RFC 8246: **HTTP Immutable Responses**

Date de publication du RFC : Septembre 2017
Auteur(s) du RFC : P. McManus (Mozilla)
Chemin des normes
Réalisé dans le cadre du groupe de travail IETF httpbis [https://tools.ietf.org/wg/httpbis]
Première rédaction de cet article le 16 septembre 2017

Lorsqu'un serveur HTTP renvoie une réponse à un client, il peut indiquer une durée de vie maximale de la rssource transmise, avec l'en-tête `Cache-Control:` (RFC 7234). Mais cela ne donne qu'une durée **maximale**. La ressource peut quand même être modifiée avant la fin de cette période. Un client HTTP prudent va donc **revalider** la fraîcheur de cette ressource de temps en temps. L'extension décrite dans ce RFC permet de se dispenser de cette revalidation, en indiquant une durée **minimale**, pendant laquelle on est sûrs et certains que la ressource ne sera pas modifiée.

Le RFC prend l'exemple (section 1) d'un journal dont la page d'accueil indique une durée de vie maximale d'une heure :

```
Cache-Control: max-age=3600
```

Le client HTTP qui reçoit cet en-tête dans la réponse sait qu'il doit revenir vers le serveur au bout d'une heure. Mais la page sera peut-être modifiée (par exemple en raison d'un évènement soudain et imprévu) avant qu'une heure soit écoulée. Le client est donc partagé entre la revalidation (qui risque de consommer des ressources pour rien, même si elle est moins coûteuse qu'un téléchargement complet, grâce aux trucs du RFC 7232) et le risque de servir à son utilisateur une ressource désormais plus à jour. Le problème est d'autant plus ennuyeux que la revalidation d'une bête page Web qui comprend des photos peut générer beaucoup de requêtes [http://httparchive.org/interesting.php#reqTotal] , dont la plupart produiront sans doute un 304 (ressource non modifiée, la revalidation était inutile).

Notez que dans certains cas, le contenu pointé par un URL peut changer, mais dans d'autres cas, il est réellement immuable. Par exemple, si on publie chaque version de sa feuille de style sous un URL différent, la CSS n'a pas besoin d'être revalidée.

Bref, il y avait un besoin de pouvoir indiquer l'immuabilité d'une ressource. C'est désormais fait (section 2 de ce RFC) avec une extension à `Cache-Control:` :

```
Cache-Control: max-age=3600, immutable
```

Avec le mot-clé `immutable`, le serveur indique que la ressource ne sera pas modifiée pendant la durée de vie indiquée, le client peut donc se dispenser de vérifier. (« Client » ici désignant aussi bien le client final, par exemple le navigateur Web, qu'un intermédiaire.)

Par exemple, comme les RFC sont immuables (on ne les change jamais même d'une virgule, même en cas d'erreur [7158.html]), la ressource qui désigne ce RFC, `https://www.rfc-editor.org/rfc/rfc8246.txt [https://www.rfc-editor.org/rfc/rfc8246.txt] ,` pourrait parfaitement renvoyer cet en-tête (elle ne le fait pas). Cela pourrait être :

```
Last-Modified: Thu, 14 Sep 2017 23:11:35 GMT
Cache-Control: max-age=315576000, immutable
```

(Oui, dix ans…)

Voilà, c'est tout, la directive a été ajoutée au registre IANA [https://www.iana.org/assignments/http-cache-directives/http-cache-directives.xml] . Mais la section 3 du RFC se penche encore sur quelques questions de sécurité. Indiquer qu'une ressource est immuable revient à la fixer pour un temps potentiellement très long, et cela peut donc servir à pérenniser une attaque. Si un méchant pirate un site Web, et sert son contenu piraté avec un en-tête d'immuabilité, il restera bien plus longtemps dans les caches. Pire, sans HTTPS, l'en-tête avec `immutable` pourrait être ajouté par un intermédiaire (le RFC déconseille donc de tenir compte de cette option si on n'a pas utilisé HTTPS).

Les navigateurs Web ont souvent deux options pour recharger une page, « douce » et « dure » (F5 et Contrôle-F5 dans Firefox). Le RFC conseille que, dans le second cas (rechargement forcé), le `immutable` soit ignoré (afin de pouvoir recharger une page invalide).

Enfin, toujours question sécurité, le RFC recommande aux clients HTTP de ne tenir compte de l'en-tête d'immuabilité que s'ils sont sûrs que la ressource a été transmise proprement (taille correspondant à `Content-Length:` par exemple).

Si vous voulez comparer deux ressources avec et sans `immutable`, regardez `http://www.bortzmeyer.org/files/maybemodified.txt [http://www.bortzmeyer.org/files/maybemodified.txt]` (sans `immutable`) et `http://www.bortzmeyer.org/files/forever.txt [http://www.bortzmeyer.org/files/forever.txt]` (avec). Si le client HTTP gère l'extension d'immuabilité, un rechargement « dou » ne fera pas de requête HTTP. Sinon, il y aura revalidation et le serveur HTTP renverra un 304 :

```
[2001:db8:abcd:1234:acd8:9bd0:27c9:3a7f]:45452 - - [16/Sep/2017:17:53:29 +0200] "GET /files/forever.txt HTTP/1.1" 304 - "-" "Mozilla/5.0 (X11; Linux i686; rv:52.0)
```

Apparemment, Firefox gère cette extension mais, ici, le Firefox était sans doute trop vieux (normalement, cela aurait dû arriver avec la version 49 [https://bugzilla.mozilla.org/show_bug.cgi?id=1267474] , puisque Mozilla était premier intéressé [https://bitsup.blogspot.fr/2016/05/cache-control-immutable.html]). Les auteurs de Squid ont annoncé qu'ils géreraient cette extension. Côté serveur, notons que Facebook envoie déjà cette extension pour, par exemple, le code JavaScript (qui est versionné et donc jamais changé) :

```
% curl -v https://www.facebook.com/rsrc.php/v3iCKe4/y2/l/fr_FR/KzpL-Ycd5sN.js
* Connected to www.facebook.com (2a03:2880:f112:83:face:b00c:0:25de) port 443 (#0)
...
< HTTP/2 200
< content-type: application/x-javascript; charset=utf-8
< cache-control: public,max-age=31536000,immutable
...
```

Idem pour une mise en œuvre [https://github.com/ipfs/go-ipfs/pull/2672] d'IPFS.

<stephane+blog@bortzmeyer.org>. Je suis les règles de Crocker [http://sl4.org/crocker.html] donc pas besoin de faire des excès de diplomatie. Ce blog est strictement personnel et les opinions exprimées ici n'engagent donc que moi, et notamment pas mon employeur présent ou mes employeurs passés ou mes éventuels employeurs futurs.

[http://prefetch.validatorsearch.verisignlabs.com]