

コードレビュー

Vol.15

USP 研究所技術研究員
written by 大内智明

CODE Review

今回は、公共機関から公開されている情報（郵便番号）の取得と加工についてお話しします。

公共機関から公開されている情報

業務システムは、店舗情報や顧客情報を管理するために、郵便番号一住所を紐づけるマスタが必要になります。このような郵便番号マスタは、公共機関が公開しているので、そのままダウンロードして使用することになります。郵便番号情報は、内容の変更が発生するため、定期的にデータ取得及び更新が必要になります（次頁／図2）。

技術的な概要

公共機関から取得した素のデータは、そのままの形式だと業務システムとデータ形式が異なる／不要なレイアウトや文字列が含まれるため、業務システムで使用できない場合があります。そこで素のデータは、業務システム

で使用できる形式に変換したり、業務アプリケーションが取扱いしやすい形に加工します。

データ取得は、次の3段階となります。

1：データ取得（LV1）

公共機関から公開されている情報は、curl コマンドで該当 HP から必要なデータのダウンロードを行います。

2：データ変換（LV2）

ダウンロードしたデータが CSV 形式の場合は、ユニケースで使用しやすいフィールド形式に変換します。

3：データ加工（LV3）

最終的にデータは、システム内で使用し易い形に加工します。

図1 データ取得

1. データ取得した LV1は、CSV形式／SJISファイルとなります。

```
01101,"060","0600042","ホッカイトウ","サッポロシチュウオウ","オホ"オリニシ(1-19チョウ),"北海道","札幌市中央区","大通西（1～19丁目）","1,0,1,0,0,0
01101,"064","0640820","ホッカイトウ","サッポロシチュウオウ","オホ"オリニシ(20-28チョウ),"北海道","札幌市中央区","大通西（20～28丁目）","1,0,1,0,0,0
01101,"060","0600031","ホッカイトウ","サッポロシチュウオウ","サッポロシチュウオウ","北海道","札幌市中央区","北一条東","0,0,1,0,0,0
:
```

2. データ変換した LV2は、フィールド形式（区切り文字が半角スペース）となります。

```
01101 060 0600042 ホッカイトウ サッポロシチュウオウ オホ"オリニシ(1-19チョウ) 北海道 札幌市中央区 大通西（1～19丁目） 1 0 1 0 0 0
01101 064 0640820 ホッカイトウ サッポロシチュウオウ オホ"オリニシ(20-28チョウ) 北海道 札幌市中央区 大通西（20～28丁目） 1 0 1 0 0 0
01101 060 0600031 ホッカイトウ サッポロシチュウオウ サッポロシチュウオウ 北海道 札幌市中央区 北一条東 0 0 1 0 0 0
:
```

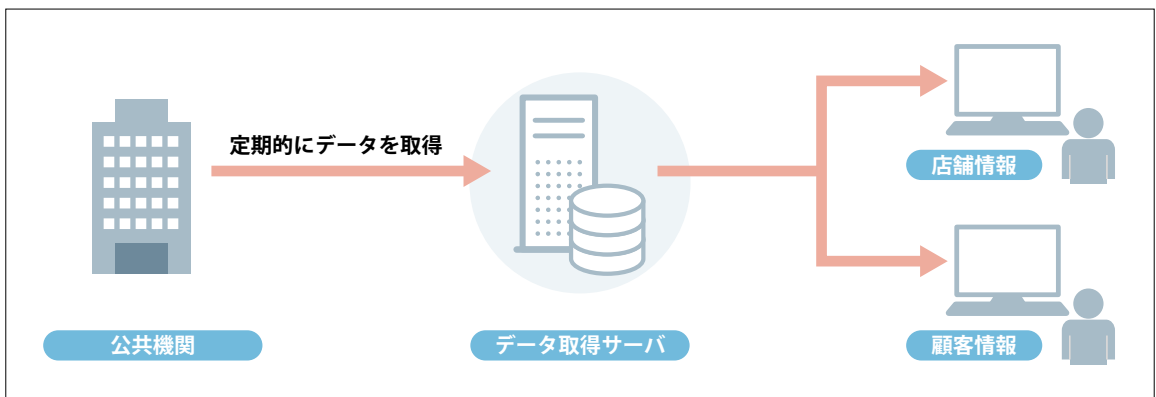
フィールド形式にして
ユニケース内で使用しやすい形にします。

3. データ加工した LV3は、業務アプリケーションで使用しやすい形にします。

```
0600042 01101 01 北海道 札幌市中央区 大通西（1～19丁目）
0640820 01101 01 北海道 札幌市中央区 大通西（20～28丁目）
0600031 01101 01 北海道 札幌市中央区 北一条東
:
```

