

RFC 8982 : Registration Data Access Protocol (RDAP) Partial Response

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 10 février 2021

Date de publication du RFC : Février 2021

<https://www.bortzmeyer.org/8982.html>

Le protocole de récupération d'informations auprès d'un registre RDAP peut parfois rapporter une grande quantité d'informations pour chaque objet récupéré (un nom de domaine, par exemple). Le RDAP originel ne permettait pas au serveur de ne renvoyer qu'une information partielle s'il trouvait que cela faisait trop. C'est désormais possible avec l'extension de ce RFC.

Le but est évidemment d'économiser les ressources du serveur, aussi bien que celles du client, en récupérant et en transférant moins de données. Avant notre RFC, la seule solution était de tout envoyer au client, même s'il n'en utiliserait pas la plus grande partie. (Si, au lieu de ne vouloir qu'une partie de chaque objet, on veut une partie des objets, il faut regarder le RFC 8977¹.)

Ce RFC étend donc le langage de requêtes de RDAP (normalisé dans le RFC 9082), avec un paramètre `fieldSet` (section 2 de notre RFC) qui permet de sélectionner une partie des champs de la réponse. Voici un exemple d'URL pour une requête de tous les noms de domaine de `.com` :

https://rdap.verisign.example/domains?name=*.com&fieldSet=afieldset

Et comment on connaît les champs possibles? Notre RFC introduit un `subsetting_metadata` qui indique l'ensemble de champs courant, et les ensembles possibles, chacun identifié par un nom, et comportant les liens (RFC 8288) à utiliser. (Vous aurez une erreur HTTP 400 si vous êtes assez vilain-e pour faire une requête RDAP avec un ensemble inexistant.) Et pour savoir si le serveur RDAP que vous interrogez accepte cette nouvelle extension permettant d'avoir des réponses partielles, regardez si `subsetting` apparaît dans le tableau `rdapConformance` de la réponse du serveur (cette extension est désormais dans le registre IANA <<https://www.iana.org/assignments/rdap-extensions/rdap-extensions.xml>>). Voici l'exemple que donne le RFC :

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc8977.txt>

```

{
  "rdapConformance": [
    "rdap_level_0",
    "subsetting"
  ],
  ...
  "subsetting_metadata": {
    "currentFieldSet": "afieldset",
    "availableFieldSets": [
      {
        "name": "anotherfieldset",
        "description": "Contains some fields",
        "default": false,
        "links": [
          {
            "value": "https://example.com/rdap/domains?name=example*.com
              &fieldSet=afieldset",
            "rel": "alternate",
            "href": "https://example.com/rdap/domains?name=example*.com
              &fieldSet=anotherfieldset",
            "title": "Result Subset Link",
            "type": "application/rdap+json"
          }
        ]
      }
    ],
    ...
  ]
},
...
"domainSearchResults": [
  ...
]
}

```

Mais il y a encore plus simple que de regarder `subsetting_metadata` : notre RFC décrit, dans sa section 4, trois ensembles de champs standards qui, espérons-le, seront présents sur la plupart des serveurs RDAP :

- `id`, qui ne renvoie dans la réponse que les identificateurs des objets (`handle` pour les contacts, `unicodeName` pour les noms de domaine, etc).
- `brief`, un peu plus bavard, qui renvoie l'information minimale (telle qu'estimée par le serveur) sur les objets.
- `full`, qui renvoie la réponse complète (actuellement, avant le déploiement de ce RFC 8982, c'est la valeur par défaut).

Un exemple de réponse avec l'ensemble `id`, où on n'a que le nom (ASCII) des domaines :

```

{
  "rdapConformance": [
    "rdap_level_0",
    "subsetting"
  ],
  ...
  "domainSearchResults": [
    {
      "objectClassName": "domain",
      "ldhName": "example1.com",
      "links": [
        {
          "value": "https://example.com/rdap/domain/example1.com",
          "rel": "self",

```

```
    "href": "https://example.com/rdap/domain/example1.com",
    "type": "application/rdap+json"
  }
]
},
{
  "objectClassName": "domain",
  "ldhName": "example2.com",
  "links": [
    {
      "value": "https://example.com/rdap/domain/example2.com",
      "rel": "self",
      "href": "https://example.com/rdap/domain/example2.com",
      "type": "application/rdap+json"
    }
  ]
},
...
]
```

Quelles mises en œuvre sont disponibles ? Il en existe une chez `.it` mais elle n'est accessible sur leur serveur de test que si on est authentifié `<https://rdap.pubtest.nic.it/doc/README.html>`.

Comme RDAP, contrairement à whois, permet l'authentification des clients (RFC 7481), on pourra imaginer des cas où l'ensemble complet de la réponse ne sera envoyé qu'à certains utilisateurs (section 8).

L'annexe A du RFC revient sur un choix sur lequel je suis passé rapidement : le fait que le client spécifie le nom d'un ensemble pré-défini de champs, plutôt que la liste des champs qui l'intéressent, ce qui serait plus souple. Mais cela aurait été compliqué pour les structures JSON profondes (il faudrait une syntaxe riche et peu lisible), et cela aurait compliqué les autorisations : que faire si un client demande deux champs et qu'il n'est autorisé que pour un seul ? Outre ces questions génériques de tout protocole utilisant JSON, il y avait des problèmes spécifiques à RDAP, comme le fait que le contenu sur une entité peut être réparti dans plusieurs structures JSON (ah, le jCard...). Des solutions existent, comme le langage de requêtes CQL `<https://github.com/gregwhitaker/catnap/wiki/Catnap-Query-Language-Reference>` mais il a semblé trop coûteux pour un intérêt limité.