

Regional SNS 開発環境構築ガイド

2012年2月29日

株式会社ネットワーク応用通信研究所

目次

1. はじめに	1
2. 前提条件	1
3. 必要なソフトウェア構成	1
4. ソフトウェアの導入手順	1
4.1. 必要ファイルのダウンロード	1
4.2. 環境設定コマンドの実行	2
4.3. RegionalSNS の実行	2
4.4. MOGOK CLI パッケージの導入 (必要に応じて)	2
5. ソースコードの修正	3
5.1. Git によるソースコード管理	3
5.2. 機能追加方法	4
5.2.1. 拡張用プラグインの作成	4
5.2.2. プラグインの構造	4
5.2.3. 拡張メニューへの反映	4
5.2.4. 機能の拡張	5

1. はじめに

本ドキュメントでは、RegionalSNS（以下、本システムと呼称）のソースコードに独自の改修を加え、オリジナルの機能追加を行うための開発環境を構築する手順を説明します。

2. 前提条件

本ドキュメントでは、OS は Windows Xp（32bit 版）のみを想定します。

3. 必要なソフトウェア構成

本システムの開発を行うには、以下のソフトウェアがローカル PC に導入されている必要があります。

ソフトウェア名	バージョン	備考
Ruby	1.9.2-p290 以上	MinGW 版を推奨
Bundler	1.0 以上	RubyGems で導入
mogok CLI	最新バージョン	MOGOK で運用する場合に必要。RubyGems で導入
MySQL	5.0 以上	
Subversion クライアント	任意のバージョン	ソースコードの管理用
ImageMagick	6.7.6 以上	依存ソフトウェアとして導入

※ Ruby on Rails の実行環境は Bundler 経由で必要なバージョンを一括導入します。

4. ソフトウェアの導入手順

前項に掲げるソフトウェアを、ローカル PC に導入します。

4.1. 必要ファイルのダウンロード

「開発環境パッケージ」をダウンロードし、任意のディレクトリに展開してください。
本ドキュメントでは、「c:\¥regionalsns_dev」に展開されたものとして説明します。

4.2. 環境設定コマンドの実行

「c:\¥regionalsns_dev¥setup.bat」を実行し、環境設定を行なってください。

以下の作業が行われます。

1. MySQL Server 5.5.15 のサービス登録とサービス起動
2. ユーザー環境変数に Ruby のバイナリディレクトリパスを設定
3. RegionalSNS のソースコードのエクスポート
4. ImageMagick のインストーラーの起動

ImageMagick のインストーラーについては、インストールウィザードの指示に従ってインストールを実施してください。

※ 特に理由が無ければ、全てデフォルトの設定で問題ありません。

4.3. RegionalSNS の実行

導入した RegionalSNS が稼働できることを以下の手順で確認します。

1. c:\¥regionalsns_dev¥msysgit¥ git-cmd.bat を実行し、コマンドプロンプトを表示
2. c:\¥regionalsns_dev¥sns_src¥config/database.yml.mysql を リネームし「database.yml」に変更
3. 以下のコマンドを実行し、初期環境設定を行う

```
> cd c:\¥regionalsns_dev¥sns_src
> rake db:create:all
> rake db:migrate
> rake db:seed
```
4. 以下のコマンドでサーバを起動し、<http://localhost:3000/> に接続して動作確認を行う

```
> rails server
```

4.4. MOGOK CLI パッケージの導入（必要に応じて）

MOGOK ヘデプロイを行う場合は、MOGOK アカウントで以下の URL に接続し、「mogok-x.x.x.gem」（x.x.x 部分はバージョン番号）をダウンロードします。

<https://portal.mogok.jp/download>

ダウンロードした最新の MOGOK の gem パッケージを、コマンドプロンプトから以下のようにして Ruby 実行環境にインストールしてください。

```
➤ gem install mogok-x.x.x.gem
```