キーの郵便番号に対して、
必要な項目だけに絞った形にします。

図2 データの流れ



リスト1 該当 HP から受信データ (LV1) を作成するシェル

```

1 # curl にてファイルを取得
2 # -v は途中経過表示, -R はファイル日付を取得元に合わせる
3 # -o (ファイル名) で出力ファイル名指定
4
5 # 郵便番号
6 curl --connect-timeout 600 -v \
7     -o ${tmp}-ken_all.zip "http://www.sss/ttt "
8
9 # 事業所
10 curl --connect-timeout 600 -v \
11     -o ${tmp}-jigyo.zip "http://www.xxx/yyy"
12

```

リスト2 受信データ (LV2) から LV3を作成するシェル

```

1 #!/bin/ush -xve
2 # システム名      :U S Pシステム
3 # サブシステム名:その他マスタ
4 # 業務名          :その他マスタ
5 # プログラム名   :日本郵政住所LV3保存
6 # 概要           :日本郵政住所LV2 から郵便番号マスタLV3を作成する
7 # 詳細           :
8 # 備考(Usage)    :LV3MAKE.MONIF.POSTNO [YYYYMMDD]
9 # シェル名       :LV3MAKE.MONIF.POSTNO
10 # 作成日        :20xx/xx/xx
11 # 会社名        :USP
12 # 作成者        :Y.Shinmi
13
14 #/////////////////////////////////////////////////////////////////
15 # 初期設定
16 #/////////////////////////////////////////////////////////////////
17
18 # 走行ログの記録
19 logd="${HOME}/LOG" # ログディレクトリ
20 logf="${logd}/LOG.${basename $0}.${date +%Y%m%d}_${date +%H%M%S}_" # ログファイル名
21 echo "${logf}" &> /dev/null
22 log 2> ${logf}
23

```

```

24 # パスの定義
25 PATH=/home/UTL:/home/TOOL:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:${PATH}
26 LANG=ja_JP.UTF-8
27
28 param=$(echo ${@} | yarr | tr '/' '-' | tr ' ' '_')
29
30 #-----
31 # 変数の定義
32 #-----
33 tmp=/tmp/${basename $0}_${date +%Y%m%d%H%M%S}          # 一時ファイル
34 hostname="$(ghostname)"                                # サーバ名
35 semd="${HOME}/SEMAPHORE"                                # セマフォディレクトリ
36
37 lv2d=/home/DATA/LV2                                     # LV2ディレクトリ
38 lv3d=/home/DATA/LV3                                     # LV3ディレクトリ
39
40 # エラー時の終了処理定義 (ushエラーハンドラ)
41 err ERROR_EXIT(){
42     echo "${hostname} ${basename $0}_${param} ERROR $(date +%Y%m%d%H%M%S) ${logf}" >> ${logd}/UPCNT
43     touch ${semd}/${basename $0}.${hostname}.ERROR.${param}
44     exit 1
45 }
46
47 # 処理日付の確認
48 sday=$(date +%Y%m%d)
49 # 引き数の確認
50 [ $# -eq 1 ] && sday=$1
51
52 # 簡易YYYYMMDD日付チェック
53 if ! isdate ${sday} ; then
54     echo "Parameter DATE error:[${sday}]"
55     ERROR_EXIT
56 fi
57
58 # 前回セマフォの消去
59 rm -f ${semd}/${basename $0}.${hostname}.*.${param} >/dev/null 2>&1
60
61 # 起動時刻の記録
62 echo "${hostname} ${basename $0}_${param} START $(date +%Y%m%d%H%M%S)" >> ${logd}/UPCNT
63 touch ${semd}/${basename $0}.${hostname}.START.${param}
64
65 # 1:項目順 2:項目名 3:出力可否(KEY_VALUE分割対象) 4:コメント
66 cat <<ETX | gawk '($1~/^#/) && length($0)>0' | msort key=1 > ${tmp}-table
67 1 POSTNO          0 郵便番号*
68 2 DMY1            0 全国地方公共団体コード
69 3 TODOFUKENCD     1 都道府県コード
70 4 TODOFUKENNM     1 都道府県名
71 5 SHIKUCHOSON     1 市区町村名
72 6 MACHICHIKI      1 町地域名
73 7 CHOUMETOUNM     1 丁目等名
74 ETX
75
76 keyname="POSTNO"
77 keynum=1
78
79 #/////////////////////////////////////////////////////////////////
80 # データ処理部
81 #/////////////////////////////////////////////////////////////////

