

Joruri Maps 1.0.0 インストールマニュアル

2014. 5. 23

内容

Joruri Maps 1.0.0 インストールマニュアル	1
1 想定環境.....	2
2 CentOS のインストール	2
3 事前準備.....	2
4 Apache の設定	4
5 MapServer のインストール.....	4
6 PostgreSQL の設定	4
7 Ruby on Rails のインストール.....	5
7.1 Ruby1.9.3 のインストール	5
7.2 Phusion Passenger のインストール	5
7.3 gem ライブラリのインストール	6
8 Joruri Maps のインストール.....	7
8.1 環境設定	7
8.2 Mapfile の設定	7
8.3 データベースの作成.....	8
8.4 memcached のインストール	9
8.5 VirtualHost の反映	10
9 Joruri Maps のインストール (ふりがな・読み上げ機能)	11
10 定期実行設定	13

1 想定環境

[システム]

OS : CentOS 6.4

Web サーバ : Apache 2.2

DB システム : PostgreSQL 9

Ruby : 1.9.3

Rails : 3.2.13

[設定]

IP アドレス : 192.168.0.2

ドメイン : 192.168.0.2

2 CentOS のインストール

CentOS をインストールします。

インストール完了後、ご利用の環境に合わせて適切なセキュリティ設定を行ってください。

CentOS に関するセキュリティ設定については、本マニュアルの範囲外となります。

3 事前準備

root ユーザに変更します。

```
$ su -
```

Joruri Maps の実行に必要なパッケージをインストールします。

```
# yum -y install wget make gcc-c++ httpd httpd-devel ¥  
mysql-server mysql-devel libjpeg-devel libpng-devel ¥  
ImageMagick ImageMagick-devel curl-devel ¥
```

```
libevent libevent-devel openldap-servers openldap-clients ¥
openldap-devel cario cairo-devel openssl-devel
```

必要なパッケージをダウンロードします。

```
# cd /usr/local/src/
# wget -O jorurimaps-1.0.0.tar.gz ¥
  https://github.com/joruri/jorurimaps/archive/1.0.0.tar.gz
# wget ftp://ftp.ruby-lang.org/pub/ruby/1.9/ruby-1.9.3-p545.tar.gz
# wget http://memcached.googlecode.com/files/memcached-1.4.13.tar.gz
# wget http://pyyaml.org/download/libyaml/yaml-0.1.4.tar.gz
# wget http://chasen.org/~taku/software/darts/src/darts-0.32.tar.gz
# wget "http://sourceforge.jp/frs/redirect.php?m=jaist&f=%2Fchasen-
legacy%2F32224%2Fchasen-2.4.4.tar.gz"
# wget "http://sourceforge.jp/frs/redirect.php?¥
m=ij&f=%2Fipadic%2F24435%2Fipadic-2.7.0.tar.gz"
# wget http://downloads.sourceforge.net/lame/lame-398-2.tar.gz
```

Joruri ユーザを作成します。

```
# useradd joruri
# passwd joruri
# 新しいパスワード: <= 任意のパスワードを入力してください
```

Joruri Maps ソースコードを設置します。

```
# mkdir /var/share
# tar xvfz jorurimaps-1.0.0.tar.gz -C /var/share
# cd /var/share/jorurimaps
# cp config/original/* config/
# cp template/original/* template/
# chown -R joruri:joruri /var/share/jorurimaps
```

4 Apache の設定

Apache の設定ファイルを編集します。

```
# vi /etc/httpd/conf/httpd.conf
-----
#実際のサーバのアドレスを設定します
ServerName 192.168.0.2
-----
```

設定ファイルにエラーがないことを確認し、Apache を起動します。

```
# /sbin/service httpd configtest
# /sbin/service httpd start
```

自動起動に設定します。

```
# /sbin/chkconfig httpd on
```

5 MapServer のインストール

Mapserver_INSTALL.pdf を参考にしてください。

6 PostgreSQL の設定

```
# which psql
# ls /etc/init.d/
# chkconfig postgresql-9.1 on
```

joruri ユーザを作成する

```
# su - postgres
$ createuser joruri -P
```

ユーザのロール及びパスワードの入力プロンプトが表示されるので、画面の表示に従って入力してください。

7 Ruby on Rails のインストール

```
# cd /usr/local/src
# tar -zxvf yaml-0.1.4.tar.gz
# cd yaml-0.1.4
# ./configure
# make && make install
```

