

平成 24 年 12 月 12 日
一部改訂 平成 25 年 12 月 13 日
一部改訂 平成 29 年 3 月 2 日
気象庁

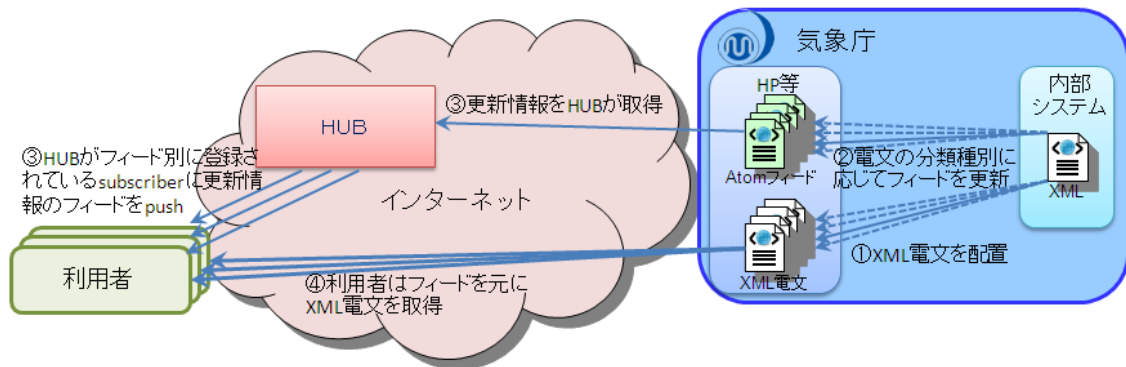
XML 電文公開 (PUSH 型) に係る仕様と Subscriber の構築について

○XML 電文公開 (PUSH 型) に係る機能構成

・当庁側機能の概要

気象庁では XML 電文の公開 (PUSH 型) にあたり、フィード情報と同情報の更新情報の提供を行うこととしました。これら情報の提供を介した XML 電文の公開に関する機能概要は以下のとおりです。

気象庁では、情報発表により生成された XML 電文を、気象庁の HP 上にユニークなファイル名にて公開すると同時に、情報の種別により分類した atom フィードに対して情報の更新を行います。フィードの更新が完了した後、PubSubHubbub の HUB に対して当該フィードの更新通知を行います。HUB では更新通知を受けて、当該 atom フィードの読み取りを行い、最新のフィードについて差分を抽出します。その後、subscriber として登録されている利用者のエンドポイント URL に対してフィードの差分を push します。利用者側はエンドポイント URL にて受領したフィードを元に、XML 電文の取得可否を判断して、必要に応じて気象庁の HP 上より XML 電文を取得します。



・HUB について

今回、フィード情報と同情報の更新情報の提供にあたっては、仕組みとしてオープンな規格である PubSubHubbub プロトコルを用いています。PubSubHubbub プロトコルについては、以下の URL に説明があります。

<http://code.google.com/p/pubsubhubbub/>

PubSubHubbub においては、publisher と subscriber を繋ぐための HUB が必要となります。HUB については、オープンな HUB がいくつか存在しますが、気象庁では Google.org

による Alert Hub を利用します。Alert Hub については、以下の URL に説明があります。

<http://alert-hub.appspot.com/>

ユーザー数の把握等を行うため、Alert Hub への subscriber の登録は気象庁にて実施します。ユーザー登録の申請方法等は XML 電文公開 (PUSH 型) の HP をご確認ください。

・気象庁から送信する atom フィードについて

HUB から push されるフィード及びエントリーの詳細については後述のとおりです。これらの情報を元に、利用する情報の選択や XML 電文の取得ソフトウェアの作成を行います。なお、気象庁から送信する atom フィードの URL は「<http://xml.kishou.go.jp/>」で始まります。

・利用者において準備するもの

XML 電文の公開 (PUSH 型) にあたって、利用者においては以下の準備が必要になります。

- ・ HUB からの更新情報を受信できるアプリケーションサーバー (HTTP サーバー)
- ・ フィード記載の URL から XML 電文を取得できるサーバー (HTTP クライアント)
- ・ subscriber 及び必要な関連機能の実装
- ・ 各サーバーのインターネット接続環境
- ・ その他、利用にあたって必要な手続き

サーバーに関する要件は特段無く、いわゆるクラウドでも自宅 PC でも構いませんが、原則として 24 時間 365 日の安定稼働を条件とします。動作の不安定により更新情報の配信が止まってしまうことが度々ある場合、HUB の登録そのものを保留とさせていただく場合があります。

