RFC 9209 : The Proxy-Status HTTP Response Header Field

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 14 juin 2022

Date de publication du RFC : Juin 2022

https://www.bortzmeyer.org/9209.html

Le protocole HTTP, qui est à la base du Web, n'est pas forcément de bout en bout, entre client et serveur. Il y a souvent passage par un relais et ce relais a parfois des choses à signaler au client HTTP, notamment en cas d'erreur. Ce RFC spécifie le champ d'en-tête `Proxy-Status` pour cela.

La norme HTTP, le RFC 9110 [1] décrit ces relais, ces intermédiaires, dans sa section 3.7. On en trouve fréquemment sur le Web. Il y a depuis longtemps des codes d'erreur pour eux, comme 502 si le serveur d'origine répond mal et 504 pour les cas où il ne répond pas du tout. Mais ce n'est pas forcément assez précis, d'où le nouveau champ dans l'en-tête (ou dans le pied). Il utilise la syntaxe des champs structurés du RFC 9651. Voici un exemple :

```
Proxy-Status: proxy.example.net; error="http_protocol_error"; details="Malformed response header: space before c
  "Example CDN"
```

Cet exemple se lit ainsi : le premier (en partant du serveur d'origine) s'identifie comme `proxy.example.net` et il signale que le serveur d'origine n'avait pas bien lu le RFC 9112. Puis la réponse est passée par un autre intermédiaire, qui s'identifie comme `"Example CDN"` (l'identificateur n'est pas forcément un nom de domaine), et n'a rien de particulier à raconter. Le champ `Proxy-Status` est désormais dans le registres des champs d'en-tête (ou de pied) <https://www.iana.org/assignments/http-fields/http-fields.xml#field-names>.

Vous avez vu dans l'exemple ci-dessus le paramètre `error`. Il peut s'utiliser, par exemple, avec le code de retour 504 :

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc9110.txt`

```
HTTP/1.1 504 Gateway Timeout
Proxy-Status: foobar.example.net; error=connection_timeout
```

Ici, le relais `foobar.example.net` n'a eu aucune réponse du serveur d'origine (ou, plus rigoureusement, du serveur qu'il essayait de contacter, qui peut être un autre intermédiaire).

Mais il existe d'autres paramètres possibles <https://www.iana.org/assignments/http-proxy-status/http-proxy-status.xml#http-proxy-status-parameters>, comme :

- `next-hop` : nom ou adresse du serveur contacté, par exemple `Proxy-Status: cdn.example.org; next-hop=backend.example.org:8001`.
- `next-protocol` : protocole (ALPN, RFC 7301) utilisé avec le serveur, par exemple `Proxy-Status: "proxy.example.org"; next-protocol=h2` pour du HTTP/2 (RFC 9113).
- `received-status` : le code de retour du serveur, comme dans `Proxy-Status: ExampleCDN; received-status=200`, pour un cas où tout s'est bien passé.

Et on peut définir de nouveaux paramètres par la procédure d'examen par un expert (RFC 8126).

Le paramètre `error` prend comme valeur un type d'erreur. Il en existe plusieurs <https://www.iana.org/assignments/http-proxy-status/http-proxy-status.xml#http-proxy-error-types>, chacun avec un code de retour recommandé dont (je ne les indique pas tous, ils sont très nombreux !) :

- `dns_timeout` (pour le code de retour 504) et `dns_error` (code 502) : c'est la faute du DNS. Le second type permet en outre d'indiquer le paramètre `rcode` (code de retour DNS, comme `NXDOMAIN`) et le paramètre `info-code`, le code étendu du RFC 8914.
- `connection_timeout` ou `connection_refused`.
- `destination_ip_prohibited` : le pare-feu ne veut pas.
- `tls_protocol_error` : là, c'est TLS qui ne veut pas.
- `tls_certificate_error` : certificat du serveur problématique, par exemple expiré <https://www.bortzmeyer.org/tester-expiration-certifs.html>.
- `http_protocol_error` : la réponse du serveur n'était pas du bon HTTP.
- `proxy_internal_error` : le relais a un problème interne.

Là aussi, on peut enregistrer de nouveaux types avec la procédure d'examen par un expert du RFC 8126.

La section 4 du RFC détaille les questions de sécurité. Comme toute information, `Proxy-Status` peut aider un attaquant, par exemple en lui donnant des idées sur comment joindre directement un intermédiaire. Pour cette raison, les logiciels doivent fournir un moyen de contrôler l'ajout (ou pas) de `Proxy-Status`, qu'on peut aussi n'inclure que dans certains cas. Notez aussi qu'un intermédiaire peut mentir (ou se tromper) et que le `Proxy-Status` ne vaut donc que ce que vaut l'intermédiaire.

Je n'ai pas de liste des logiciels qui mettent en œuvre ce champ `Proxy-Status`.