On peut tout mettre dans le DNS, même les codes postaux

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 24 septembre 2017. Dernière mise à jour le 9 octobre 2020

https://www.bortzmeyer.org/dns-code-postal-lonlat.html

Petit service juste pour s'amuser : mettre l'ensemble des codes postaux français dans le DNS avec pour chaque code, la ou les communes concernées, leur longitude et leur latitude.

À quoi ça sert ? Pas à grand'chose, je voulais juste illustrer l'utilisation de deux types d'enregistrements DNS peu connus, LOC (RFC 1876 [1]) et URI (RFC 7553). (Au passage, si vous lisez que le DNS sert à traduire des noms en adresses IP, allez lire autre chose : c'est faux, ou, en tout cas, très réducteur.) Et traiter un peu de JSON en prévision de la prochaine sortie du nouveau RFC sur JSON, le RFC 8259.. Voyons d'abord le résultat avec le traditionnel client DNS dig :

```
% dig +short +nodnssec TXT 34000.cp.bortzmeyer.fr
"MONTPELLIER"
```

L'enregistrement TXT permet d'indiquer le nom de la ville concernée. Parfois, il y en a plusieurs :

```
% dig +short +nodnssec TXT 52100.cp.bortzmeyer.fr
"HALLIGNICOURT"
"VALCOURT"
"CHANCENAY"
"SAPIGNICOURT"
"ST DIZIER"
"PERTHES"
"ST EULIEN"
"BETTANCOURT LA FERREE"
"LANEUVILLE AU PONT"
"VILLIERS EN LIEU"
"MOESLAINS"
```

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc1876.txt

Le même domaine `cp.bortzmeyer.fr` contient également la position de la ville (ou plus probablement celle du bureau de poste principal), avec le type `LOC`, normalisé dans le RFC 1876) :

```
% dig +short +nodnssec LOC 34000.cp.bortzmeyer.fr
43 36 47.108 N 3 52 9.113 E 0.00m 1m 10000m 10m
```

Nous voyons ici que Montpellier est à 43° 36′ 47″ de latitude Nord et 3° 52′ 9″ de longitude Est. Si vous voulez voir tout de suite où ça se trouve, vous avez également un `URI` (type d'enregistrement DNS normalisé dans le RFC 7553) qui pointe vers OpenStreetMap (merci à OpenStreetMap de permettre de fabriquer des URL simples <https://www.bortzmeyer.org/beaux-urls.html>) :

```
% dig +short +nodnssec URI 52100.cp.bortzmeyer.fr
10 1 "http://www.openstreetmap.org/?mlat=48.646559&mlon=4.912187&zoom=12"
```

(L'enregistrement URI étant relativement récent, il ne faut pas utiliser un dig trop vieux. Par exemple, avec la version 9.8.3, on n'a pas de résultat avec cette commande.) Enfin, vous pouvez aussi retrouver les codes postaux à partir du nom de la ville :

```
% dig +short +nodnssec TXT ROUEN.cp.bortzmeyer.fr
"76100"
"76000"
```

Mais il y a quelques pièges : la base utilisée n'a que des caractères ASCII donc il faudra écrire « epernay » et pas le nom correct. Et pour Paris, il faudra ajouter un tiret et l'arrondissement, par exemple « paris-16 ».

Ce n'est pas très joli avec dig ? Pas convivial ? Vous pouvez aussi essayer un "*DNS Looking Glass*" <https://www.bortzmeyer.org/dns-lg-usage.html>. Par exemple, toutes les données pour 73370 <https://dns.bortzmeyer.org/73370.cp.bortzmeyer.fr/ANY>, la localisation de 73700 <https://dns.bortzmeyer.org/73700.cp.bortzmeyer.fr/LOC>, les les noms associés à 97434 <https://dns.bortzmeyer.org/97434.cp.bortzmeyer.fr/TXT> ou bien le(s) code(s) postal(aux) de Beaugency <https://dns.bortzmeyer.org/beaugency.cp.bortzmeyer.fr/TXT>.

Comment est-ce que ce service a été réalisé ? Je suis parti d'un fichier JSON contenant les informations nécessaires, un petit programme Python le transforme en fichier de zone DNS (section 5 du RFC 1035). Que du très simple. Le plus dur était certainement de trouver les données. En France, il y a une tradition régalienne de ne pas donner les informations au citoyen. L'État préfère les garder pour lui, ce qui obligeait à des contorsions compliquées <http://owni.fr/2012/10/12/le-datajournalisme-appliquee-a-l
index.html>. Le mouvement « données ouvertes » a heureusement changé les choses, même si c'est encore loin d'être parfait.

