RFC 8521 : Registration Data Access Protocol (RDAP) Object Tagging

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 23 novembre 2022

Date de publication du RFC : Novembre 2018

https://www.bortzmeyer.org/8521.html

Contrairement à son lointain prédécesseur whois, le protocole d'accès aux informations des registres RDAP a un mécanisme standard de découverte du serveur faisant autorité. Ce mécanisme est prévu pour des données organisées de manière arborescente comme les noms de domaine ou les adresses IP. Pour des objets qui n'ont pas cette organisation, comment faire ? Ce RFC fournit une solution pour des objets quelconques, s'ils obéissent à une convention de nommage simple. C'est le cas des "*handles*", ces identificateurs d'entités (titulaires de noms de domaine, contacts pour un préfixe IP, etc) ; s'ils sont de la forme QUELQUECHOSE-REGISTRE, on pourra trouver le serveur responsable.

Le mécanisme habituel de découverte du serveur est normalisé dans le RFC 9224[1]. En gros, l'IANA garde un registre des serveurs, indexé par un nom de domaine ou par un préfixe IP. Ce registre est au format JSON. Un client RDAP est juste censé télécharger ce fichier, trouver le serveur faisant autorité, puis le contacter. Pour pouvoir faire la même chose avec des objets non structurés, il faut leur imposer une convention de nommage. Si on interroge avec RDAP (ou whois) l'adresse IP 88.198.21.14, on voit que son contact technique est HOAC1-RIPE. Cette convention de nommage est « d'abord un identifiant unique au sein du registre, puis un tiret, puis le nom du registre, ici le RIPE-NCC ». L'IANA a juste à garder un registre de ces registres <https://data.iana.org/rdap/object-tags.json> qui nous dira, par exemple, que RIPE a comme serveur RDAP https://rdap.db.ripe.net/. On peut ensuite l'interroger en RDAP :

```
%  curl https://rdap.db.ripe.net/entity/HOAC1-RIPE
   "vcardArray" : [ "vcard", [ [ "version", { }, "text", "4.0" ], [ "fn", { }, "text", "Hetzner Online GmbH - Co
    "label" : "Hetzner Online GmbH\nIndustriestrasse 25\nD-91710 Gunzenhausen\nGermany"
  }, "text", [ "", "", "", "", "", "", "" ] ], [ "tel", {
    "type" : "voice"
  }, "text", "+49 9831 505-0" ], [ "tel", {
...
```

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc9224.txt

Petit piège, la partie du début peut elle-même comporter des tirets donc lorsqu'on coupe en deux un "*handle*", il faut le faire sur le **dernier** tiret (section 2 du RFC). Pourquoi le tiret, d'ailleurs ? Parce qu'il est couramment utilisé comme séparateur, et parce que son utilisation dans un URL ne pose pas de problème particulier (RDAP utilise HTTPS et donc des URL).

Le registre des serveurs RDAP est géré à l'IANA <https://www.iana.org/assignments/rdap-object-ta rdap-object-tags.xml> comporte un tableau JSON des services, chaque entrée étant composée de trois tableaux, la liste des adresses de courrier du responsable, la liste des suffixes (en général, un seul par registre) et la liste des serveurs RDAP. Par exemple, pour l'ARIN, on aura :

```
[
      [
        "andy@arin.net"
      ],
      [
        "ARIN"
      ],
      [
        "https://rdap.arin.net/registry/",
        "http://rdap.arin.net/registry/"
      ]
]
```

Les entrées dans le registre sont ajoutées selon la politique « Premier Arrivé, Premier Servi » du RFC 8126.

Comme ce RFC décrit une extension à RDAP <https://www.iana.org/assignments/rdap-extensions/ rdap-extensions.xml#rdap-extensions-1>, le serveur RDAP doit ajouter à sa réponse, dans l'objet rdapConformance, rdap_objectTag_level_0. (Apparemment, personne ne le fait.)

Vous voulez du code ? La bibliothèque ianardap.py, qui est disponible en ligne <https://forge. chapril.org/bortzmeyer/check_expire>, peut récupérer la base et l'analyser. Cela permet ensuite de faire des requêtes pour des objets :

```
% ./dump-object.py F11598-FRNIC  | jq .
...
       [
          "",
          "",
          "Association Framasoft c/o Locaux Motiv",
          "Lyon",
          "",
          "69007",
          "FR"
        ]
...
```

Le programme dump-object.py étant simplement :

```
#!/usr/bin/env python3

import ianardap

import requests

import sys

rdap = ianardap.IanaRDAPDatabase(category="objects")
for object in sys.argv[1:]:
    url = rdap.find(object)
    if url is None:
        print("No RDAP server for %s" % object, file=sys.stderr)
        continue
    response = requests.get("%sentity/%s" % (url[0], object))
    print(response.text)
```