7.1 Ruby1.9.3 のインストール

Ruby をインストールします。

```
# cd /usr/local/src
# tar xvzf ruby-1.9.3-p545.tar.gz
# cd ruby-1.9.3-p545
# ./configure --prefix=/usr/local
# make && make install
```

インストール結果を確認します。

```
# /usr/local/bin/ruby -v
=> ruby 1.9.3p545 (2014-02-24 revision 45159) [i686-linux]
```

7.2 Phusion Passenger のインストール

Phusion Passenger をインストールします。

```
# gem install passenger -v 3.0.13
# passenger-install-apache2-module
```

(画面の内容を確認して Enter キーを押してください。)

Phusion Passenger の動作環境を設定します。

```
# vi /etc/httpd/conf/httpd.conf
-----
#下記を追加
LoadModule passenger_module
/usr/local/lib/ruby/gems/1.9.1/gems/passenger-
3.0.13/ext/apache2/mod_passenger.so
PassengerRoot /usr/local/lib/ruby/gems/1.9.1/gems/passenger-3.0.13
PassengerRuby /usr/local/bin/ruby
-----
```

7.3 gem ライブラリのインストール

Joruri Maps 実行に必要なライブラリをインストールします。

```
# gem install bundler -v 1.1.4
# cd /var/share/jorurimaps
```

通常とは違う場所に PostgreSQL をインストールしている場合、変数を渡しておく必要があります。

※MapServer のインストールで「Mapserver_INSTALL.pdf」の手順どおりにインストールした場合は、以下のオプションを付与して実行してください。

```
# bundle config build.pg --with-pg-config=/usr/pgsql-9.1/bin/pg_config
# bundle install
```

ネットワーク経由でファイルのダウンロードとインストールが行われます。正しく完了すれば次のようなメッセージが表示されます。

=>Your bundle is complete! Use `bundle show [gemname]` to see where a bundled gem is installed.

8 Joruri Maps のインストール

8.1 環境設定

環境に応じて設定ファイルを編集します。
設定ファイル作成

DB 接続情報

```
$ vi config/database.yml
```

LDAP 接続情報

```
$ vi config/ldap.yml
```

※LDAP に関する設定は、Joruri Gw と同様です。ダウンロードページより補足資料をご参照下さい。

<http://joruri.org/download/jorurigw/> 【LDAP 関連補足資料】
LDAP_LDIF.zip

SMTP 接続情報

```
$ vi config/environments/development.rb
```

※ production 環境の場合は `$ vi config/environments/production.rb`

アプリケーションの設定

```
$ vi config/application.yml
```

8.2 Mapfile の設定

Mapfile は MapServer の設定ファイルです。地図情報をレンダリングする際に参照されます。

DB 接続設定が含まれるので、環境に応じて編集を行ってください。

```
$ vi template/line.tpl
```

```
$ vi template/point.tpl
$ vi template/point_data_layer.tpl
$ vi template/polygon.tpl
```

各ファイルの以下の部分を設定に応じて編集する。

```
-----
    CONNECTION "user=joruri dbname=joruri_maps host=127.0.0.1"
-----
```

8.3 データベースの作成

データベースを作成します。

Joruri Maps の DB を作成する

PostGIS テンプレート用 DB を作成

```
$ su - postgres
$ createdb template_postgis -E UTF-8
$ createlang plpgsql template_postgis
```

テンプレート用 DB をコピーして Joruri Maps 用の DB を作成

```
$ createdb -T template_postgis joruri_maps
ユーザを作成
$ createuser joruri
```

joruri ユーザから PostgreSQL の操作ができるよう設定を変更します。

```
# vi /var/lib/pgsql/9.1/data/pg_hba.conf
```

```
-----
# TYPE  DATABASE      USER      ADDRESS      METHOD

# "local" is for Unix domain socket connections only
#local  all        all        peer
local  all all trust
```



```
# IPv4 local connections:
```

```
host    all             all             127.0.0.1/32     trust
host    all             all             192.168.16.0/24  trust
-----
```