○XML 電文公開 (PUSH 型) に係る機能制限等

XML 電文の公開 (PUSH 型) にあたっては、各種調査や実装の都合上、以下のような制限がありますので、ご了承願います。

- ・ 負荷の抑制のため、atom フィードに対する直接アクセスはできないようにしています。
- ・ Alert Hub への subscriber 登録は気象庁にて実施します。
- ・ 公開サーバー上での XML 電文の保存期間は 24 時間です。
- ・ その他、利用上の注意事項に留意願います。特に情報更新の停止や遅延が発生する場合がありますこと、XML 電文に独自の予報を付加する等して公開する際には予報業務許可が必要となることに留意願います。

○Subscriber の構築概要

利用者で準備する subscriber については、その実装見本が以下の URL にあります。

<http://code.google.com/p/pubsubhubbub/wiki/SubscriberClients>

subscriber には PubSubHubbub プロトコルに従い HUB からのフィード等を受け取る機能のほか、

- ・フィードの内容から情報の取得可否について判定する機能。
- ・判定の結果に応じて、フィードに記載の URL から XML 電文を取得する機能。
- ・取得した XML 電文を永続化や他の処理へ引き渡す等、次の処理に引き継ぐ機能。

等が必要になります。

一通り完成しましたら動作確認をお願いします。動作確認の方法として、別紙 1 に例を挙げます。

安定稼働が可能となりましたら、ユーザー登録のページから登録様式をダウンロードして必要事項を記入の上、気象庁宛にユーザー登録の申請をお願いします。

○その他

過剰な負荷・アクセス等の異常があった場合には公開を一時休止する場合があります。

また、subscriber の実装や搭載にあたって、PubSubHubbub プロトコルやその実プログラムに係る質問についてはお受けできませんので、ご了承ください。

○atom フィード詳細仕様

気象庁が作成し、Alert Hub 経由で push されるフィードの詳細は以下のとおりです。なお、本仕様については、今後変更される場合があります。

要素名	属性名	説明
feed		atom フィードのルート要素
(feed 子要素)		
title		"JMAXML publishing feed" (固定値)
subtitle		"this feed is published by JMA" (固定値)
updated		フィード作成時刻を記述する
id		フィード URL の UUID 値 (version 3) を記述する
link	rel	"related" (固定値)
	href	"http://www.jma.go.jp/" (固定値)
link	rel	"self" (固定値)
	href	"http://xml.kishou.go.jp/***/**/*.xml"等 フィード URL を記述する
rights		"Published by Japan Meteorological Agency" (固定値)
entry		1つのフィードに1つ、もしくは複数、存在する
(entry 子要素)		
title		XML 電文管理部の情報名称を記述する
link	type	"application/xml" (固定値)
	href	XML 電文ファイルの URL を記述する
id		XML 電文ファイルの UUID 値 (version 3) を記述する
updated		XML 電文管理部の発表時刻値を記述する
author		XML 電文管理部の発表官署名値を記述する
content		XML 電文ヘッダ部の標題値、見出し文値を記述する
	type	"text" (固定値)
(author 子要素)		
name		XML 電文管理部の発表官署名値を記述する。

atom フィードの例 :

```
<?xml version="1.0" encoding="utf-8"?>
<feed xmlns="http://www.w3.org/2005/Atom" xml:lang="ja">
<title>JMAXML publishing feed</title>
<subtitle>this feed is published by JMA</subtitle>
<updated>2012-08-15T07:01:01+09:00</updated>
<id>urn:uuid:be4342e2-ff73-363c-a3ed-66e05e977224</id>
<link href="http://www.jma.go.jp/" rel="related"/>
<link href="http://xml.kishou.go.jp/*/*.xml" rel="self"/>
<rights>Published by Japan Meteorological Agency</rights>

<entry>
<title>気象警報・注意報</title>
<id>urn:uuid:8e55b8d8-518b-3dc9-9156-7e87c001d7b5</id>
<updated>2012-08-15T07:00:00+09:00</updated>
<author><name>富山地方気象台</name></author>
<link href="http://*/*/8e55b8d8-518b-3dc9-9156-7e87c001d7b5.xml" type="application/xml" />
<content type="text">【富山県気象警報・注意報】富山県では、強風、高波に注意してください。
</content>
</entry>
<entry>
<title>気象警報・注意報</title>
<id>urn:uuid:b60694a6-d389-3194-a051-092ee9b2c474</id>
<updated>2012-08-15T07:00:00+09:00</updated>
<author><name>京都地方気象台</name></author>
<link href="http://*/*/b60694a6-d389-3194-a051-092ee9b2c474.xml" type="application/xml"
/>
<content type="text">【京都府気象警報・注意報】京都府では、15日昼過ぎから高波に注意し
て下さい</content>
</entry>
</feed>
```

subscriber 動作確認方法の例

どなたでも利用できる Google の HUB がありますので、こちらを用いた subscriber の動作確認方法の例を挙げます。なお、本方法はあくまで例ですので、この通りの対応がうまくいかなかった場合でも、気象庁では対応致しかねますので、ご了承ください。

