気象庁XML取得API
7年間の運用のまとめ その2

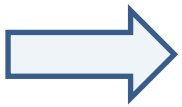
2020年10月1日

先端IT活用推進コンソーシアム
クラウド・テクノロジー活用部会　上村準也

7年間の運用のまとめ その2

REST API の中の人が、ふりかえりとネタばらし

1. 作り始めた頃に考えていたこと
2. JSON形式のデータはどうやって作っているの？
3. なぜ大雨危険度通知を利用できないの？
4. 運用を終える今に考えていること



■気象庁XML用API

クラウド・テクノロジー活用部会の活動の一環として、気象庁より試行配信中の気象庁XMLを受信し、簡単に参照できるAPIを作成してみました。

・蓄積データ参照＆REST API
・SPARQL クエリ発行
・WebSocketによる配信

APIの使い方はこちらを参照してください。

作り始めた頃に考えていたこと

- PostgreSQL でXML型が使えるのを試したい
  作り物でなく本物のデータを扱う方が面白い
  - 後にJSONも扱うようになるのも同じ理由

- 実際にデータを活用したサービスを作るのに
  生データをブラウズできるものがあれば捗る

JSON形式のデータの作り方

- http://api.aitc.jp/jmardb-api/help
  ここにある記載のものです

検索結果に含まれる link で、気象庁防災情報 XML の元データを参照できます。
URL の末尾に ".json" を付け加えることで JSON 形式のデータも参照できますが、これは当サイトで適当に生成したもので、公式フォーマットで

さらにパラメータ path を追加することで元データの一部を参照できます。以下の例を試してみてください。

http://api.aitc.jp/jmardb-api/reports/761ef459-a93d-4b46-8cd6-c2d13ec4a85c.json?path=/report/head

JSON は元データである XML とは異なり、単語の先頭は小文字で統一されています。
親オブジェクトの名前を順に指定する、いわゆる「パス」のような記述を受け取ります。 JSONPath のような特別な構文は持っていません。
また制限事項として、パスが指し示すオブジェクトが配列の場合はインデックスを指定しなければなりません。（集約のようなことができません）
例えば配列 /report/head/headline/information の最初の要素は /report/head/headline/information**/0** のように記述します。

JSON形式のデータの作り方

こちらに貼っておきました

https://gist.github.com/5d284a4d2b3755d3962b9eba7e6c882f

- １００行くらいのJavaで、JAXBとJSONICというライブラリを使用しています

JSON形式のデータの作り方

気象庁防災情報 XML で JAXB する時の注意点 (1)

Body に実際のインスタンスを入れてもらうため、 @XmlElementRef を入れる必要がある。
また、JSON 化する際は getter/setter の名前が見られるため、名前を変えておく。

```
public class TypeReport {

    @XmlElement(name = "Control", required = true)
    protected TypeControl control;

    @XmlElement(name = "Head", namespace = "http://xml.kishou.go.jp/jmaxml1/informationBasis1/" , required = true)
    protected TypeHead head;

    @XmlAnyElement(lax = true)
    @XmlElementRefs({
        @XmlElementRef(name="Body", namespace="http://xml.kishou.go.jp/jmaxml1/body/meteorology1/" , type=jp.go.kishou.xml.jmaxml1.bod
        @XmlElementRef(name="Body", namespace="http://xml.kishou.go.jp/jmaxml1/body/seismology1/" , type=jp.go.kishou.xml.jmaxml1.body
        @XmlElementRef(name="Body", namespace="http://xml.kishou.go.jp/jmaxml1/body/volcanology1/" , type=jp.go.kishou.xml.jmaxml1.bod
      })
    protected Object any;
```

JSON形式のデータの作り方

気象庁防災情報 XML で JAXB する時の注意点 (2)

3種類の Body が定義されるが、それぞれのクラスで @XmlRootElement を追加する。
この要素の構築方法がプロセッサに見えるようにする必要がある。

```
@XmlRootElement(name="Body" , namespace="http://xml.kishou.go.jp/jmaxml1/body/meteorology1/" )
@XmlAccessorType(XmlAccessType.FIELD )
@XmlType(name = "type.Body", propOrder = {
    "targetArea",
    "notice",
    "warning",
    "meteorologicalInfos",
    "comment",
    "officeInfo",
    "additionalInfo"
})
public class TypeBody {
```

JSON形式のデータの作り方

```
@Override
protected boolean ignore(Context context, Class<?> clazz, Member member) {

        // 以下のメンバは JSON には含めない
        String name = member.getName();
        if (name.equals("getXMLSchemaType")) {
                return true;
        }

        return super.ignore(context, clazz, member);
}

@Override
protected Object preformat(Context context, Object value)
                throws Exception {

        // 以下の型は出力を簡略化する
        if (value instanceof XMLGregorianCalendar) {
                return value.toString();
        }
        if (value instanceof Duration) {
                return value.toString();
        }
        if (value instanceof TypeDateTime) {
                return preformat(context, ((TypeDateTime) value).getValue());
        }

        return super.preformat(context, value);
}
```

JSON形式のデータの作り方

- PostgreSQLの xslt_process() 関数を利用してSQL文でXMLスタイルシートを呼んで変換もできたのですが…
  - 気象防災情報の種類が多すぎる
  - スタイルシートを記述しようとしてもどうしても頭が追い付かない

