



Rancho: Open Source Group/Project Management Tool

Author: The Rancho Team
Contact: info@getrancho.com
Date: 2009-06-25
Web site: <http://www.getrancho.com>
License: Rancho is published under the GNU AGPL v3 license.

Contents

Rancho: Open Source Group/Project Management Tool	1
Introduction	1
System Requirements	1
Getting the Code	2
Configuration	2
External Modules	2
Installation on Debian	2
Upgrading Rancho	3
Important Configuration	3
Deploying Rancho in mod_wsgi and mod_python	4
mod_wsgi	4
mod_python	5

Introduction

Rancho is an Open Source web based tool developed with Django to manage groups of people and projects.

This document gives a brief explanation on how to install Django. Please note that deploying or optimizing Rancho is not the purpose of this document, thus, such subjects are not covered.

The installation of Rancho on a Debian based system using the PostgreSQL database is also included in order to provide an example.

System Requirements

Here are the system requirements to run Rancho.

- Linux/UNIX operating system;
- Django (<http://www.djangoproject.com>);
- MySQL<=4.0.1 (<http://www.mysql.com>) OR PostgreSQL<=8.3 (<http://www.postgresql.com>);
- Python MySQLDB (<http://mysql-python.sourceforge.net/>) in case of using the MySQL database;
- Psycopg (<http://www.initd.org>) in case of using the PostgreSQL database;
- pytz (<http://pytz.sourceforge.net/>);
- PIL - Python Image Library - (<http://www.pythonware.com/products/pil/>);
- ReportLab<=2.2 (<http://www.reportlab.org/downloads.html>);
- html5lib - Library for working with HTML5 documents (<http://code.google.com/p/html5lib/>);

- XHTML2PDF - HTML to PDF converter (<http://www.xhtml2pdf.com/>)
- ElementTree - A container object, designed to store hierarchical data structures. THIS IS ONLINE NEEDED IN PYTHON<=2.4 (<http://effbot.org/zone/element-index.htm>)

Getting the Code

Rancho's latest release can be downloaded from:

<http://www.getrancho.com/download>

The GIT version of Rancho can be checked out using the following command:

```
$ git clone http://git.getrancho.com/rancho.git
```

Configuration

Once the system requirements are met and the code is downloaded, the first thing to do is to edit the `settings_customize.py` file found on the project's root folder. This file already contains several strings that give hints on how to configure it.

External Modules

Rancho uses some external Django apps. There is a script to easily get those apps called `get_externals.sh`. To run the script you can do as follows:

```
$ ./get_externals.sh
```

Installation on Debian

This section covers the installation of Django on a Debian system (or Debian based like Ubuntu). The database used for this example is PostgreSQL which is assumed to be already installed locally as well as Rancho. For this example, the database user is assumed be 'jsmith' with the password 'qwerty'.

Run the following command as superuser to install the other requirements:

```
# apt-get install python-psycopg2 python-tz python-imaging python-reportlab python-html5lib  
# easy_install pisa
```

Once the packages finish being installed, create a database to be used by Rancho:

```
$ createdb ranchodb
```

After this, Rancho needs to be configured. Edit the `settings_customize.py` file as follows:

```
SEARCH_ENGINE = 'postgresql'  
  
DATABASE_ENGINE = 'postgresql'  
  
DATABASE_NAME = 'ranchodb'  
  
DATABASE_USER = 'jsmith'  
  
DATABASE_PASSWORD = 'qwerty'
```

```
DATABASE_HOST = ''  
  
DATABASE_PORT = ''
```

After the database settings, the email settings need to be configured in order for notifications to be sent. The values given to the settings below are also fictional in order to better exemplify the configuration:

```
DEFAULT_FROM_EMAIL = 'no-reply@example.com'  
  
EMAIL_HOST = 'smtp.example.com'  
  
EMAIL_HOST_USER = 'jsmith@example.com'  
  
EMAIL_HOST_PASSWORD = 'qwerty'
```

After this, rename the file as settings.py and that's it with Rancho configuration. The next step is to create the database tables and Rancho's initial information like the account owner and the default company:

```
$ python manage.py syncdb
```

Answer yes to the question to create a superuser and then supply the rest of the information asked.

