RFC 8674 : The "safe" HTTP Preference

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 5 décembre 2019

Date de publication du RFC : Décembre 2019

https://www.bortzmeyer.org/8674.html

Ce nouveau RFC définit une nouvelle préférence qu'un client HTTP peut envoyer au serveur. « safe » (sûr) indique que le client ne souhaite pas recevoir du contenu qu'il trouve contestable.

Je vous arrête tout de suite : vous allez me demander « mais qui définit du contenu contestable ? Ça dépend des gens » et, je vous rassure, l'auteur du RFC a bien vu le problème. La préférence safe est juste une possibilité technique, le RFC ne définit pas ce qui est sûr et ce qui ne l'est pas, cela devra se faire dans un autre cadre, une discussion à ce sujet serait bien trop casse-gueule pour l'IETF. En pratique, le résultat de l'utilisation de cette préférence dépendra de bien des choses, comme la politique du serveur (le RFC dit « "*the cultural context of the site*" »), et une éventuelle relation pré-existante entre le serveur et un utilisateur particulier. Le RFC donne quand même une indication : safe peut vouloir dire « adapté aux mineurs ».

Il y a manifestement une demande, puisque bien des sites Web ont un mode « sûr », où on peut sélectionner « je ne veux pas voir des choses que je n'aime pas ». Notez que, dans ces cas, la définition de ce qui est sûr ou pas dépend du site Web. S'ils est géré aux États-Unis, « sûr » sera sans doute « aucune nudité humaine », en Arabie saoudite, « aucune femme visible », etc. Ce mode « sûr » des sites Web n'est pas pratique pour l'utilisateurice, car il nécessite de sélectionner l'option pour chaque site, et de se créer un compte, soit explicite, soit implicite via les "*cookies*" (RFC 6265 [1]). À moins que le mode « sûr » soit le mode par défaut et, dans ce cas, ce sont les gens qui n'en voudront pas qui auront du travail. D'où l'idée, très controversée à l'IETF, de configurer cela dans le navigateur Web (ou bien dans le système d'exploitation, pour que tous les clients HTTP le fassent), qui va indiquer au serveur les préférences (un peu comme le "*Do Not Track*", dont on sait qu'il est largement ignoré par les sites Web). La technique utilisée est celle des préférences HTTP, normalisées dans le RFC 7240, préférences dont je rappelle que leur respect par le serveur est optionnel. La préférence safe envoyée par le client est donc à prendre comme un appel à « faire au mieux » et « je te fais confiance pour la définition de "sûr" », pas plus.

La section 2 de notre RFC est plus concrète, présentant la syntaxe exacte de safe. Voici un exemple de requête HTTP exprimant cette préférence :

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc6265.txt

```
GET /foo.html HTTP/1.1
Host: www.example.org
User-Agent: ExampleBrowser/1.0
Prefer: safe
```

La préférence est enregistrée à l'IANA <https://www.iana.org/assignments/http-parameters/http-parameters.xml#preferences>. Le RFC impose que les requêtes « sûres » soient faites en HTTPS, pour éviter la surveillance spécifique des gens qui demandent du `safe` (et qui peuvent être des enfants), et pour éviter qu'un intermédiaire ne bricole la requête, ajoutant ou enlevant cette préférence. Une réponse possible à la requête ci-dessus serait :

```
HTTP/1.1 200 OK
Transfer-Encoding: chunked
Content-Type: text/html
Preference-Applied: safe
Server: ExampleServer/2.0
Vary: Prefer
```

Le `Vary:` (RFC 7231, section 7.1.4) indique aux relais/caches intermédiaires qu'ils doivent tenir compte de la valeur de `Prefer:` avant de renvoyer la page mémorisée à un autre client.

Si vous voulez tester en vrai, la page vous renverra un contenu différent selon que vous envoyez l'en-tête `Prefer: safe` ou pas. Voici un exemple avec curl :

```
%  curl --header "Prefer: safe" https://www.bortzmeyer.org/apps/porn
```

Le code est du Python/WSGI et se résume à :

```
def porn(start_response, environ):
    # Apache/WSGI always give us one Prefer: header even if the client sent several.
    preferences = re.split("\s*,\s*", environ['HTTP_PREFER'])
    safe = False
    for pref in preferences:
        if pref.lower() == 'safe':
            safe = True
            break
    begin = """<html><head>..."""
    end = """</body></html>"""
    safe_content = """<p>Safe ..."""
    unsafe_content = """<p>Unsafe ..."""
    if safe:
        output = begin + safe_content + end
    else:
        output = begin + unsafe_content + end
    status = '200 OK'
    response_headers = [('Content-type', 'text/html'),
                        ('Content-Length', str(len(output)))]
    start_response(status, response_headers)
    return [output]
```

Cette préférence semble répondre à une forte demande, puisqu'elle est déjà reconnue :

— Dans Internet Explorer (cf. la documentation <https://support.microsoft.com/en-hk/help/2980016/update-to-support-a-new-safe-preference-feature-in-internet-explorer-1>
— Dans Bing (cf. Bing <https://developer.microsoft.com/en-us/microsoft-edge/testdrive/demos/familysearch/>),
— Dans les boitiers Cisco (cf. la documentation) <https://blogs.cisco.com/security/filtering-explicit

La section 3 du RFC rassemble quelques informations de sécurité :

— HTTPS est indispensable, sinon un intermédiaire pourra ajouter (ou retirer) la préférence `safe`,
— Paradoxalement, `safe` peut être dangereux puisqu'il transmet une information supplémentaire au serveur, aidant au "*fingerprinting*",
— On n'a évidemment aucune garantie que le serveur a bien adapté le niveau du contenu à ce qu'on voulait. Le RFC fait même remarquer qu'un serveur sadique pourrait envoyer du contenu « pire » lorsque la préférence `safe` est présente.

Enfin, l'annexe A donne quelques conseils aux auteurs de navigateurs quant à la mise en œuvre de cette préférence. L'UI n'est pas évidente. Il est crucial de ne pas donner à l'utilisateur ou l'utilisatrice l'impression que cette préférence fournirait des garanties. Le RFC suggère un texte plus prudent pour une case à cocher « Demander du contenu "sûr" aux sites Web ». Et l'annexe B a des conseils pour les gérants de sites Web comme, par exemple, ne pas permettre aux utilisateurs de demander (via leur profil, par exemple) d'ignorer la préférence `safe` puisqu'elle a pu être placée par un logiciel de contrôle parental.

Ce RFC a le statut « pour information » et a été publié sur la voie indépendante (cf. RFC 5742) puisqu'il n'a pas fait l'objet d'un consensus à l'IETF (euphémisme…) Les principales objections <https://datatracker.ietf.org/doc/conflict-review-nottingham-safe-hint/ballot/> étaient :

— « Sûr » n'a pas de définition précise et universelle, le terme est vraiment trop vague,
— Les navigateurs Web envoient déjà beaucoup trop d'informations aux serveurs, en ajouter une, qui pourrait, par exemple, permettre de cibler les mineurs, n'est pas forcément une bonne idée.