

Les types de média, c'est quoi, et à quoi sert l'haptique ? (1/18)

Stéphane Bortzmeyer
`stephane+cdl@bortzmeyer.org`

Capitole du Libre, 16 novembre 2025

On trouve de tout sur l'Internet

On trouve de tout sur l'Internet

- Du texte, des images, des films, des odeurs (non, ça, pas encore)...

On trouve de tout sur l'Internet

- Du texte, des images, des films, des odeurs (non, ça, pas encore)...
- Même pour le même type de contenu, il y a plusieurs formats (GIF, JPEG, PNG, WebP...)

On trouve de tout sur l'Internet

- Du texte, des images, des films, des odeurs (non, ça, pas encore)...
- Même pour le même type de contenu, il y a plusieurs formats (GIF, JPEG, PNG, WebP...)
- Des logiciels savent gérer différents types de contenus et différents formats, ou bien le faire suivre à un logiciel plus adapté.

On trouve de tout sur l'Internet

- Du texte, des images, des films, des odeurs (non, ça, pas encore)...
- Même pour le même type de contenu, il y a plusieurs formats (GIF, JPEG, PNG, WebP...)
- Des logiciels savent gérer différents types de contenus et différents formats, ou bien le faire suivre à un logiciel plus adapté.
- Mais comment connaître le format d'une ressource qu'on vient de récupérer ?

Les mauvaises méthodes pour identifier le format

Les mauvaises méthodes pour identifier le format

- Se fier à l'extension (mais un `.php` identifie le producteur, pas la production),

Les mauvaises méthodes pour identifier le format

- Se fier à l'extension,
- Se fier au nombre magique au début du fichier (mais tous les formats ne l'ont pas),

Les mauvaises méthodes pour identifier le format

- Se fier à l'extension,
- Se fier au nombre magique au début du fichier,
- Combiner diverses heuristiques en examinant le contenu (fragile. . .).

Les mauvaises méthodes pour identifier le format

- Se fier à l'extension,
- Se fier au nombre magique au début du fichier,
- Combiner diverses heuristiques en examinant le contenu.
- Solution extrême, rare aujourd'hui mais qui a été courante : n'accepter qu'un format (le courrier, il y a très longtemps).

Exemple de nombre magique

00000000:	4749 4638 3961 6e00 2c00 b300 00ff ffff	GIF89a.,.....
00000010:	e7e7 e7de dede d6d6 d6bd bdbd 8484 8442	B
00000020:	4242 3939 3931 3131 2929 2918 1818 1010	BB999111)))....
00000030:	1008 0808 ffde 0000 63a5 0000 0021 f904	c....!..

La bonne méthode

La bonne méthode

- Le serveur indique le **type de média** (dit aussi type MIME pour des raisons historiques).

La bonne méthode

- Le serveur indique le **type de média** (dit aussi type MIME pour des raisons historiques).
- Vous avez sans doute déjà vu `text/plain`, `image/png`, `application/octet-stream`, `video/webm` ou `audio/ogg`.

La bonne méthode

- Le serveur indique le **type de média** (dit aussi type MIME pour des raisons historiques).
- Vous avez sans doute déjà vu `text/plain`, `image/png`, `application/octet-stream`, `video/webm` ou `audio/ogg`.
- Un **type** et un **sous-type**, séparés par une barre oblique.

La bonne méthode

- Le serveur indique le **type de média** (dit aussi type MIME pour des raisons historiques).
- Vous avez sans doute déjà vu `text/plain`, `image/png`, `application/octet-stream`, `video/webm` ou `audio/ogg`.
- Un **type** et un **sous-type**, séparés par une barre oblique.
- Utilisé par le courrier électronique, le Web, Gemini et plein d'autres systèmes.

Le type de média vu par curl

```
% curl -i -s https://http.pizza/static/codes/pizza/large/200.avif
HTTP/2 200
...
content-type: image/avif
```

Le type de média vu par Firefox

Informations sur la page

 <u>G</u> énéral	 <u>M</u> édias	 <u>P</u> ermissions	 <u>S</u> écurité
--	--	---	--

Titre :	200.avif (Image AVIF, 1600 × 1400 pixels) - Redimensionnée (66%)
Adresse (URL) :	https://http.pizza/static/codes/pizza/large/200.avif
Type :	image/avif

Et dans Gemini

 gemini://gemini.bortzmeyer.org/software/lupa/stats.gmi

Most common media (MIME) types

- text/gemini: 411,582 URLs
- text/plain: 36,935 URLs
- image/jpeg: 36,605 URLs
- image/png: 22,711 URLs
- application/octet-stream: 12,222 URLs
- application/zip: 11,877 URLs
- image/webp: 7,896 URLs
- application/pdf: 5,749 URLs
- text/x-go: 5,177 URLs
- image/gif: 2,499 URLs
- octet/stream: 1,955 URLs
- application/x-mscardfile: 1,928 URLs
- text/html: 1,529 URLs
- audio/mpeg: 1,217 URLs
- text/markdown: 968 URLs

Types et sous-types

Types et sous-types

- Le type indique une catégorie très générale : texte, son, image. . .

Types et sous-types

- Le type indique une catégorie très générale : texte, son, image. . .
- Il n'y a que 11 types normalisés et les ajouts sont rares.

Types et sous-types

- Le type indique une catégorie très générale : texte, son, image. . .
- Il n'y a que 11 types normalisés et les ajouts sont rares.
- Le sous-type indique un format précis.

Types et sous-types

- Le type indique une catégorie très générale : texte, son, image. . .
- Il n'y a que 11 types normalisés et les ajouts sont rares.
- Le sous-type indique un format précis.
- Innombrables sous-types enregistrés et des nouveaux apparaissent tout le temps. Aucun logiciel ne peut les connaître tous.

