhost:3000. You can log in with username "super_user_1" and password "pass".

    This gives you a test setup using the pure Ruby WebRick web server. For production setups, we recommend mod_passenger. See the "Installing a production environment" section of this guide for more information.

12. Set up a system cronjob that sends nightly e-mail reminders and, more importantly, updates all the models' availability counts. There are many ways to schedule repeating tasks on GNU/Linux, but here's a line in crontab-format that you can add to your leihs user's crontab using e.g. `crontab -e`:

```
1 00    * * *   cd /home/leihs && RAILS_ENV='production' rake leihs:cron
```

    The important bit here is to run the "leihs:cron" rake task. How you do this exactly is irrelevant.

13. Optional: If you want to use images of inventory items, install ImageMagick:

```
# apt-get install imagemagick graphicsmagick-libmagick-dev-compat
```

    And then install the rmagick gem:

```
# gem install rmagick
```

3 Default admin username/password

After installation, a default user is created for the Database Authentication module. Username: super_user_1. Password: pass.

4 Upgrading from leihs 2.0 to leihs 2.1

The upgrade should be painless.

1. Download the new leihs version (in this example, the .tar.gz version is used) and unpack it to a new location:

```
$ tar xvfz leihs-2.1.tar.gz
$ cd leihs-2.1
```

2. Copy any images uploaded to your old leihs version so they are also available in the new leihs:

```
$ rm -rf public/images/attachments
$ cp -pr ../leihs-2.0/public/images/attachments public/images/
```

3. Copy your old database configuration file:

```
$ cp ../leihs-2.0/config/database.yml config/
```

4. Look at your old config/environment.rb file in a text editor. Open the new config/environment.rb. Copy the configuration items from your old config to your new config as necessary. You might find new options in the new environment.rb that weren't present in the old one. Save your new configuration.
5. Run the database migration to get all your data up to the new version:

```
$ RAILS_ENV='production' rake db:migrate
```

6. Rerun the steps from the section "Configure and start the Sphinx server" relevant for your operating system in order to create/update your fulltext search index and make sure Sphinx is running.
7. Run the server. Make sure you stop your previous leihs server!

```
$ RAILS_ENV='production' ./script/server
```

If everything went correctly, you should see leihs coming up at http://localhost:3000.

5 Upgrading from leihs 2.1 to leihs 2.2

The upgrade should be painless.

1. Download the new leihs version (in this example, the .tar.gz version is used) and unpack it to a new location:

```
$ tar xvfz leihs-2.2.tar.gz
$ cd leihs-2.2
```

2. Copy any images uploaded to your old leihs version so they are also available in the new leihs:

```
$ rm -rf public/images/attachments
$ cp -pr ../leihs-2.1/public/images/attachments public/images/
```

3. Copy your old database configuration file:

```
$ cp ../leihs-2.1/config/database.yml config/
```

4. Look at your old config/environment.rb file in a text editor. Open the new config/environment.rb. Copy the configuration items from your old config to your new config as necessary. You might find new options in the new environment.rb that weren't present in the old one. Save your new configuration.
5. Run the database migration to get all your data up to the new version:

```
$ RAILS_ENV='production' rake db:migrate
```

6. Run the server. Make sure you stop your previous leihs server!

```
$ RAILS_ENV='production' ./script/server
```

If everything went correctly, you should see leihs coming up at http://localhost:3000.

6 Upgrading from leihs 2.2 to leihs 2.3

The upgrade should be painless, but please note that the access levels were replaced by user groups in this release. This can give you almost the same behavior, but it's different to manage. You now specify groups in the admin interface or your inventory pool's backend, and then specify how many items of each model that each group can have.

If you had existing user levels in your system, these will be automatically converted to user groups during the database migration (db:migrate). This is documented below.