```

msort はユニケージコマンド。
sort コマンドの高速版。

```

82 ddaytime=$(date +%Y%m%d%H%M%S)
83 userid=batch
84 sym=${sday}.1.6)
85
86 #-----
87 # 受信ファイル存在チェック
88 #-----
89 lv2outd="${lv2d}/SONOTA/JAPANPOST/${sym}"
90
91 [ ! -e ${lv2outd}/JAPANPOST_POSTNO.gz ] && ERROR_EXIT
92 [ ! -e ${lv2outd}/JAPANPOST_JIGYOUSHO.gz ] && ERROR_EXIT
93
94 # 郵便番号
95 zcat ${lv2outd}/JAPANPOST_POSTNO.gz |
96 # 1:全国地方公共団体コード 2:旧郵便番号 3:郵便番号
97 # 4:都道府県名カナ 5:市区町村名カナ 6:町地域名カナ
98 # 7:都道府県名 8:市区町村名 9:町地域名
99 # 10:一町地域が二以上 11:小字毎に番地が起番の町域
100 # 12:丁目を有する町域 13:一郵便番号で二以上町域
101 # 14:更新の表示 15:変更理由
102 self 3 1 1.1.2 7/9 - |
103 # 1:郵便番号 2:全国地方公共団体コード
104 # 3:全国地方公共団体コードの前2桁
105 # 4:都道府県名 5:市区町村名 6:町地域名
106 gawk '{ print $0, "_", "2" }' |
107 # 1:郵便番号
108 # 2:全国地方公共団体コード
109 # 3:全国地方公共団体コードの前2桁(都道府県コード)
110 # 4:都道府県名 5:市区町村名 6:町地域名
111 # 7:丁目等名(空白)
112 # 8:ソートキー"2"
113 self 1 NF 2/NF-1 - > ${tmp}-wk.postno
114 # 1:郵便番号 2:ソートキー 3:全国地方公共団体コード 4:都道府県コード
115 # 5:都道府県名 6:市区町村名 7:町地域名 8:丁目等名
116
117 # 事業所
118 zcat ${lv2outd}/JAPANPOST_JIGYOUSHO.gz |
119 # 1:大口事業所所在地JISコード 2:大口事業所名カナ 3:大口事業所名
120 # 4:都道府県名 5:市区町村名 6:町地域名
121 # 7:小字名丁目番地等 8:大口事業所個別番号 9:旧郵便番号
122 # 10:取扱局 11:個別番号種類 12:複数番号有無
123 # 13:修正コード
124 self 8 1 1.1.2 4/7 - |
125 # 1:大口事業所個別番号(郵便番号)
126 # 2:大口事業所所在地JISコード(全国地方公共団体コード)
127 # 3:大口事業所所在地JISコードの前2桁(都道府県コード)
128 # 4:都道府県名 5:市区町村名 6:町地域名 7:小字名丁目番地等
129 gawk '{ print $0, "1" }' |
130 # 8:ソートキー"1"
131 self 1 NF 2/NF-1 - > ${tmp}-wk.jigyō
132 # 1:郵便番号 2:ソートキー 3:全国地方公共団体コード 4:都道府県コード
133 # 5:都道府県名 6:市区町村名 7:町地域名 8:丁目等名
134
135 #-----
136 # 郵便番号+事業所郵便番号をマージ、郵便番号をキーに
137 # 重複は郵便番号を優先して処理する
138 #-----
139 cat ${tmp}-wk.postno ${tmp}-wk.jigyō |

```

ush(ユニーク専用シェル)で変数から文字列を抜き出す。

\$ a="123456"

\$ echo \${a.2.2}

23

self(はユニークコマンド。
フィールド選択して抽出する。

2

```

140 # 1:郵便番号 2:ソートキー 3:全国地方公共団体コード 4:都道府県コード
141 # 5:都道府県名 6:市区町村名 7:町地域名 8:丁目等名
142 msort key=1/2
143 getlast 1 1
144 delf 2 -
145 # 1:郵便番号 2:全国地方公共団体コード 3:都道府県コード
146 # 4:都道府県名 5:市区町村名 6:町地域名 7:丁目等名
147 tee ${tmp}-wk.new |
148 gawk '{ print $0, "1", "${sday}";'
149 "${userid}"; "${ddaytime}"; }' > ${tmp}-wk.rireki
150 # 1:郵便番号 2:全国地方公共団体コード 3:都道府県コード
151 # 4:都道府県名 5:市区町村名 6:町地域名 7:丁目等名
152 # NF-3:更新フラグ NF-2:適用日 NF-1:更新者 NF:更新日時
153 -----delfはユニケージコマンド。選択したフィールドを削除する。
154
155 #-----
156 # LV3ディレクトリ
157 #-----
158 tblid=${lv3d}/SONOTA/POSTNO/TBL
159 rirekid=${lv3d}/SONOTA/POSTNO/RIREKI
160
161 mkdir -p ${tblid}
162 mkdir -p ${rirekid}
163
164 # 履歴
165
166 # データは上書き
167 mv ${tmp}-wk.rireki ${rirekid}/${sym}
168 gzip -f ${rirekid}/${sym}
169
170 # テーブル
171 cp ${tmp}-wk.new ${lv3d}/SONOTA/POSTNO/TBL/POSTNO
172
173 #-----
174 # レコード形式をKEY_VALUEにする
175 #-----
176 cat ${tmp}-table |
177 # 1:項目順 2:項目名 3:出力可否 4:コメント
178 self 2 - |
179 # 1:項目名
180 # 並び順を変えずに縦列から横行へ変更
181 yarr |
182 # マスタの1レコード目へ追加 ※1
183 cat - ${tmp}-wk.new |
184 #
185 # キー(郵便番号)=1x項目名=1にてunmapする ※2
186 unmap num=${keynum}x1 |
187 # 1:郵便番号 2:(各項目名) 3:(項目名に連動してunmapされたフィールド)
188 # 項目名でファイルを分割する
189 sorter -d ${tmp}-postno.%2
190 # 1:郵便番号 2:項目の値
191
192 # ※1
193 # POSTNO DMY1 TODOFUKENC D TODOFUKENM SHIKUCHOSON MACHICHIKI CHOUMETOUNM
194 # 0010000 01102 01 北海道 札幌市北区 以下に掲載がない場合 _
195 # 0010010 01102 01 北海道 札幌市北区 北十条西(1~4丁目) _
196 # :
197

