

Installation de Redmine

Système d'exploitation

Redmine devrait fonctionner sur la plupart des Unix, Linux, Mac et systèmes Windows aussi longtemps que Ruby est disponible sur ces plates-formes.

Ruby, Ruby on Rails, RubyGem

- Les versions de Ruby on Rails pour une version donnée de Redmine sont les suivantes :

Version de Redmine	Versions de Ruby supportées	Version de Rails exigée	Version de RubyGems exigée
current trunk	ruby 1.8.7, 1.9.2, 1.9.3, 2.0.0, jruby-1.6.7, jruby-1.7.2	Rails 3.2.13	
2.3.0	ruby 1.8.7, 1.9.2, 1.9.3, 2.0.0, jruby-1.6.7, jruby-1.7.2	Rails 3.2.13	
2.2.4	ruby 1.8.7, 1.9.2, 1.9.3, 2.0.0, jruby-1.6.7, jruby-1.7.2	Rails 3.2.13	
2.1.0	ruby 1.8.7, 1.9.2, 1.9.3, jruby-1.6.7	Rails 3.2.8	RubyGems <= 1.8
2.0.3	ruby 1.8.7, 1.9.2, 1.9.3, jruby-1.6.7	Rails 3.2.6	RubyGems <= 1.8
2.0.2	ruby 1.8.7, 1.9.2, 1.9.3, jruby-1.6.7	Rails 3.2.5	RubyGems <= 1.8
2.0.0, 2.0.1	ruby 1.8.7, 1.9.2, 1.9.3, jruby-1.6.7	Rails 3.2.3	RubyGems <= 1.8
1.4.x	ruby 1.8.7, 1.9.2, 1.9.3, jruby-1.6.7	Rails 2.3.14	RubyGems <= 1.8

Base de données

- MySQL 5.0 ou plus (recommandé)
 - Assurez-vous d'avoir installé les liens C pour Ruby qui améliorent considérablement les performances. Vous pouvez les obtenir en exécutant la commande `gem install mysql2`. Si vous rencontrez des problèmes lors de l'installation de cette gem, consultez la [page dédiée de Ruby on Rails](#)
 - les versions Mysql 4.1 et 4.X supérieurs peuvent poser des problèmes
- PostgreSQL 8 ou plus (8.2 ou plus à partir de Redmine 2.0.0)
 - Il faut être sûr que le format de date (datestyle) de votre base de données est positionné à "ISO" (paramètre par défaut de Postgres). Vous pouvez la positionner avec la commande: `ALTER DATABASE "redmine_db" SET datestyle="ISO,MDY";`
 - Certaines anomalies dans PostgreSQL 8.4.0 et 8.4.1 affectent le comportement de Redmine ([#4259](#), [#4314](#)), elles sont corrigées dans PostgreSQL 8.4.2

- SQLite 3

Composants optionnels

- binaires pour les gestionnaires de version (SCM, par exemple: `svn`), pour la consultation des dépôts (doivent être disponible dans votre PATH). Voir [RedmineRepositories](#) pour consulter les exigences et comptabilités des différents SCM.
- [RMagick](#) (pour activer l'export Gantt dans une image au format png)
- [La bibliothèque Ruby OpenID](#) (pour activer le support OpenID) ; seulement sur les versions de développement (trunk) de Redmine / 0.9-dev; la version 2 ou supérieure est obligatoire.

Installation

1. [Téléchargez \(en\)](#) et extrayez l'archive ou faites un [checkout \(en\)](#) de Redmine.