La source que j'ai utilisée est la « Base officielle des codes postaux <https://datanova.legroupe.laposte.fr/explore/dataset/laposte_hexasmal/export/?disjunctive.code_commune_insee&disjunctive.nom_de_la_commune&disjunctive.code_postal&disjunctive.libell_d_acheminement&disjunctive.ligne_5> » (trouvée via data.gouv.fr <http://www.data.gouv.fr/fr/datasets/base-officielle-des-codes-postaux/>). Je ne dis pas que cette base est meilleure que les autres, mais son côté « officiel » me permet de répondre « c'est pas ma faute » si quelqu'un y trouve une erreur ou une approximation. J'ai utilisé la version GeoJSON du fichier. Elle n'est pas parfaite. Ainsi il y a un certain nombre de fautes d'orthographe comme Marolles-sous-Lignières écrit MAROLLES SOUS LIGNIERES, sans l'accent (les tirets et les apostrophes manquent également). D'autre part, certaines communes, notamment en Polynésie française, n'ont pas d'information de localisation. Ici, à Fakarava :

```
% dig +short +nodnssec LOC 98764.cp.bortzmeyer.fr
%
```

Dernier problème, la liste n'est pas à jour. La commune de Meaulne-Vitray (code postal 03360), créée le 1er janvier 2017, n'apparait pas encore dans la liste, téléchargée le 24 septembre 2017. (Et elle n'est pas la seule.)

D'autres sources possibles de données avaient été considérées (et merci à Sabine Blanc, Olivier Macchioni, Pascal Corpet, Laurent Penou, Adrien Clerc, Ainiton et tous les autre qui m'ont fait d'excellentes suggestions) :

- On trouve à certains endroits des fichiers structurés contenant cette information, mais ils ne sont pas forcément à jour (les codes postaux changent rarement mais ils changent). Ainsi, `http://www.aito.fr/maps_ville.sql` est une excellente base MySQL, où les noms sont bien orthographiés, mais qui est vieille de nombreuses années. D'autre part, elle semble incomplète (Lille n'y a qu'un seul code postal alors qu'il en existe plusieurs en pratique).
- Geofla <http://professionnels.ign.fr/geofla> a remplacé l'ancien RGC de l'IGN mais il ne semble pas accessible dans un format ouvert. (Et, de toute façon, il a les longitude et latitude mais pas les codes postaux, il aurait donc fallu faire une jointure sur les noms des communes.) À l'IGN, il y a aussi Admin Express <http://professionnels.ign.fr/adminexpress>, dans un format que je ne connais pas.
- Souvent, les meilleures bases sont celles gérées par des initiatives citoyennes, comme NosDonnées. Le fichier des communes <http://www.nosdonnees.fr/dataset/listes-des-communes-par-r-gions- semble une piste intéressante , et où les noms sont correctement orthographiés (mais je préfère JSON à CSV).
- À une époque, la Poste, pourtant service public, vendait la base des codes postaux. Il semble qu'il existe toujours un service commercial <https://www.laposte.fr/entreprise/produits-et-services/sna-normalisation-des-adresses?id_rubrique=40> à ce sujet.
- Les données nécessaires se trouvent dans Wikipédia (regardez par exemple la fiche de Saint-Dizier, code postal 52100, postition 48° 38[Caractère Unicode non montré [2]] 21[Caractère Unicode non montré] nord, 4° 57[Caractère Unicode non montré] 09[Caractère Unicode non montré] est). On pourrait l'extraire à partir de liste des communes. Mais peut-être que Wikidata offre désormais une solution plus simple (un exemple de requête SPARQL <https://query.wikidata.org/#SELECT%20DISTINCT%20%3Fitem%20%3FitemLabel%20%3Fcode_postal%20%3Fcoordonnees_geographiques%20WHERE%20{%0A%20%3Fitem%20(wdt%3AP31%2Fwdt%3AP279*)%20wd%3AQ484170.%0A%20%0A%20OPTIONAL%20{%20%3Fitem%20wdt%3AP281%20%3Fcode_postal.%20}%0A%20OPTIONAL%20{%20%3Fitem%20wdt%3AP625%20%3Fcoordonnees_geographiques.%20}%0A%20%0A%20SERVICE%20wikibase%3Alabel%20{%20bd%3AserviceParam%20wikibase%3Alanguage%20%22fr%22.%20}%0A}>).
- La base d'adresses nationale <https://adresse.data.gouv.fr> est téléchargeable sous forme d'un gros fichier JSON, apparemment à jour, où on trouve les longitudes et latitudes de chaque rue (et associées au code postal). Donc, cela pourrait être une meilleure alternative, et elle est sous une licence libre.
- Le site permet de trouver codes postaux et positions géographiques, et sa base (à jour ? correcte ?) est téléchargeable (au format Microsoft Excel).
- La base de données de Geonames <http://www.geonames.org/export/> semble prometteuse.

