

RFC 9581 : Concise Binary Object Representation (CBOR) Tags for Time, Duration, and Period

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 17 août 2024

Date de publication du RFC : Août 2024

<https://www.bortzmeyer.org/9581.html>

Ce RFC ajoute au format de données binaire CBOR la possibilité de stocker des données temporelles plus détaillées, incluant par exemple l'échelle utilisée (UTC ou TAI), ou bien ayant une précision supérieure.

Petit rappel : CBOR est normalisé dans le RFC 8949¹, il incluait deux étiquettes pour les données temporelles, permettant d'indiquer date et heure en format lisible par un humain, et sous forme d'un nombre de secondes depuis l'"epoch", avec une résolution d'une seconde. Le RFC 8943 y ajoute les dates (sans indication de l'heure). Au passage, le concept d'étiquette en CBOR est normalisé dans la section 3.4 du RFC 8949.

Notre nouveau RFC 9581 ajoute :

- Une étiquette, 1001, pour un temps étendu, sous forme d'un dictionnaire CBOR. La clé 1 doit être présente, pour indiquer le temps de base, des clés négatives servent pour indiquer des fractions de seconde, la clé -1 sert pour l'échelle (UTC ou TAI, notamment).
- Une étiquette, 1002, pour indiquer une durée. C'est également un dictionnaire CBOR.
- Une étiquette, 1003, pour indiquer une période, représentée par une date de début et une date de fin.

Ces nouvelles étiquettes ont été ajoutées au registre des étiquettes CBOR <<https://www.iana.org/assignments/cbor-tags/cbor-tags.xml#tags>>. Les clés possibles pour les dictionnaires indiquant les temps étendus sont dans un nouveau registre à l'IANA <<https://www.iana.org/assignments/cbor-tags/cbor-tags.xml#time-tag-map-keys>>. Ajouter une entrée à ce registre nécessite un RFC et un examen par un expert.

Un service sur ce blog, permet d'obtenir plusieurs valeurs utilisant ces formats. Si nous utilisons le gem Ruby `cbor-diag` <<https://rubygems.org/gems/cbor-diag>>, nous voyons :

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc8949.txt>

```
% curl -s https://www.bortzmeyer.org/apps/date-in-cbor | cbor2diag.rb
["Current date in CBOR, done with Python 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0] and the flunn lib
  0("2024-04-17T14:21:07Z"),
  1(1713363667),
  100(19830),
  1004("2024-04-17"),
  1001({1: 1713363667, -1: 0, -9: 193986759}),
  1001({1: 1713363704, -1: 1}),
  "Duration since boot: ",
  1002({1: 1742903})]
```

On voit alors successivement :

- Les deux valeurs étiquetées 0 et 1 du RFC 8949 et les deux valeurs étiquetées 100 et 1004 du RFC 8943.
- Puis les nouveautés de notre RFC.
- D'abord, un temps étendu étiqueté 1001. Le dictionnaire a trois éléments, celui de clé 1 est la date et heure, celui de clé -1 indique qu'il s'agit d'un temps UTC, celui de clé -9 indique qu'il s'agit de nanosecondes (-3 : millisecondes, -6 : microsecondes, etc). On a ainsi une meilleure résolution.
- Un autre temps étendu est en TAI (la clé -1 a la valeur 1) et vous voyez qu'il est situé 37 secondes dans le futur (l'actuel décalage entre UTC et TAI).
- Une durée, qui est le temps écoulé (en secondes) depuis le démarrage de la machine.
- On n'indique pas des informations comme la qualité de l'horloge car, franchement, je n'ai pas vraiment compris comment elle était représentée.

Question programmation, le service a été écrit en Python. Ce langage a une fonction `time_ns`, qui permet d'obtenir les nanosecondes. Pour le TAI, cela a été un peu plus difficile, la machine doit être configurée spécialement <<https://www.bortzmeyer.org/tai-on-debian.html>>.