2. Depuis la version 1.4.0, Redmine utilise Bundler pour gérer les dépendances de *gems*. Vous devez installer Bundler en premier:

```
gem install bundler
```

Après vous pouvez installer tous les *gems* requis avec la commande suivante:

```
bundle install --without development test
```

Si ImageMagick n'est pas installé sur votre système, vous pouvez retirer l'installation du *gem rmagick* avec cette commande:

```
bundle install --without development test rmagick
```

Note concernant l'installation de *rmagick* sur Windows:

Au moment d'écrire ces lignes, il se peut que l'installation du *gem rmagick* fonctionne mal avec la commande `bundle install` si elle n'est pas déjà installée. Vous pouvez trouver de l'aide [ici](#)

Vous pouvez aussi retirer l'installation des adaptateurs que vous n'utilisez pas. Par exemple si vous utilisez MySQL, vous pouvez retirer les *gems* `postgresql` et `sqlite` avec la commande ``bundle install --without development test postgresql sqlite``.

Si vous avez besoin de *gems* non requis par le noyau Redmine (eg. `puma`, `fcgi`), créez un fichier nommé `Gemfile.local` à la racine de votre répertoire de Redmine et il sera chargé automatiquement au lancement de la commande ``bundle install``. Exemple:

```
# Gemfile.local
gem 'puma'
```

3. Créez une base de données vide nommée par exemple `redmine`.

Pour MySQL:

```
create database redmine character set utf8;  
create user 'redmine'@'localhost' identified by 'my_password';  
grant all privileges on redmine.* to 'redmine'@'localhost';
```

Pour les versions de MySQL antérieures à la 5.0.2 - passez l'étape 'create user' et optez plutôt pour cette commande :

```
grant all privileges on redmine.* to 'redmine'@'localhost' identified by 'my_password';
```

Pour PostgreSQL:

```
CREATE ROLE redmine LOGIN ENCRYPTED PASSWORD 'my_password' NOINHERIT VALID UNTIL 'infinity';  
CREATE DATABASE redmine WITH ENCODING='UTF8' OWNER=redmine;
```

4. Copiez config/database.yml.example dans config/database.yml et éditez ce fichier pour configurer vos paramètres de base dans l'environnement "production".

Exemple pour une base de données MySQL:

```
production:  
  adapter: mysql  
  database: redmine  
  host: localhost  
  username: redmine  
  password: my_password
```

Si votre serveur ne fonctionne pas sur le port standard (3306), utiliser plutôt cette configuration:

```
production:  
  adapter: mysql  
  database: redmine  
  host: localhost  
  port: 3307  
  username: redmine  
  password: my_password
```

Exemple pour la base de données PostgreSQL (port par défaut):

```
production:  
  adapter: postgresql  
  database: <le nom de votre base de données>  
  host: <nom de la machine de la DB>  
  username: <identifiant postgres>  
  password: <mot de passe postgres>  
  encoding: utf8  
  schema_search_path: <schéma de la base de données> (par défaut - public)
```

5. Générer un secret de session.

- Avec Redmine 1.4.x:

```
rake generate_session_store
```

- Avec Redmine 2.x:

```
rake generate_secret_token
```

6. Créez la structure de base en lançant la commande suivante depuis le répertoire racine de l'application:

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

Ceci va créer les tables et le compte administrateur.

Si vous obtenez cette erreur:

```
Rake aborted!  
no such file to load -- net/https
```

Vous devez installer libopenssl-ruby1.8 dans Ubuntu avec cette commande: apt-get install libopenssl-ruby1.8

7. Insérez les données de la configuration par défaut dans la base en lançant la commande suivante:

```
RAILS_ENV=production rake redmine:load_default_data
```

Cette étape est optionnelle, mais **fortement recommandée**, sauf si vous définissez votre nouvelle configuration. Elle va charger les rôles par défaut, les trackers, statuts, workflows et énumérations.

8. Paramétrez les permissions

NB: *Les utilisateurs sous Windows doivent sauter cette section.*

L'utilisateur qui exécute Redmine doit avoir les permissions en écriture sur les sous-répertoires suivants: files, log, tmp (créer ce dernier s'il n'est pas présent).

