RFC 7071 : A Media Type for Reputation Interchange

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 22 novembre 2013

Date de publication du RFC : Novembre 2013

https://www.bortzmeyer.org/7071.html

Le cadre général d'accès à l'information sur la **réputation** d'une entité (identifiée par son nom de domaine, son adresse IP ou d'autres identificateurs) a été défini dans le RFC 7070 [1]. Cet autre RFC est plus concret et définit le format d'un **reputon**, une information structurée (en JSON) sur la réputation d'une entité, ainsi que le type de media associé, le nouveau `application/reputons+json`. Il crée également des registres IANA pour les noms d'applications de réputation et les réponses possibles selon l'application.

Le cadre du RFC 7070 prévoit que l'accès aux informations de réputation peut se faire par plusieurs mécanismes comme par exemple HTTP (RFC 7072). Mais le format transporté est toujours le même, celui d'un reputon. Un reputon est un objet JSON comportant les informations de réputation : identificateur de l'entité jugée, assertions sur cette identité, classement de l'identité selon ces assertions. Une requête va donc renvoyer un ou plusieurs reputons.

La section 3 liste les attributs d'un reputon, notamment :
— L'identité de l'entité ("*rater*") qui a jugé de la réputation,
— Celle de l'entité jugée ("*rated*"),
— L'assertion,
— Le classement ("*rating*") de l'entitée jugée, selon cette assertion, de 0,0 (assertion complètement fausse) à 1,0 (assertion tout à fait vraie).

Les identités dépendent de l'application et peuvent être des noms de domaine, des adresses IP, etc. La section 4 décrit en détail la notion de classement. Il y a aussi des attributs facultatifs dans un reputon :
— Degré de confiance du juge vis-à-vis de son classement, de 0,0 à 1,0,
— Classement considéré comme normal, pour pouvoir estimer si l'entité jugée est « dans les clous » ou pas,
— Taille de l'échantillon sur lequel s'est fondé le jugement,

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc7070.txt`

— Date et heure où le classement a été établi (”*generated*”),
— Date et heure à partir desquelles le jugement cessera d'être valable (”*expires*”).

Ces deux derniers attributs sont représentés par un nombre de secondes depuis le 1er janvier 1970. L'attribut `expires` permet (mais n'oblige pas) de mettre en place des caches devant le serveur qu'on interroge (section 5). Le RFC recommande de mettre des durées de validité d'autant plus courtes qu'on n'est pas sûr du jugement, par exemple parce qu'on n'a pas encore récolté beaucoup de données.

La section 6 indique la syntaxe concrète des reputons, sous forme d'un objet JSON, dont les membres (couples {clé, valeur}) représentent les attributs présentés plus haut. Les reputons sont mis dans un tableau JSON, puisqu'on peut en avoir plusieurs (s'il existe plusieurs assertions). Les assertions présentes dans un reputon sont typiquement celles qui ont été demandées par le client. Si ce dernier ne précise rien, le serveur peut renvoyer toutes les assertions qu'il connait. Un reputon peut être vide, si le serveur n'a aucune information sur l'entité et l'assertion demandées.

Voici l'exemple de reputon donné par le RFC, emprunté au baseball :

```
{
       "application": "baseball",
       "reputons": [
         {
           "rater": "RatingsRUs.example.com",
           "assertion": "is-good",
           "rated": "Alex Rodriguez",
           "rating": 0.99,
           "sample-size": 50000
         }
       ]
}
```

On a un seul reputon, pour l'assertion « est un bon joueur ». Vu le classement (quasiment 1), on peut dire que `RatingsRUs.example.com` estime qu'Alex Rodriguez est un bon joueur.

L'assertion est ici très générale. Elle pourrait être plus précise, si les concepteurs de l'application `baseball` le voulaient :

```
{
       "application": "baseball",
       "reputons:" [
         {
           "rater": "baseball-reference.example.com",
           "assertion": "hits-for-power",
           "rated": "Alex Rodriguez",
           "rating": 0.99,
           "sample-size": 50000
         },
         {
           "rater": "baseball-reference.example.com",
           "assertion": "strong-hitter",
           "rated": "Alex Rodriguez",
           "rating": 0.4,
           "confidence": 0.2,
           "sample-size": 50000
         }
       ]
}
```

Si vous ne connaissez pas le baseball et que vous vous demandez ce que veut dire `hits-for-power` (ne frappe pas forcément beaucoup mais fort) ou `strong-hitter` (frappe souvent des coups sûrs), voyez Wikipédia. On voit que le même Alex Rodriguez a une nettement moins bonne réputation pour `strong-hitter` que pour `hits-for-power`. Notez aussi l'indication de la taille de l'échantillon (ici, 50 000 points de mesure), qui permet aux statisticiens de se faire une idée de la validité de ces classements.

Un exemple moins artificiel de service de réputation serait évidemment lié à la lutte contre le spam. Prenons donc cette fois une application réelle, enregistrée dans le registre IANA des applications de réputation <https://www.iana.org/assignments/reputation-parameters/reputation-parameters.xhtml#applications>, l'application `email-id` définie dans le RFC 7073. Ce tableau contient deux reputons :

```
{
  "application": "email-id",
  "reputons": [
    {
      "rater": "rep.example.net",
      "assertion": "spam",
      "identity": "dkim",
      "rated": "example.com",
      "confidence": 0.95,
      "rating": 0.012,
      "sample-size": 16938213,
      "updated": 1317795852
    },
    {
      "rater": "rep.example.net",
      "assertion": "spam",
      "identity": "spf",
      "rated": "example.com",
      "confidence": 0.98,
      "rating": 0.023,
      "sample-size": 16938213,
      "updated": 1317795852
    }
  ]
}
```

Il se lit ainsi : « `example.com`, authentifié par DKIM (regardez l'attribut `identity`, spécifique à cette application) envoit du spam 1,2 % du temps. L'échantillon compte près de 17 millions de messages ». À noter que le second reputon, basé sur SPF, indique presque deux fois plus de spam. Cela peut vouloir dire que la liste des serveurs SMTP autorisés par SPF comprend quelques moutons noirs.

Ces reputons seront étiquetés avec le nouveau type `application/reputons+json` désormais enregistré à l'IANA <https://www.iana.org/assignments/media-types/application/reputon+json>. Il utilise les suffixes (`+json`) du RFC 6839. Est également enregistrée à l'IANA la liste des applications <https://www.iana.org/assignments/reputation-parameters/reputation-parameters.xhtml#applications>. Les nouvelles applications qui voudraient être incluses dans ce registre doivent avoir un examen par l'IETF ou bien une spécification stable (le RFC 5226 contient la liste des règles d'enregistrement à l'IANA).

Vous voulez voir des vrais reputons ? Il n'y a pas encore beaucoup de services disponibles publiquement. Essayons avec celui d'OpenDKIM <http://www.opendkim.org/> :

```
% curl 'http://repute.opendkim.org/repute.php?subject=ietf.org&assertion=spam&application=email-id&service=r
Content-Type: application/reputon+json

{
  "application": "email-id",
  "reputons": [
    {
"rater": "repute.opendkim.org",
"assertion": "spam",
"rated": "ietf.org",
"rating": 0,
"identity": "dkim",
"rate": 4,
"sample-size": 2,
"generated": 1338014959
    }
  ]
}
```

Très bon score pour `ietf.org`, une spamicité nulle. Mais faites attention à la taille de l'échantillon, seuls deux messages ont été examinés...

Merci à Vincent Levigneron pour ses explications sur le baseball (les erreurs qui restent sont les miennes, je n'ai pas forcément tout compris).