

RFC 9082 : Registration Data Access Protocol (RDAP) Query Format

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 16 juin 2021

Date de publication du RFC : Juin 2021

<https://www.bortzmeyer.org/9082.html>

Le protocole d'information RDAP, qui vise à remplacer whois, est décrit dans un ensemble de RFC. Celui présenté ici normalise la façon de former les requêtes RDAP. Celles-ci ont la forme d'une URL, puisque RDAP repose sur l'architecture REST. Ce RFC remplace l'ancienne norme sur les requêtes RDAP, qui était dans le RFC 7482¹, mais il n'y a pas de changement significatif.

RDAP peut être utilisé pour beaucoup de sortes d'entités différentes mais ce RFC ne couvre que ce qui correspond aux usages actuels de whois, les préfixes d'adresses IP, les AS, les noms de domaine, etc. Bien sûr, un serveur RDAP donné ne gère pas forcément tous ces types d'entités, et il doit renvoyer le code HTTP 501 ("*Not implemented*") s'il ne sait pas gérer une demande donnée. Ce RFC ne spécifie que l'URL de la requête, le format de la réponse est variable (JSON, XML...) et le seul actuellement normalisé, au-dessus de JSON, est décrit dans le RFC 9083. Quant au protocole de transport, le seul actuellement normalisé pour RDAP (dans le RFC 7480) est HTTP. D'autre part, ces RFC RDAP ne décrivent que le protocole entre le client RDAP et le serveur, pas « l'arrière-cuisine », c'est-à-dire l'avitaillement (création, modification et suppression) des entités enregistrées. RDAP est en lecture seule et ne modifie pas le contenu des bases de données qu'il interroge.

Passons aux choses concrètes. Une requête RDAP est un URL (RFC 3986). Celui-ci est obtenu en ajoutant un chemin spécifique à une base. La base (par exemple <https://rdap.example.net/>) va être obtenue par des mécanismes divers, comme celui du RFC 7484, qui spécifie un registre que vous pouvez trouver en ligne <<http://data.iana.org/rdap/dns.json>>. On met ensuite un chemin qui dépend du type d'entité sur laquelle on veut se renseigner, et qui indique l'identificateur de l'entité. Par exemple, avec la base ci-dessus, et une recherche du nom de domaine `internautique.fr`, on construirait un URL complet <https://rdap.example.net/domain/internautique.fr>. Il y a cinq types d'entités possibles :

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc7482.txt>

- `ip` : les préfixes IP (notez qu'on peut chercher un préfixe en donnant juste une des adresses IP couvertes par ce préfixe),
- `autnum` : les numéros de systèmes autonomes,
- `domain` : un nom de domaine (notez que cela peut être un domaine dans `in-addr.arpa` ou `ipv6.arpa`),
- `nameserver` : un serveur de noms,
- `entity` : une entité quelconque, comme un bureau d'enregistrement, ou un contact identifié par un *"handle"*.

La requête est effectuée avec la méthode HTTP `GET` (les méthodes permettant de modifier le contenu du registre n'ont pas de sens ici, les modifications dans le registre sont plutôt faites avec EPP). Pour juste savoir si un objet existe, on peut aussi utiliser la méthode `HEAD`. Si on n'obtient pas de code 404, c'est que l'objet existe.

Pour `ip`, le chemin dans l'URL est `/ip/XXX` où `XXX` peut être une adresse IPv4 ou IPv6 sous forme texte. Il peut aussi y avoir une longueur de préfixe à la fin donc `/ip/2001:db8:1:a::/64` est un chemin valable. Ainsi, sur le service RDAP du RIPE-NCC, `https://rdap.db.ripe.net/ip/2001:4b98:dc0:41::` est un URL possible. Testons-le avec `curl` (le format de sortie, en JSON, est décrit dans le RFC 9083, vous aurez peut-être besoin de passer le résultat à travers `jq` <<https://www.bortzmeyer.org/jq.html>> pour l'afficher joliment) :

```
% curl https://rdap.db.ripe.net/ip/2001:4b98:dc0:41::
{
  "handle" : "2001:4b98:dc0::/48",
  "startAddress" : "2001:4b98:dc0::/128",
  "endAddress" : "2001:4b98:dc0:ffff:ffff:ffff:ffff:ffff/128",
  "ipVersion" : "v6",
  "name" : "GANDI-HOSTING-DC0",
  "type" : "ASSIGNED",
  "country" : "FR",
  "rdapConformance" : [ "rdap_level_0" ],
  "entities" : [ {
    "handle" : "GAD42-RIPE",
    "vcardArray" : [ "vcard", [ [ "version", { }, "text", "4.0" ], [ "fn", { }, "text", "Gandi Abuse Departm
      "label" : "63-65 Boulevard Massena\n75013 Paris\nFrance"
    ...
  ]
```

J'ai utilisé `curl` mais, notamment pour formater plus joliment la sortie de RDAP, les vrais utilisateurs se serviront plutôt d'un client RDAP dédié comme `RDAPBrowser` <<https://viagenie.ca/rdapbrowser/>> sur Android, ou `nicinfo` <<https://github.com/arineng/nicinfo>>. Voici une vue de `RDAPbrowser` :

Pour `autnum`, on met le numéro de l'AS après `/autnum/` (au format « `asplain` » du RFC 5396). Toujours dans l'exemple RIPE-NCC, `https://rdap.db.ripe.net/autnum/208069` permet de chercher de l'information sur l'AS 208069 <<https://blog.ataxya.net/un-as-chez-soi-cest-possible/>> :

```
% curl https://rdap.db.ripe.net/autnum/208069
{
  "handle" : "AS208069",
  "name" : "ATAXYA",
  "type" : "DIRECT ALLOCATION",
  "entities" : [ {
    "handle" : "mc40833-RIPE",
    "roles" : [ "administrative", "technical" ],
    "objectClassName" : "entity"
  }, {
  ...
}
```