A supposer que vous fassiez fonctionner Redmine avec un utilisateur redmine:

```
mkdir tmp public/plugin_assets  
sudo chown -R redmine:redmine files log tmp public/plugin_assets  
sudo chmod -R 755 files log tmp public/plugin_assets
```

9. Testez l'installation en exécutant le serveur web WEBrick:

- Avec Redmine 1.4.x:

```
ruby script/server webrick -e production
```

- Avec Redmine 2.x:

```
ruby script/rails server webrick -e production
```

Une fois WEBrick démarré, saisir l'URL dans votre navigateur <http://localhost:3000/>. Vous devriez voir apparaître la page de bienvenue de Redmine.

10. Utilisez le compte administrateur par défaut pour vous connecter:

- identifiant: admin
- mot de passe: admin

Vous pouvez aller sur Administration pour modifier la configuration de l'application.

Configuration

Depuis la version 1.2.0, les configurations de Redmine sont définies dans le fichier `config/configuration.yml`.

Copier le fichier `config/configuration.yml.example` en `config/configuration.yml` et éditer le fichier pour adapter les paramètres SMTP et surcharger les configuration par défaut; le fichier est bien documenté par lui-même et devrait vous guider facilement.

Les configurations peuvent être définies par environnements *Rails*(production/development/test).

Ne pas oublier de redémarrer l'application après avoir effectué les modifications

Configuration du serveur SMTP

La configuration du SMTP fait l'objet d'un article distinct. Consultez les [exemples de configuration de mail](#).

Sauvegarde

Une sauvegarde Redmine inclue les éléments suivants:

- Les données (stockées dans votre base de données Redmine)
- Les fichiers (Stockés dans le répertoire files de votre installation de Redmine)

Voici un simple script shell que vous pouvez utiliser quotidiennement pour sauvegarder vos données (en supposant que vous utilisez une base de données mysql)

```
# Base de données
/usr/bin/mysqldump -u <username> -p<password> <redmine_database> | gzip >
/path/to/backup/db/redmine_`date +%y_%m_%d`.gz
```

```
# Fichiers
```

```
rsync -a /path/to/redmine/files /path/to/backup/files
```

Pour Windows, Robocopy fonctionne efficacement pour les fichiers avec sa fonctionnalité miroir.

```
Robocopy /path/to/redmine/files /path/to/backup/files /MIR
```

Notes à propos de l'installation sur Windows

Récupérer et installer rubyinstaller depuis le site <http://rubyforge.org>. Depuis le menu démarrer, exécuter la console cmd

Depuis la console les commandes suivantes :

```
RAILS_ENV=production rake db:migrate  
RAILS_ENV=production rake redmine:load_default_data
```

Doivent être changées par les commandes suivantes :

```
set RAILS_ENV=production  
rake db:migrate  
rake redmine:load_default_data
```

Vous pouvez avoir besoin d'installer mysql gem, avec la commande

```
gem install mysql
```

Et dans certains cas, il peut être nécessaire de copier le fichier libmysql.dll file dans votre répertoire ruby/bin.

Toutes les librairies ne semblent pas fonctionner, récupérer sur

<http://instantrails.rubyforge.org/svn/trunk/InstantRails-win/InstantRails/mysql/bin/libmySQL.dll>

Les Problèmes avec MySQL 5.1 et Rails 2.2+

Avec la dernière version de MySQL, vous pouvez rencontrer plusieurs erreurs, telles que :
"!!! The bundled mysql.rb driver has been removed from Ruby 2.2" et quand vous effectuez la commande "gem install mysql" vous pouvez aussi rencontrer plusieurs erreurs. Une solution possible :

Télécharger le fichier DLL depuis le projet InstantRails:

<http://instantrails.rubyforge.org/svn/trunk/InstantRails-win/InstantRails/mysql/bin/libmySQL.dll>

Copier ce fichier dans le répertoire \Ruby\bin directory et essayer d'effectuer une nouvelle fois votre commande rake.