Si vous voulez proposer d'autres bases, n'oubliez pas de regarder ces points :

- Format ouvert,
- Base légalement téléchargeable et utilisable,
- Base complète (les COM...),

2. Car trop difficile à faire afficher par LaTeX

— Disponibilité de la longitude et de la latitude,
— Téléchargeable en bloc (pas limité à une API),
— Base à jour (les communes changent, et les codes postaux aussi, d'ailleurs, si quelqu'un connait une liste des récents changements, afin de pouvoir tester la fraîcheur des bases…Je trouve des informations non officielles <https://www.data.gouv.fr/fr/datasets/communes-nouvelles-au-1e> ou anciennes <https://www.insee.fr/fr/information/2549968>, Christian Quest me signale que la bonne est apparemment à la Poste <https://datanova.legroupe.laposte.fr/explore/dataset/laposte_commnouv/?disjunctive.insee&disjunctive.commune_deleguee>),
— Orthographe correcte (respect des accents, pas d'abréviation genre ST pour Saint).

Ce service utilise les codes postaux, mais il existe aussi les COG, dits « codes INSEE », qui sont également disponibles <https://www.data.gouv.fr/fr/datasets/code-officiel-geographique-cog/>. Ce sera pour un autre service.

Un peu de technique pour finir. Le script lancé par cron tous les mois fait :

```
wget -O laposte_hexasmal.geojson 'https://datanova.legroupe.laposte.fr/explore/dataset/laposte_hexasmal/down
./laposte2dns.py laposte_hexasmal.geojson > cp.zone
ods-signer sign bortzmeyer.fr
```

La première commande télécharge le fichier au format JSON. La seconde (le script (en ligne sur https://www.bortzmeyer.org/files/laposte2dns.py)) convertit le JSON en fichier de zone DNS qui ressemble donc à :

```
10130.cp IN TXT "MAROLLES SOUS LIGNIERES"
   IN LOC 48 3 7.201 N 3 55 56.876 E 0
   IN URI 10 1 "http://www.openstreetmap.org/?mlat=48.052000&mlon=3.932466&zoom=12"

10170.cp IN TXT "LES GRANDES CHAPELLES"
   IN LOC 48 29 29.856 N 3 56 0.604 E 0
   IN URI 10 1 "http://www.openstreetmap.org/?mlat=48.491627&mlon=3.933501&zoom=12"

10110.cp IN TXT "MERREY SUR ARCE"
   IN LOC 48 6 57.679 N 4 25 51.357 E 0
   IN URI 10 1 "http://www.openstreetmap.org/?mlat=48.116022&mlon=4.430933&zoom=12"

...
```

Le fichier ainsi produit (dans les onze mégaoctets) est chargé depuis le fichier de zone principal de bortzmeyer.fr, via une directive d'inclusion :

```
; Codes Postaux
$INCLUDE /etc/nsd/primary/cp.incl
```

Enfin, la troisième commande utilise OpenDNSSEC pour signer la zone, et recharger le serveur de noms.

Un tel service dans le DNS existe déjà pour le Royaume-Uni, via le domaine find.me.uk :

```
% dig +short DY76JP.find.me.uk LOC
52 26 46.924 N 2 13 5.686 W 0.00m 0.00m 0.00m 0.00m
```

(Ou bien, avec le Looking Glass <https://dns.bortzmeyer.org/DY76JP.find.me.uk/LOC>.) Un travail similaire a été fait <https://github.com/gryphius/ch-loc> pour la Suisse avec zipdns.ch. Par exemple :

```
% dig +short LOC 1204.zipdns.ch
46 12 14.214 N 6 8 47.064 E 1.00m 1m 10000m 10m
```

Ce qui est plus joli avec le Looking Glass <https://dns.bortzmeyer.org/1204.zipdns.ch/ANY>. Pour l'Allemagne, c'est en cours <https://gist.github.com/jpmens/b271db3b9ca8ee7d66514bd30ba3a> (il n'y a pas encore la position physique)) sur zipde.jpmens.net. Essayons avec Ratisbonne :

```
 %  dig +short TXT 93047.zipde.jpmens.net
"Regensburg"
```

Il y a un œuf de Pâques dans ce service, faisant référence à une célèbre blague allemande. Et il y a enfin un service similaire pour les codes IATA à trois lettres des aéroports (souvent utilisés pour nommer les routeurs dans l'Internet, ce qui aide à interpréter les traceroutes) en air.jpmens.net :

```
% dig +short ORY.air.jpmens.net LOC
48 43 24.000 N 2 22 46.000 E 89.00m 1m 10000m 10m
```

Une requête TXT vous en dira plus :

```
% dig +short ORY.air.jpmens.net TXT
"u:https://en.wikipedia.org/wiki/Orly_Airport_(Paris)"
"cc:FR; m:Paris; t:large, n:Paris-Orly Airport"
```

Là aussi, vous pouvez préferer le Looking Glass <https://dns.bortzmeyer.org/ORY.air.jpmens.net/ANY>. Ce service est documenté dans un intéressant article <https://jpmens.net/2020/10/04/airports-of-the-world/>. (Un service analogue utilisant les UN/LOCODE existe en locode.sha256.net mais semble expérimental.)

Et pour rire, une quetsion difficile à la fin : faut-il indiquer la latitude ou la longitude en premier ? (ISO 6709 dit qu'on met la latitude avant).

---

https://www.bortzmeyer.org/dns-code-postal-lonlat.html