以下の URL に Google の HUB があります。

<http://pubsubhubbub.appspot.com/>

1. subscriber を準備し、サーバーに配置します。サーバーは上記 HUB からアクセス可能である必要があります。
2. 任意にエントリーの追加が可能な atom feed をインターネット上に用意します。最も簡便な方法は、blog を利用することです。
3. テスト用 atom feed を受け取るための設定を行います。

以下の説明では、仮に、

- subscriber の URL

<http://subscriber-test.xxxxxxxx/subscriber.php>

- テスト用 atom feed の URL

<http://publisher-test.xxxxxxxx/atom.xml>

とします。

- HUB の subscriber 登録/解除用 URL にブラウザでアクセスします (図 1)

<http://pubsubhubbub.appspot.com/subscribe>

- 「Callback」欄に、subscriber の URL を入力します
- 「Topic」欄に、テスト用 atom feed の URL を入力します
- 「Mode」欄で、「Subscribe」を選択します
- その他の欄は、そのままとします
- 「Do it」ボタンを押すと、登録完了です (但し、画面は変化しません)

Subscribe/Unsubscribe

Create a new subscription.

Callback: *(the subscriber URL)*

Topic: *(the feed URL)*

Verify type:

Mode:

Verify token:

HMAC secret: *(optional)*

Do it

Note: submission will result in a HTTP 204 response to acknowledge; in browsers this looks like a no-op

(図 1)

4. atom feed にエントリーを追加し、publish を行います。

まず、blog に書き込みを行う等して、atom feed にエントリーを追加します。その後、以下の要領で publish を行います。

- HUB の publish 用 URL にブラウザでアクセスします (図 2)
http://pubsubhubbub.appspot.com/publish
- 「Topic」欄に、テスト用 atom feed の URL を入力します
- [Publish] ボタンを押すと、HUB へ更新通知されます (画面は変化しません)

Publish

Topic: *(the feed URL)*

Publish

Note: submission will result in a HTTP 204 response to acknowledge; in browsers this looks like a no-op

(図 2)

このとき、HUB から subscriber へフィードが配信されますので、正常に受信できたか確認します。(不可の場合は次項の「診断」で HUB 側の状況を確認してみてください)

これを幾度か繰り返し、subscriber がフィードを安定して受信できるか確認します。

5. HUB から見て Subscriber が正常に動作しているか、以下の要領で診断します。

- HUB の subscriber 登録/解除用 URL にブラウザでアクセスします (図 3)

`http://pubsubhubbub.appspot.com/subscribe`

(Subscriber Diagnostics のほうの)

- 「Callback」欄に、subscriber の URL を入力します
- 「Topic」欄に、テスト用 atom feed の URL を入力します
- 「Secret」欄は、空白のままとします
- 「Get info」ボタンを押すと、診断結果画面 (図 4) に遷移します

Subscriber Diagnostics

Retrieve details about a subscription.

Callback: *(the subscriber URL)*

Topic: *(the feed URL)*

Secret: *(optional)*

Get info

(図 3)

Subscription Details – `http://publisher-test.xxxxxxxx/atom.xml`

Callback URL: `http://subscriber-test.xxxxxxxx/subscribe.php`
Created time (UTC): `2012-08-20T08:00:20Z`
Last modified time (UTC): `2012-08-20T08:00:20Z`
Expiration time (UTC): `2012-08-25T08:00:19Z`
State: `verified`
Confirmation failures: `0`
Delivery to domain: `OK`
Delivery short-term: `0% errors`

(図 4 診断結果画面例)

図 4 にて以下の点を確認します。

- 「State: `verified`」となっているか
- 「Confirmation failures: `0`」となっているか
- 「Delivery to domain: `OK`」となっているか
- 「Delivery short-term: `0% errors`」となっているか

6. subscriber 側の受信状況を確認し、問題無ければ、気象庁宛に登録申請を出します。

7. テスト用の更新通知が不要となった場合は、以下のとおり「購読解除」を行います。

- HUB の subscriber 登録/解除用 URL にブラウザでアクセスします (図 1)

<http://pubsubhubbub.appspot.com/subscribe>

- 「Callback」欄に、subscriber の URL を入力します
- 「Topic」欄に、テスト用 atom feed の URL を入力します
- 「Mode」欄で、“Unsubscribe”を選択します
- その他の欄は、そのままとします
- [Do it] ボタンを押すと、登録完了です (但し、画面は変化しません)