

RFC 9910 : Registration Data Access Protocol (RDAP) Regional Internet Registry (RIR) Search

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 8 janvier 2026

Date de publication du RFC : Janvier 2026

<https://www.bortzmeyer.org/9910.html>

Le protocole RDAP, successeur de whois, n'est pas utilisé que par les registres de noms de domaine. Il sert aussi chez les RIR, pour obtenir des informations sur les entités qui sont derrière une adresse IP (ou un AS). RDAP dispose dès le début de fonctions de recherche (regardez le RFC 9082¹) mais ce nouveau RFC ajoute des choses en plus.

Traditionnellement, les serveurs whois des RIR disposaient de fonction de recherche « avancées » comme la possibilité de chercher par valeur (« quels sont tous les préfixes IP de tel titulaire ? »). L'idée de ce RFC est de permettre la même chose en RDAP. RDAP de base permet les recherches des informations associées à une adresse IP :

```
% curl -s https://rdap.db.ripe.net/ip/2001:41d0:302:2200::180 | jq .
{
  "handle": "2001:41d0::/32",
  "name": "FR-OVH-20041115",
  "country": "FR",
  "parentHandle": "2001:4000::/23",
  ...
  "status": [
    "active"
  ],
  "entities": [
    {
      "handle": "OK217-RIPE",
      ...
      "text",
      "Octave Klabaa"
    ...
  
```

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc9082.txt>

En plus de `ip`, ce RFC ajoute `ips` (section 2.1) qui permet une recherche sur tous les préfixes dont le nom correspond à un certain motif (ici, tous ceux d'OVH) :

```
% curl -s https://rdap.db.ripe.net/ips\?name="FR-OVH-*" | jq '.ipSearchResults.[] .handle'  
"109.190.0.0 - 109.190.255.255"  
"135.125.0.0 - 135.125.255.255"  
"137.74.0.0 - 137.74.255.255"  
"141.94.0.0 - 141.95.255.255"  
"145.239.0.0 - 145.239.255.255"  
"147.135.128.0 - 147.135.255.255"  
"149.202.0.0 - 149.202.255.255"  
"152.228.128.0 - 152.228.255.255"  
"159.173.0.0 - 159.173.255.255"  
"162.19.0.0 - 162.19.255.255"  
...  
...
```

J'ai utilisé ici `curl` et `jq` (<https://www.bortzmeyer.org/jq.html>) mais, évidemment, l'avantage de RDAP est que vous pouvez utiliser un client dédié ou bien n'importe quel logiciel qui sait faire du HTTP et du JSON.

Et chez un autre RIR :

```
% curl -s https://rdap.arin.net/registry/ips\?name='CLOUDFLARE*' | \  
  jq '.ipSearchResults.[] .handle'  
"NET-104-16-0-0-1"  
"NET-108-162-192-0-1"  
"NET-156-146-101-152-1"  
"NET-162-158-0-0-1"  
"NET-172-64-0-0-1"  
"NET-173-245-48-0-1"  
"NET-198-41-128-0-1"  
"NET-199-27-128-0-1"  
"NET6-2606-4700-1"
```

De la même façon, vous pouvez chercher par numéro d'AS :

```
% curl -s https://rdap.db.ripe.net/autnums\?name="*NIC-FR*" | \  
  jq '.autnumSearchResults.[] .handle'  
"AS2483"  
"AS2484"  
"AS2485"  
"AS2486"
```

La section 3 décrit ensuite des moyens de trouver les objets parents et enfants (puisque l'allocation des adresses IP est hiérarchique, toute adresse est dans un préfixe plus général et contient des préfixes plus spécifiques, cf. la section 3.2.1 du RFC) :

```
% curl -s https://rdap.db.ripe.net/ips/rirSearch1/rdap-up/2001:41d0:302:2200::180 | \  
  jq '.handle'  
"2001:41d0::/32"
```

```
% curl -s https://rdap.db.ripe.net/ips/rirSearch1/rdap-down/2001:4000::/23 | \
    jq '.ipSearchResults[].handle'
"2001:4000::/32"
"2001:4010::/32"
"2001:4018::/29"
"2001:4020::/32"
...
```