```

getlastはユニケージコマンド。
同一キーの最後の行を出力する。

```

$ cat data
0000007 セロリ 20060201 117
0000007 セロリ 20060202 136
0000007 セロリ 20060203 221
0000017 練馬大根 20060201 31
0000017 練馬大根 20060202 127
$ getlast 1 1 data
0000007 セロリ 20060203 221
0000017 練馬大根 20060202 12723

```

2

unmapはユニケージコマンド。
mapされたファイルを元に戻す。

```

$ cat data
001 X X 1 5
001 Y Y 2 6
002 X X 3 7
002 Y Y 4 8
$ map num=1x2 data > data2
$ keta data2
* * X Y
* * X Y
001 A 1 2
001 B 5 6
002 A 3 4
002 B 7 8
$ unmap num=1x2 data2 > data3
$ cat data3
001 X X 1 5
001 Y Y 2 6
002 X X 3 7
002 Y Y 4 8

```

3

```

198 # ※2
199 # 0010000 DMY1 01102
200 # 0010000 TODOFUKENC0 01
201 # 0010000 TODOFUKENNM 北海道
202 # 0010000 SHIKUCHOSON 札幌市北区
203 # 0010000 MACHICHIKI 以下に掲載がない場合
204 # 0010000 CHOMETOUNM _
205 # 0010010 DMY1 01102
206 # 0010010 TODOFUKENC0 01
207 # :
208
209 # 1-7: 各項目の項目名
210 # 2レコード目以降、以下の並びの実データ
211 # 1:郵便番号 2:全国地方公共団体コード 3:都道府県コード
212 # 4:都道府県名 5:市区町村名 6:町地域名 7:丁目等名
213
214 # 出力可否 = 1 の項目名でループ
215 for fldnm in $(cat ${tmp}-table | gawk '$3=="1"{ print $2}'); do
216
217 # 論理エラー
218 [ ! -s ${tmp}-postno.${fldnm} ] && ERROR_EXIT
219
220 # KEY_VALUE 形式の保存
221 mv ${tmp}-postno.${fldnm} ${lv3d}/SONOTA/POSTNO/TBL/${keyname}_${fldnm}
222 # 1:郵便番号 2:項目の値
223 done
224
225 #/////////////////////////////////////////////////////////////////
226 # 終了
227 #/////////////////////////////////////////////////////////////////
228 # 終了時刻の保存
229 echo "${hostname} ${basename $0}_${param} END $(date +%Y%m%d%H%M%S)" >> ${logd}/UPCNT
230 touch ${semd}/${basename $0}.${hostname}.END.${param}
231 #
232 set +x
233 rm -f ${tmp}.* > /dev/null 2>&1
234 set -x
235 echo "${basename $0} exit 0"
236 exit 0

```

3

コードの見どころ

- [1] 日本郵政 HP から住所データをダウンロードします。
(❶リスト1)
- [2] 住所データ (LV2) から不要な項目を除いたデータ (LV3) を作成します。
(❷リスト2. 95 ~ 169行目まで)
- [3] 業務アプリケーションで使用しやすい形のデータを作成します。
(❸リスト2. 171 ~ 223行目まで)

まとめ

業務システムには、今回のように公共機関から公開されている情報を使用することがよくあります。今回は、郵便番号と住所でしたが、過去には地域（店舗）と天気情報を取得して、売上と天気との関連を見る案件もありました。

社外にある HP から取得した情報を利用することは、今後もあると思いますので、読者の方の参考になればと思います。