

iPhone Push Notification with im.kayac.com

五十嵐邦明

<https://github.com/igaiga/>

twitter : igaiga555

<http://www.facebook.com/kuniaki.igarashi>

2013.10.27

iPhoneに
Push通知
する仕組みを
作って
みましょう。

20:47

1月5日(土曜日)

im.kayac.com

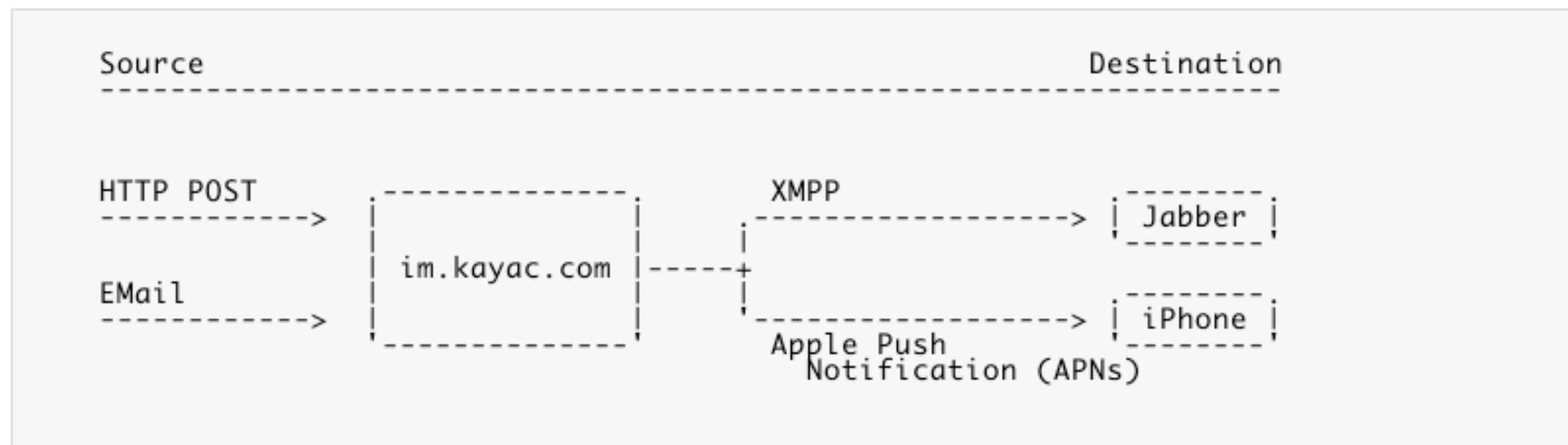


1月5日の天気:曇り 気温:°C/°C 東京地方では、空気の乾燥した状態が続きますので、火の取り扱いに注意してください。伊豆諸島北...



表示

Welcome to Notification HUB API



im.kayac.com は通知用のいくつかのAPIを提供するウェブサービスです。

HTTP POSTもしくはEメールを使用して、簡単にあなたの Jabber アカウントや、iPhoneに通知を送信することが可能です。

現在サポートしているAPIソースは：

**im.kayac.comを使うと簡単にPush通知する
仕組み一式を作ることができます。**

通知先として：

- Jabber, Google Talk

<http://im.kayac.com/>

iPhonePush通知に必要なもの

iPhoneアプリ

Push通知を受ける

送信サーバー

ApplePush通知網へ送信する

送信コード

送信サーバーAPIを叩く

iPhonePush通知に必要なもの

iPhoneアプリ
Push通知を受ける

im kayac が用意

送信サーバー
ApplePush通知網へ送信する

im kayac が用意

送信コード
送信サーバーAPIを叩く

**ここだけ
書けばOK**

事前準備

1. im.kayac.com でアカウントを作成します。

2. im.kayac.com アカウント設定 - API認証のページからAPI用パスワードを設定します。

※これを設定しないと誰でもPush送信できる設定になります。

im.kayac.com

API 認証

認証

- ☐ 認証を使用しない。誰でもあなたにたいしてメッセージを送ることができます。
- ☒ パスワード認証を使用する
- ☐ 秘密鍵認証を使用する

パスワード

変更

事前準備

3. im.kayac.com のiPhoneアプリをインストールします。

無料版

<https://itunes.apple.com/jp/app/im.kayac.com-free-edition/id336129317>

有料版

<https://itunes.apple.com/jp/app/im.kayac.com-receive-your/id336122737>

**4. im.kayac.com のiPhoneアプリの設定で
im.kayac.comのアカウント/パスワードを設定して
おきます。**

im-kayac でPushするコード



im-kayac-post.rb

```
1  # $ gem i im-kayac
2  # http://im.kayac.com/
3  require 'im-kayac'
4  user = 'USER'
5  password = 'PASSWORD'
6  message = ARGV[0] ? ARGV[0].to_s : 'blank'
7  begin
8    p ImKayac.to(user).password(password).post(message)
9  rescue => e
10    STDERR.puts e
11  end
```