5. ソースコードの修正

必要に応じて、RegionalSNS のソースコードを修正し、目的に応じたコードを追加・修正します。

エクスポートされたソースコードは、デフォルトでは `c:\¥regionalsns_dev¥sns_src` ディレクトリに保存されています。

このまま開発を開始できる状態になっていますが、何らかのバージョン管理システムを導入し、バージョン管理することを強く推奨します。

運用先として MOGOK を想定する場合は、`git` による管理を行うと良いでしょう。

5.1. Git によるソースコード管理

オープンソースソフトウェアとして本システムの機能追加を行うのであれば、Github などインターネット上の Git リポジトリサービスを利用することが良いと思われませんが、クラウドな環境で開発を行う場合は、何らかの方法で Git リポジトリを用意せねばなりません。最初はローカル PC において Git リポジトリを作って開発し、ある程度規模が大きくなってからリモートサーバ上に Git リポジトリを作ってそちらに統合する方法を取ることも可能です。

ローカルの Git リポジトリを作成する場合は、エクスポートしたディレクトリで以下のコマンドを実行します (`git-cmd.bat` を利用してください)。

- `git init`
- `git add .`
- `git commit -m "first commit"`

Git の詳しい利用方法などについては、以下をご参照ください。

- Git 公式サイト
<http://git-scm.com/>
- Git 入門 (分かりやすくまとめられた Git の使い方紹介サイト)
http://www8.atwiki.jp/git_jp/

5.2. 機能追加方法

本システムに機能を追加する際の一般的な手順をご紹介します。

5.2.1. 拡張用プラグインの作成

RegionalSNS では、機能拡張はプラグイン形式で行います。

プラグイン内に `app` ディレクトリがあり、MVC 全ての部品を備えた、一種の Rails アプリケーションの形でプラグインを作り、SNS に新しい機能を追加します。

まずは、以下のコマンドを `RAILS_ROOT` で実行して、プラグインの雛型を作ります。

```
➤ rails generate ext_plugin プラグイン名
```

今回、プラグイン名として「`my_plugin`」を指定したものと説明します。

5.2.2. プラグインの構造

作成したプラグインは、Rails の標準的なプラグインと同じディレクトリ構造を採って、`RAILS_ROOT/vendor/plugins` 配下に作成されます。

ディレクトリツリーは以下のようになります。

`my_plugin`

```
├─ app
  │   └─ controllers
  │   └─ helpers
  │   └─ models
  │   └─ views
├─ config
├─ lib
└─ test
```

その他、プラグインディレクトリ直下に `Rakefile` や `README` など、通常のプラグインと同じファイルが配置されます。

5.2.3. 拡張メニューへの反映

生成されたプラグインは、プラグイン名をリンク名として、SNS の「拡張メニュー」に自動的にリンクされます。

このリンクは、プラグイン側の「`config/setting.yml`」の内容に基づいて生成されます。

リンク名や、リンク先 URL を変更したい場合は、このファイルを編集してください。

5.2.4. 機能の拡張

プラグインディレクトリの配下は、通常の Rails アプリケーションと同じようにコントローラやモデル、ビューなどを配置できます。

ルーティングの制御も、プラグインディレクトリ配下の `config/routes.rb` に、通常の Rails アプリケーションと同じ方法で追加すれば、自動的に SNS のルーティングの一部として追加されます。

例えば、`my_plugin` に `pages` という名前のコントローラを追加し、画面に「**FOO**」と表示するようプラグインを構成してみましょう。

以下の手順で作業してください。

※ 作業は全て「`vendor/plugins/my_plugin`」配下で行います。

1. `app/controller/foo_controller.rb` を以下のように編集する

```
class PagesController < ApplicationController
  def index
    render text: "FOO"
  end
end
```

2. `config/routes.rb` を以下のように編集する

```
NinjinSns::Application.routes.draw do
  resources :pages
end
```

3. `config/setting.yml` を以下のように編集する

```
plugin:
  name: "MyPlugin"
  menu:
    login_require: true
    title: "my_plugin"
    url_method: "pages_path"
```

4. Rails サーバを再起動する

以上で、拡張メニューの「`my_plugin`」リンクをクリックして「**FOO**」という画面に遷移するようになります。