1. Download the new leihs version (in this example, the .tar.gz version is used) and unpack it to a new location:

   ```
   $ tar xvfz leihs-2.3.tar.gz
   $ cd leihs-2.3
   ```

2. Copy any images uploaded to your old leihs version so they are also available in the new leihs:

   ```
   $ rm -rf public/images/attachments
   $ cp -pr ../leihs-2.2/public/images/attachments public/images/
   ```

3. Copy your old database configuration file:

   ```
   $ cp ../leihs-2.2/config/database.yml config/
   ```

4. Look at your old config/environment.rb file in a text editor. Open the new config/environment.rb. Copy the configuration items from your old config to your new config as necessary. You might find new options in the new environment.rb that weren't present in the old one. Save your new configuration.

5. Run the database migration to get all your data up to the new version:

   ```
   $ RAILS_ENV='production' rake db:migrate
   ```

6. Run the server. Make sure you stop your previous leihs server! If you use mod_passenger, do `touch tmp/restart.txt` instead.

   ```
   $ RAILS_ENV='production' ./script/server
   ```

7. Stop the Sphinx server that's running for leihs 2.2.:

   ```
   $ cd leihs-2.2
   $ RAILS_ENV=production rake ts:stop
   ```

8. Reindex Sphinx at the new location and restart the Sphinx server:

   ```
   $ cd leihs-2.3
   $ RAILS_ENV=production rake ts:config
   $ RAILS_ENV=production rake ts:reindex
   $ RAILS_ENV=production rake ts:start
   ```

If everything went correctly, you should see leihs coming up at http://localhost:3000 or, if using mod_passenger, at the location you configured.

7 Installing a production environment

Please note that this quick-start guide does not cover running a Ruby on Rails application for production. Please look into Phusion Passenger or the Apache Mongrel cluster in the reference section for how to set up a production environment.

Without a real production environment, leihs can handle upwards of 1000 users (not concurrently, of course!). If you need better performance or want an easier to handle setup, you must set up a proper production environment.

8 Free Software Statement

The Zurich University of the Arts supports Free Software as defined by the Free Software Foundation. That's why leihs is Free Software licensed under the GNU GPL version 3.0.

One of the advantages this freedom brings with it is that it enables anyone in the world to provide local support services for leihs at the same quality level as the Zurich University of the Arts can provide itself.

If you would like to take part in the development of leihs, please see our project page.

9 References

A ZHdK configuration file for mongrel cluster

As an example to help you set up a production environment, here is the Apache configuration we use. You will need mod_proxy:

```
<VirtualHost *:80>
   ServerName ausleihe.zhdk.ch

   DocumentRoot /home/rails/leihs/leihs/public/
   ErrorLog /var/log/apache2/leihs/error.log

   # Which Rails environment to use (production, testing, production)
   DefaultInitEnv RAILS_ENV production
   SetEnv RAILS_ENV production

  # Proxy balancer for leihs
  <Proxy balancer://leihs_cluster>
    BalancerMember http://127.0.0.1:10010
    BalancerMember http://127.0.0.1:10011
    BalancerMember http://127.0.0.1:10012
    BalancerMember http://127.0.0.1:10013
  </Proxy>


   # We want to completely ignore the application's own
   # .htaccess, as all relevant options are configured
   # right here in this file.
   <Directory /home/rails/leihs/leihs/public>
      AllowOverride none
   </Directory>


   # Don't do forward proxying
   ProxyRequests Off

   # Enable reverse proxying
   <Proxy *>
     Order deny,allow
     Allow from all
   </Proxy>

   RewriteEngine On

   # Check for maintenance file. Let apache load it if it exists
   RewriteCond %{DOCUMENT_ROOT}/system/maintenance.html -f
   RewriteRule . /system/maintenance.html [L]

   # Rewrite index to check for static
```

```
   RewriteRule ^/$ /index.html [QSA]

   # Let apache serve static files (send everything via mod_proxy that
   # is *no* static file (!-f)
   RewriteCond %{DOCUMENT_ROOT}%{REQUEST_FILENAME} !-f
   RewriteRule .* balancer://leihs_cluster%{REQUEST_URI} [L,P,QSA]
</VirtualHost>
```