im-kayac でPushするコード解説

 im-kayac-post.rb

```
1  # $ gem i im-kayac  gemをインストールしておく
2  # http://im.kayac.com/
3  require 'im-kayac'
4  user = 'USER'          im-kayacのユーザー名と
5  password = 'PASSWORD' API用パスワード(アカウントのパスではなく)
6  message = ARGV[0] ? ARGV[0].to_s : 'blank'
7  begin  コマンドの引数をメッセージにします。空なら blank という文字列。
8    p ImKayac.to(user).password(password).post(message)
9  rescue => e  ↑もしうまく動かない場合は、im.kayac.comで
10    STDERR.puts e  API passwordの設定をOFFにして、
11  end  ImKayac.to(user).post(message)
      で動作確認してみてください。
```

<https://gist.github.com/4160992>

使い方

ターミナルを起動して以下を実行します。

gemをインストール

```
gem i im-kayac
```

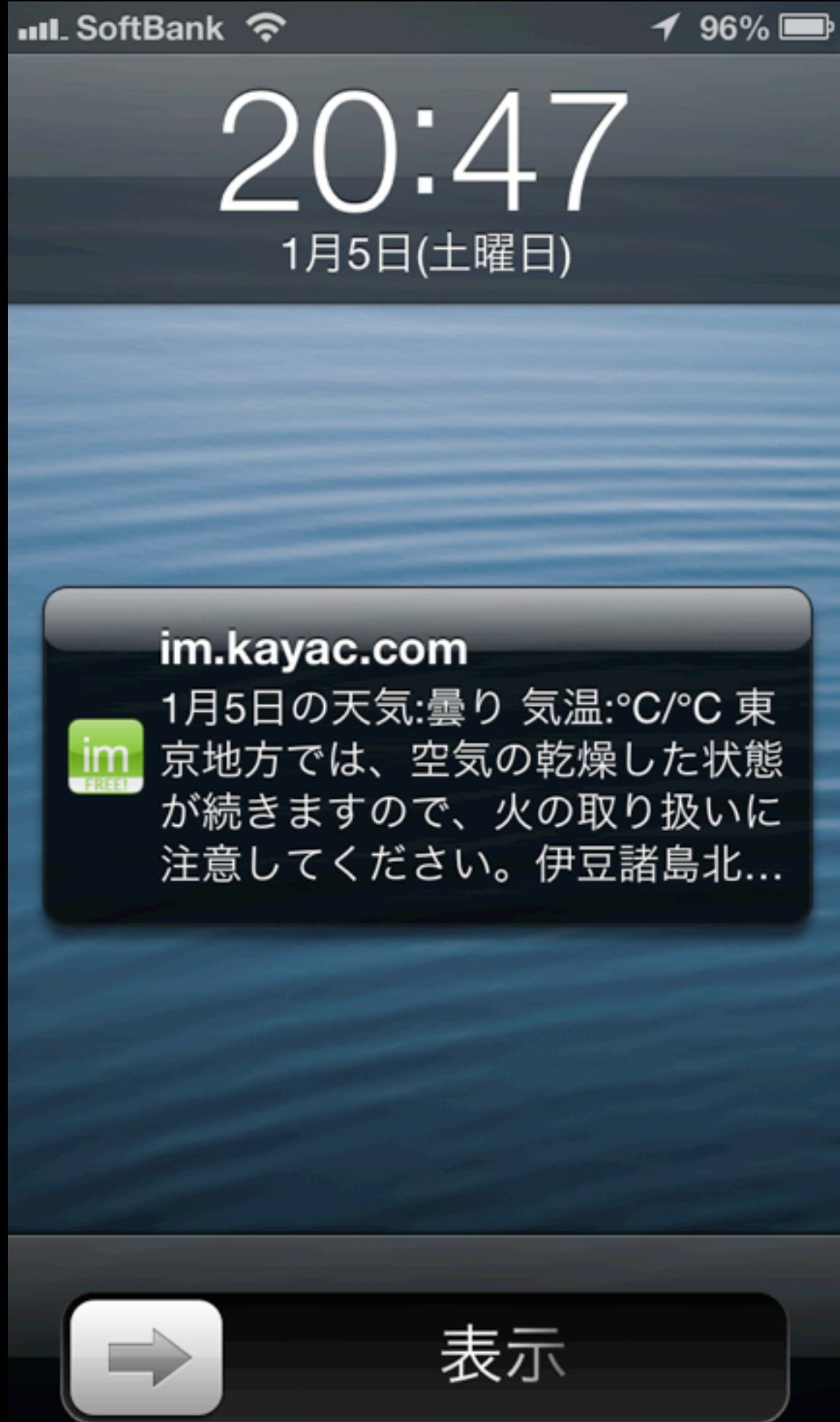
実行

```
ruby im-kayac-post.rb 送りたい文
```

つくってみた

冬休みの工作 で作ってみた

毎朝iPhone
Push通知で
天気予報をお知らせ



手作りのコードが
毎朝ちゃんと動い
て天気予報を教え
てくれるのがうれ
しい。

* . ° ° . * : : ° . (*'▽'*)パ ア . ° : : * . ° ° . *

SoftBank

96%

20:47

1月5日(土曜日)

im.kayac.com



1月5日の天気:曇り 気温:°C/°C 東
京地方では、空気の乾燥した状態
が続きますので、火の取り扱いに
注意してください。伊豆諸島北...



表示

コードは約40行

総制作期間約8時間

```
1  # -*- coding: utf-8 -*-
2  # using Livedoor Weather Web Service / LWWS
3  # http://weather.livedoor.com/forecast/websevice.html
4
5  require 'open-uri'
6  require 'nokogiri'
7  require 'date'
8  require 'awesome_print'
9  require 'pry'
10
11 # im-kayac
12 require 'im-kayac'
13 require 'pit'
14 config_im_kayac = Pit.get 'im-kayac'
15 user = config_im_kayac['user']
16 password = config_im_kayac['pass']
17
18 def parse(xml)
19   h = {}
20   h[:telop] = xml.xpath("//telop").children.first.to_s
