RFC 10037 : Registration Data Access Protocol (RDAP) Extension for DNS Time-to-Live (TTL) Values

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 29 août 2026

Date de publication du RFC : Août 2026

https://www.bortzmeyer.org/10037.html

Puisque le RFC 9803[1] étend le protocole d'avitaillement EPP pour ajouter des TTL spécifiques aux noms de domaine enregistrés, il était logique que le protocole d'interrogation RDAP permette d'obtenir cette information. C'est ce que permet l'extension normalisée dans ce nouveau RFC.

L'extension est simple, et le RFC court. RDAP permet d'obtenir des informations, structurées en JSON, sur divers objets enregistrés, comme les noms de domaine. Ces réponses (RFC 9083) peuvent inclure la valeur d'enregistrements DNS de type DS, NS, A ou AAAA mais, jusqu'à présent, pas leurs TTL (section 5 du RFC 9499, sur ce concept de TTL). Ce n'était pas trop grave tant que la quasi-totalité des registres imposaient le même TTL, choisi par eux, à toutes les données. Maintenant que le RFC 9803 normalise un moyen pour le client de choisir son TTL, cette limitation de RDAP était plus gênante.

Donc, concrètement, le serveur RDAP qui gère cette extension doit ajouter une propriété `ttl0_data` aux objets de type domaine (RFC 9083, section 5.3) et serveur de noms (RFC 9083, section 5.2). Cette propriété est un objet JSON comportant les membres `values` et (facultativement) `remarks`. `values` est un dictionnaire indexé par le type d'enregistrement, et dont les valeurs sont le TTL en secondes. C'est le TTL tel qu'enregistré dans la base de données du registre, et pas celui que verra une requête DNS auprès de votre résolveur <https://www.bortzmeyer.org/resolveur-dns.html>, celui-ci étant la durée restante dans la mémoire du résolveur. Voici l'exemple du RFC :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc9803.txt`

```
{
  "objectClassName": "domain",
  "rdapConformance": ["rdap_level_0", "ttl0"],
  "ldhName": "domain.example",
  "ttl0_data": {
    "values": {
      "NS": 3600,
      "DS": 300
    },
    "remarks": [
      {
        "description": [
          "For more information about the .example",
          " registry policy relating to DS record TTL changes,",
          "see https://domain.example/policy.html"
        ]
      }
    ]
  }
}
```

Et pour un serveur de noms <https://www.bortzmeyer.org/serveur-dns-faisant-autorite.html> enregistré (cf. RFC 5732) :

```
 {
  "objectClassName": "nameserver",
  "rdapConformance": ["rdap_level_0", "ttl0"],
  "ldhName": "ns1.domain.example",
  "ttl0_data": {
    "values": {
      "A": 86400,
      "AAAA": 86400
    },
    "remarks": [
      {
        "description": [
          "The .example registry does not permit TTL ",
          "values for nameservers to be changed."
        ]
      }
    ]
  }
}
```

Notez aussi le `ttl0` dans le tableau `rdapConformance` pour indiquer que le serveur RDAP connait cette extension.

L'extension est désormais enregistrée à l'IANA <https://www.iana.org/assignments/rdap-extensions/rdap-extensions.xml#rdap-extensions-1>. Quels logiciels la gèrent ? La bibliothèque RDAP de l'ICANN <https://github.com/icann/icann-rdap> ainsi que la bibliothèque Perl Net::RDAP <https://metacpan.org/pod/Net::RDAP> et son client rdapper <https://metacpan.org/pod/App::rdapper>.

Petit rappel important au passage : la section 5.2 du RFC 2181 impose que le TTL s'applique à un ensemble d'enregistrements ("*RRset*", pour "*Resource Record Set*") pas à un seul enregistrement. Ainsi, pour un ensemble NS, tous les enregistrements ont forcément le même TTL.