Once the previous command is finished, it's time to run Rancho. For this, the Django test server is used:

```
$ python manage.py runserver
```

If everything went okay, you can now visit <http://localhost:8000> in your web browser to access Rancho. Log in with the information you supplied previously and that's it! Rancho is installed!

Important: Note that it is not advisable to use this server for production.

Upgrading Rancho

If you have a Rancho version installed and upgrade it, there might be changes to the database that must be solved in order, for example, to preserve data.

Important: It is advisable to back up all your data before upgrading Rancho.

The first thing to do after having upgraded Rancho is to synchronize the database using the following command:

```
$ python manage.py syncdb
```

If there are changes to the database, a message will be returned warning that an evolution is needed.

To perform the evolution, run the command:

```
$ python manage.py evolve --execute
```

If all went well, Rancho will be ready to be run.

Important Configuration

When using Rancho in a production environment you have to configure some additional items.

Sending files through Django is very inefficient so you should configure Rancho to use Apache mod-xsendfile. You can get it from: http://tn123.ath.cx/mod_xsendfile/

HOW_SEND_FILE='apache-modsendfile'

You should also define a value for SECRET_KEY since this is used to provide a seed in secret-key hashing algorithms. Set this to a random string -- the longer, the better.

Finally you should setup the cron:

```
# crontab -e
```

add the following lines:

```
0 23 * * * RANCHO_INSTALL_DIR/rancho/manage.py run_cron
0 * * * * RANCHO_INSTALL_DIR/rancho/manage.py send_mail
```

The cron is run every day at 23h00 and the mail is sent every hour.

Deploying Rancho in mod_wsgi and mod_python

Here you have a sample deployment for mod_wsgi and mod_python. Visit the Django web site to get more complete information on how to deploy Rancho in a production environment.

mod_wsgi

Create a new site configuration for apache with:

```
<VirtualHost *>
    ServerName sitename.com
    Include "INSTALL_DIR/rancho/apache/apache_django_wsgi.conf"
    ErrorLog /var/log/apache2/error.log
    LogLevel warn
    CustomLog /var/log/apache2/access.log combined
    ServerSignature Off
    XSendFile on
    XSendFileAllowAbove on
</VirtualHost>
```

Create a directory for apache configuration in your INSTALL_DIR and create two files:

```
apache_django_wsgi.conf

Alias /media/ "RANCHO_INSTALL_DIR/rancho/media/"
<Directory "RANCHO_INSTALL_DIR/rancho/media">
    Order allow,deny
    Options -Indexes
    Allow from all
    AddOutputFilterByType DEFLATE text/html text/plain text/xml text/css application/x-javascript
</Directory>

WSGIScriptAlias / "RANCHO_INSTALL_DIR/rancho/apache/rancho.wsgi"
<Directory "RANCHO_INSTALL_DIR/rancho/apache/">
    Allow from all
</Directory>

rancho.wsgi

import os, sys
```

```
#Calculate the path based on the location of the WSGI script.
apache_configuration= os.path.dirname(__file__)
```

```
project = os.path.dirname(apache_configuration)
workspace = os.path.dirname(project)
sys.path.append(workspace)
sys.path.append(project)
os.environ['DJANGO_SETTINGS_MODULE'] = 'rancho.settings'
import django.core.handlers.wsgi
application = django.core.handlers.wsgi.WSGIHandler()
```

mod_python

Create a new site configuration for apache with:

```
<VirtualHost *>
  <Location "/">
    SetHandler python-program
    PythonHandler django.core.handlers.modpython
    SetEnv DJANGO_SETTINGS_MODULE rancho.settings
    PythonDebug On
    PythonPath "['RANCHO_INSTALL_DIR', 'RANCHO_INSTALL_DIR/rancho' ] + sys.path"
  </Location>
</VirtualHost>
```