21   h[:temperature_max] = xml.xpath("//temperature//max//celsius").children.first.to_s
22   h[:temperature_min] = xml.xpath("//temperature//min//celsius").children.first.to_s
23   h[:description] = xml.xpath("//description").children.first.to_s
24   h[:date] = Date.parse(xml.xpath("//forecastdate").children.first.to_s)
25   h[:city] = xml.xpath("//location").attr('city').value.to_s
26   h
27 end
28
29 def message(h)
30   "#{h[:date].month}月#{h[:date].day}日の天気:#{h[:telop]} 気温:#{h[:temperature_max]}℃/#{h[:temperature_min]}℃ #{h[:descri]
31 end
32
33 url = 'http://weather.livedoor.com/forecast/websevice/rest/v1?city=63&day=today'
34 # city = 63 : Tokyo
35 # 地区ID: http://weather.livedoor.com/forecast/rss/forecastmap.xml
36 xml = Nokogiri::XML(open(url).read)
37 # for sample.xml
38 # xml = File.open("sample.xml") { |f| Nokogiri::XML(f) }
39
40 h = parse xml
41 p ImKayac.post(user, message(h), {:password => password})
42
```

<https://github.com/igaiga/weather>

Contents

- chapter 1 「たのしい開発」を求めて
- chapter 2 Ruby の基礎知識
- chapter 3 Ruby を使ってみよう
- chapter 4 Ruby on Rails とは
- chapter 5 Rails を触ってみよう
- chapter 6 Ruby の文化
- chapter 7 Ruby の開発環境
- chapter 8 Ruby のアプリケーション
- chapter 9 Ruby のコミュニティ
- chapter 10 とある企業の Ruby 導入事例

Have a Happy Ruby Time!

たのしい開発
スタートアップ
Ruby

たのしい開発

スタートアップ

Ruby

大場寧子 監修
櫻井達生
大場光一郎
五十嵐邦明 著

これから Ruby を始める人、必読。

Ruby の良さがピンと来ない人、必読。

Ruby を使いたくない人、必読。

推薦!

これから Ruby はもっと楽しくなる。

講義資料置き場

五十嵐が作成したRubyとRailsの教科書です。
プログラミングが初めての学生向きに作りました。
<https://github.com/igaiga/hitotsubashi-ruby-2013>