Notez la relation (`rdap-up` et `rdap-down`). Notez aussi que `rdap-up` renvoie au maximum un objet alors que `rdap-down` peut en renvoyer plusieurs (cf. section 4.2), et c'est pour cela qu'il a fallu itérer en `jq` (le `.[]` parcourt le tableau). Quant au `rirSearch1`, le `1` indique la version de cette extension de recherche chez les RIR (désormais enregistrée à l'IANA <<https://www.iana.org/assignments/rdap-extensions/rdap-extensions.xml#rdap-extensions-1>>).

Et les recherches inverses, telles que décrites dans le RFC 9536 ? La section 5 du RFC les présente et voici un exemple qui marche (trouver tous les préfixes de Webflow) :

```
% curl -s 'https://rdap.arin.net/registry/ips/reverse_search/entity?handle=WEBFL' | \
    jq '.ipSearchResults[].handle'
"NET-198-202-211-0-1"
"NET6-2620-CB-2000-1"
```

Un serveur RDAP qui gère les extensions de ce RFC doit le signaler dans ses réponses (section 6) :

```
...
"rdapConformance" : [ "geofeed1", "rirSearch1", "ips", "cidr0",
    "rdap_level_0", "nro_rdap_profile_0", "redacted" ],
...
```

Mais aussi dans les liens donnés dans les réponses (ici, en réponse à une requête traditionnelle `ip`) :

```
"links": [
  {
    "value": "https://rdap.db.ripe.net/ip/2001:41d0:302:2200::180",
    "rel": "rdap-up",
    "href": "https://rdap.db.ripe.net/ips/rirSearch1/rdap-up/2001:41d0::/32",
    "type": "application/rdap+json"
  },
...
```

Et bien sûr dans les réponses d'aide :

```
% curl -s https://rdap.arin.net/registry/help
{
  "rdapConformance" : [ "nro_rdap_profile_0", "rdap_level_0", "cidr0", "nro_rdap_profile_asn_flat_0", "arin_originator_id" ],
  "notices" : [ {
    "title" : "Terms of Service",
    "description" : [ "By using the ARIN RDAP/Whois service, you are agreeing to the RDAP/Whois Terms of Use" ],
    "links" : [ {
      "value" : "https://rdap.arin.net/registry/help",
      "rel" : "terms-of-service",
      "type" : "text/html",
      "href" : "https://www.arin.net/resources/registry/whois/tou/"
    } ]
  }
}
```

Les fonctions de recherche, c'est très bien mais c'est, par construction, indiscret. La section 8 de notre RFC détaille les risques de ces extensions de recherche pour la vie privée (relire le RFC 7481 est une bonne idée).

Les extensions de ce RFC, `rirSearch1`, `ips`, `autnums`, `ipSearchResults` et `autnumSearchResults` ont été enregistrées à l'IANA <<https://www.iana.org/assignments/rdap-extensions/rdap-extensions-xhtml#rdap-extensions-1>> (section 10 du RFC). Les relations `rdap-up`, `rdap-down`, `rdap-top` et `rdap-bottom` sont dans le registre des liens <<https://www.iana.org/assignments/link-relations-link-relations.xml#link-relations-1>> (RFC 8288). Et le registre des recherches inverses <<https://www.iana.org/assignments/rdap-reverse-search/rdap-reverse-search.xml#rdap-reverse-search>> inclut désormais `fn`, `handle`, `email` et `role`, avec leur correspondance <<https://www.iana.org/assignments/rdap-reverse-search-mapping/rdap-reverse-search-mapping.xml#rdap-reverse-search-mapping>> en JSONPath.

Et question mise en œuvre et déploiement, on a quoi ? ARIN et RIPE ont annoncé avoir programmé les extensions de ce RFC mais elles ne sont pas forcément accessibles via le serveur RDAP public, entre autre pour les raisons de vie privée discutées dans la section 8. Aujourd'hui, comme vous le voyez dans les exemples ci-dessus, au moins ARIN et RIPE rendent une partie de ces extensions utilisables.