PostgreSQL を再起動し、設定を適用します。

```
# /etc/init.d/postgresql-9.1 restart
```

テーブルを作成します。

```
$ rake db:schema:load RAILS_ENV=production
```

初期データを登録します。

```
$ rake db:seed RAILS_ENV=production
```

サンプルデータを登録します。

```
$ rake db:seed:demo RAILS_ENV=production
```

8.4 memcached のインストール

スーパーユーザーに戻ります。

```
$ su -
```

memcached をインストールします。

```
# cd /usr/local/src/
# tar xvzf memcached-1.4.13.tar.gz
# cd memcached-1.4.13
# ./configure
# make && make install
```

自動起動に設定します。

```
# cp /usr/local/src/memcached-1.4.13/scripts/memcached.sysv
/etc/rc.d/init.d/memcached
# vi /etc/rc.d/init.d/memcached
-----
#16 行目に下記を追加
PATH=$PATH:/usr/local/bin ; export PATH
-----
# mkdir /var/run/memcached
# /sbin/chkconfig --add memcached
# chkconfig --level 35 memcached on
```

memcached を起動します。

```
# /sbin/service memcached start
```

8.5 VirtualHost の反映

Apache に設定を追加します。

```
# vi /etc/httpd/conf.d/vhosts.conf
-----
NameVirtualHost *:80

<VirtualHost *:80>
    ServerName default
    DocumentRoot /var/share/jorurimaps/public
    RackEnv production
    PassengerHighPerformance on
</VirtualHost>
-----
```

※Rails の実行環境が production の場合、RackEnv は下記のようにしてください。

RackEnv production

Apache を再起動します。

```
# /sbin/service httpd configtest  
# /sbin/service httpd restart
```

以上で Joruri Maps のインストールが完了しました。

9 Joruri Maps のインストール (ふりがな・読み上げ機能)

スーパーユーザーに戻ります。

```
$ su -
```

必要なパッケージをインストールします

```
# yum -y install nkf libxslt libxslt-devel
```

LAME をインストールします。

```
# cd /usr/local/src  
# tar xvf lame-3.98.2.tar.gz  
# cd lame-3.98.2  
# ./configure --prefix=/usr  
# make && make install
```

Darts をインストールします。

```
# cd /usr/local/src  
# tar xvf darts-0.32.tar.gz  
# cd darts-0.32  
# ./configure --prefix=/usr  
# make && make install
```

ChaSen をインストールします。

```
# cd /usr/local/src
# tar xvzf chasen-2.4.4.tar.gz
# cd chasen-2.4.4
# ./configure --prefix=/usr
# make && make install
```

IPAdic をインストールします。

```
# cd /usr/local/src
# tar xvzf ipadic-2.7.0.tar.gz
# cd ipadic-2.7.0
# ./configure --prefix=/usr
```

辞書を UTF8 に変換します。

```
# vi to_utf8.sh  #(新規作成)
-----

#!/bin/sh
for file in *.dic *.cha chasenrc
do
if [ -f $file ]; then
    nkf --utf8 $file > tmpfile
    mv tmpfile $file
fi
done
exit
-----

# chmod 744 to_utf8.sh
# ./to_utf8.sh
# `chasen-config --mkchadic`/makemat -i w
# `chasen-config --mkchadic`/makeda -i w chadic *.dic
# make install
```

ChaOne をインストールします。

```
# cd /var/share/jorurimaps/ext/morph/chaone
# chmod 775 configure
# ./configure
# make && make install
```

GalateaTalk をインストールします。

```
# cd /var/share/jorurimaps/ext/gtalk
# sed -i "s/ getline/ my_getline/" main.c
# sed -i "s/ getline/ my_getline/" getline.c
# chmod 775 configure
# ./configure
# make
# chmod 747 /var/share/jorurimaps/ext/gtalk_filter.rb
```

10 定期実行設定

```
# su - joruri
$ crontab -e
-----
*/15 * * * * /bin/bash -l -c 'cd /var/share/jorurimaps/ && /usr/local/bin/ruby
script/rails runner Misc::Script::Feed.import_rss -e production'
-----
```