JSON形式データの作り方

ちなみに REST API で JSON の指定部分を切り出す機能は PostgreSQLの json_extract_path() 関数で作られています

```
SELECT json_extract_path(report_json, VARIADIC ?) FROM reports WHERE id = ?
```

検索結果に含まれる link で、気象庁防災情報 XML の元データを参照できます。
URL の末尾に ".json" を付け加えることで JSON 形式のデータも参照できますが、これは当サイトで適当に生成したもので、公式フォーマットで

さらにパラメータ path を追加することで元データの一部を参照できます。以下の例を試してみてください。

http://api.aitc.jp/jmardb-api/reports/761ef459-a93d-4b46-8cd6-c2d13ec4a85c.json?path=/report/head

JSON は元データである XML とは異なり、単語の先頭は小文字で統一されています。
親オブジェクトの名前を順に指定する、いわゆる「パス」のような記述を受け取ります。 JSONPath のような特別な構文は持っていません。
また制限事項として、パスが指し示すオブジェクトが配列の場合はインデックスを指定しなければなりません。（集約のようなことができません）
例えば配列 /report/head/headline/information の最初の要素は /report/head/headline/information**/0** のように記述します。

大雨危険度通知がない理由

気象庁防災情報 XML 一覧

注意事項 2019-09-17 **「蓄積データ参照＆REST API」では大雨危険度通知を利用できません。**「SPARQL クエリ発行」「WebSocketによる配信」を使用してください。

変更履歴 2014-06-26 API の応答に ISO8601 形式の日時が追加されました。

大雨危険度通知がない理由

テーブルにINSERTできないから！

```
< 2019-07-13 06:02:22.714 JST >ERROR: index row requires 33240 bytes, maximum size is 8191
< 2019-07-13 06:02:22.714 JST >STATEMENT: INSERT INTO reports (id, report, received) VALUES
($1, $2, CURRENT_TIMESTAMP)
```

大雨危険度通知がない理由

これまで紆余曲折を経て20個近いインデックスが定義されていて…

"reports_pkey" PRIMARY KEY, btree (id)

"report_areacode" gin ((xpath('//basis:Area/basis:Code/text()'::text, report, ARRAY[ARRAY['basis'::text, 'http://xml.kishou.go.jp/jmaxml1/informationBasis1/'::text]])::text[]))

"report_areacode_mete" gin ((xpath('//mete:Area/mete:Code/text()'::text, report, ARRAY[ARRAY['mete'::text, 'http://xml.kishou.go.jp/jmaxml1/body/meteorology1/'::text]])::text[]))

"report_areaname" gin ((xpath('//basis:Area/basis:Name/text()'::text, report, ARRAY[ARRAY['basis'::text, 'http://xml.kishou.go.jp/jmaxml1/informationBasis1/'::text]])::text[]))

"report_areaname_mete" gin ((xpath('//mete:Area/mete:Name/text()'::text, report, ARRAY[ARRAY['mete'::text, 'http://xml.kishou.go.jp/jmaxml1/body/meteorology1/'::text]])::text[]))

以下略

大雨危険度通知がない理由

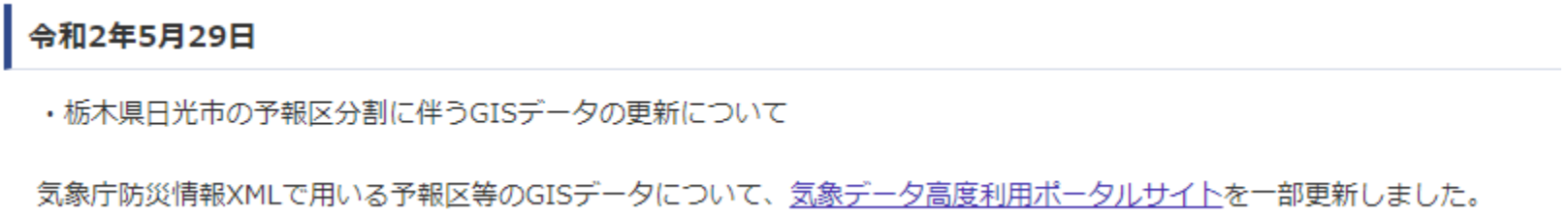
- 大雨危険度通知に大量のAreaが含まれる
  そのためインデックスの最大サイズを超える
- いくつかインデックスを廃止してみたが、
  他の情報の検索にも影響するため断念
- データのハッシュ値をインデックスする手法も
  あるらしいが、それでも超えそう

今、考えていること

- 作り始めた頃に考えていたことに影響されて特化しなさすぎるのも、しんどい
  - すべての種類の情報をテーブル1つで
  - PostgreSQLの機能で解決する
  - 種類が増えたりしても呑み込めるはずだったが…
- 実際のサービスを作る時は、目的に合わせてきちんと設計できると良いなあ…

今、考えていること

- 特に気にかかるのはエリアに関して
  - どの目的のサービスでも速度のネックになりそう
  - ときどき更新されるので、どう追従するか…



令和2年5月29日

・栃木県日光市の予報区分割に伴うGISデータの更新について

気象庁防災情報XMLで用いる予報区等のGISデータについて、気象データ高度利用ポータルサイトを一部更新しました。

REST API に関してはここまでです