

RFC 8297 : An HTTP Status Code for Indicating Hints

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 20 décembre 2017

Date de publication du RFC : Décembre 2017

<https://www.bortzmeyer.org/8297.html>

Lorsqu'un serveur HTTP répond, la réponse contient souvent des **liens** vers d'autres ressources. Un exemple typique est celui de la page Web dont le chargement va déclencher le chargement de feuilles de style, de JavaScript, etc. Pour minimiser la latence, il serait intéressant de prévenir le client le plus tôt possible. C'est le but de ce RFC, qui décrit le code de retour intérimaire 103, qui prévient le client qu'il peut tout de suite commencer à charger ces ressources supplémentaires.

Il existe un type de lien pour cela, `preload`, décrit par ses auteurs <<https://w3c.github.io/preload/>> et enregistré dans le registre des types de liens <<https://www.iana.org/assignments/link-relations/link-relations.xml>> (cf. RFC 8288¹). Il peut être utilisé dans la réponse « normale » :

```
HTTP/1.1 200 OK
Date: Fri, 26 May 2017 10:02:11 GMT
Content-Length: 1234
Content-Type: text/html; charset=utf-8
Link: </main.css>; rel="preload"; as="style"
Link: </script.js>; rel="preload"; as="script"
```

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc8288.txt>

Mais cela ne fait pas gagner grand'chose : une toute petite fraction de seconde après, le client HTTP verra arriver le source HTML et pourra y découvrir les liens. On voudrait renvoyer tout de suite la liste des ressources à charger, sans attendre que le serveur ait fini de calculer la réponse (ce qui peut prendre du temps, s'il faut dérouler mille lignes de Java et plein de requêtes SQL...)

Le nouveau code de retour, 103, lui, peut être envoyé immédiatement, avec la liste des ressources. Le client peut alors les charger, tout en attendant le code de retour 200 qui indiquera que la ressource principale est prête. (Les codes de retour commençant par 1, comme 103, sont des réponses temporaires, « pour information », en attendant le succès, annoncé par un code commençant par 2. Cf. RFC 7231, sections 6.2 et 6.3.) La réponse HTTP utilisant le nouveau code ressemblera à :

```
HTTP/1.1 103 Early Hints
Link: </main.css>; rel="preload"; as="style"
Link: </script.js>; rel="preload"; as="script"

HTTP/1.1 200 OK
Date: Fri, 26 May 2017 10:02:11 GMT
Content-Length: 1234
Content-Type: text/html; charset=utf-8
Link: </main.css>; rel="preload"; as="style"
Link: </script.js>; rel="preload"; as="script"
```

Les détails, maintenant (section 2 du RFC). 103 indique au client qu'il y aura une série de liens vers des ressources supplémentaires qu'il peut être intéressant, par exemple, de charger tout de suite. Les liens finaux seront **peut-être différents** (dans l'exemple ci-dessus, ils sont identiques). 103 est juste une optimisation, pas une obligation. ("*Hint*" = suggestion.) Les liens qu'il indique ne font pas autorité. Le serveur peut indiquer des liens supplémentaires, ne pas indiquer des liens qui étaient dans la réponse 103, indiquer des liens différents, etc.

Il peut même y avoir plusieurs 103 à la suite, notamment si un relais sur le trajet ajoute le sien, par exemple en se basant sur une réponse qu'il avait gardée en mémoire. 103 n'est en effet pas toujours envoyé par le serveur d'origine de la ressource, il peut l'être par un intermédiaire. Voici un exemple qui donne une idée des variantes possibles :

```
HTTP/1.1 103 Early Hints
Link: </main.css>; rel="preload"; as="style"

HTTP/1.1 103 Early Hints
Link: </style.css>; rel="preload"; as="style"
Link: </script.js>; rel="preload"; as="script"

HTTP/1.1 200 OK
Date: Fri, 26 May 2017 10:02:11 GMT
Content-Length: 1234
Content-Type: text/html; charset=utf-8
Link: </main.css>; rel="preload"; as="style"
Link: </newstyle.css>; rel="preload"; as="style"
Link: </script.js>; rel="preload"; as="script"
```

On voit que la réponse finale n'est ni la première suggestion, ni la deuxième (ni une union des deux).

Note pour les programmeurs Python/WSGI. Je ne suis pas arrivé à utiliser ce code « intérimaire » avec WSGI, cela ne semble pas possible en WSGI <<https://stackoverflow.com/questions/47775640/returning-a-http-1xx-code-before-the-200-one-with-wsgi>>. Mais on trouvera sans doute d'autres implémentations...

Le code 103 est désormais dans le registre IANA des codes de retour <<https://www.iana.org/assignments/http-status-codes/http-status-codes.xml>>.