L'intérêt de la distinction type/sous-type

L'intérêt de la distinction type/sous-type

- L'idée est qu'un logiciel qui ne connaît pas le sous-type peut quand même tenter de traiter selon le type seul.

L'intérêt de la distinction type/sous-type

- L'idée est qu'un logiciel qui ne connaît pas le sous-type peut quand même tenter de traiter selon le type seul.
- Ça marche bien pour `text/`, mais moins bien pour `image/` (on peut toujours lancer un visualisateur d'images, on verra bien) et pas du tout pour `application/`.

Le type text/

Le type text/

- Pour du texte qu'on peut éditer avec un éditeur quelconque (mais il faut évidemment utiliser emacs, vi est nul).

Le type text/

- Pour du texte qu'on peut éditer avec un éditeur quelconque.
- Et lire avec cat, more, less. . .

Le type text/

- Pour du texte qu'on peut éditer avec un éditeur quelconque.
- Et lire avec cat, more, less. . .
- Le HTML est-il du texte ? (Oui, le type officiel a toujours été `text/html`.)

Le type text/

- Pour du texte qu'on peut éditer avec un éditeur quelconque.
- Et lire avec cat, more, less. . .
- Le HTML est-il du texte ? (Oui, le type officiel a toujours été text/html.)
- Le même débat a eu lieu pour les langages de programmation (text/javascript remplaçant application/javascript, RFC 9239).

Le type application

Le type application

- Le reste. . .

Le type application

- Le reste. . .
- Ce type est prévu pour du binaire.

Le type application

- Le reste...
- Ce type est prévu pour du binaire.
- `application/octet-stream` : données binaires quelconques.

Le type application

- Le reste...
- Ce type est prévu pour du binaire.
- `application/octet-stream` : données binaires quelconques.
- Mais aussi des formats standards comme `application/tzif` ou `application/pdf`.

Mais qui enregistre les types et les sous-types ?

Mais qui enregistre les types et les sous-types ?

- L'enregistrement officiel est à l'IANA (*Internet Assigned Numbers Authority*, cf. <https://www.iana.org/assignments/media-types/media-types.xml>)

Mais qui enregistre les types et les sous-types ?

- L'enregistrement officiel est à l'IANA
- Textes à lire : les RFC 6838 "*Media Type Specifications and Registration Procedures*" et 4289 "*Multipurpose Internet Mail Extensions (MIME) Part Four : Registration Procedures*".

Mais qui enregistre les types et les sous-types ?

- L'enregistrement officiel est à l'IANA
- Textes à lire : les RFC 6838 et 4289.
- Pour un type, il faut écrire un RFC et qui ait le statut de norme (voir RFC 9694 "*Guidelines for the Definition of New Top-Level Media Types*").

Mais qui enregistre les types et les sous-types ?

- L'enregistrement officiel est à l'IANA
- Textes à lire : les RFC 6838 et 4289.
- Pour un type, il faut écrire un RFC et qui ait le statut de norme (voir RFC 9694).
- Exemple font/ (RFC 8081 en 2017)

Mais qui enregistre les types et les sous-types ?

- L'enregistrement officiel est à l'IANA
- Textes à lire : les RFC 6838 et 4289.
- Pour un type, il faut écrire un RFC et qui ait le statut de norme (voir RFC 9694).
- Exemple font/
- Il faut bien décrire les critères d'acceptation des sous-types et en avoir des réels.

Mais qui enregistre les types et les sous-types ?

- L'enregistrement officiel est à l'IANA
- Textes à lire : les RFC 6838 et 4289.
- Pour un type, il faut écrire un RFC et qui ait le statut de norme (voir RFC 9694).
- Exemple font/
- Il faut bien décrire les critères d'acceptation des sous-types et en avoir des réels.
- Enregistrer les sous-types est bien plus facile.
<https://www.iana.org/form/media-types>

Et haptics/ ?

Key features



Adjustable stroking so
you control the pace



Sync with partners, cam
performers & 2D/VR
content



Best suited for average
or smaller penis sizes



2 hours of power to keep
coming back for more

L'exemple haptics/

L'exemple haptics/

- Pour les ressources décrivant des actions physiques,

L'exemple haptics/

- Pour les ressources décrivant des actions physiques,
- RFC 9695 "*The 'haptics' Top-level Media Type*" en mars 2025.

L'exemple haptics/

- Pour les ressources décrivant des actions physiques,
- RFC 9695 “*The 'haptics' Top-level Media Type*” en mars 2025.
- `haptics/ivs`, `haptics/hjif`, `haptics/hmpg`.

Facettes des sous-types

Facettes des sous-types

- Préfixe vnd pour *vendor*;

`application/vnd.adobe.flash.movie`

`application/vnd.openxmlformats-officedocument.wordprocessingml.document`

Facettes des sous-types

- Préfixe `vnd` pour *vendor*;
`application/vnd.adobe.flash.movie`
`application/vnd.openxmlformats-officedocument.wordprocessingml.document`
- Convention `+formatdebase` pour des types dérivés
(`application/activity+json`, `application/atom+xml` ou `application/dots+cbor`).

On n'utilise plus x-

On n'utilise plus x-

- Autrefois, il était courant d'utiliser des types expérimentaux préfixés par x-.

On n'utilise plus x-

- Autrefois, il était courant d'utiliser des types expérimentaux préfixés par x-.
- Auto-attribués sans formalités.

On n'utilise plus x-

- Autrefois, il était courant d'utiliser des types expérimentaux préfixés par x-.
- Auto-attribués sans formalités.
- C'est (théoriquement) fini depuis le RFC 6648 en 2012.

On n'utilise plus x-

- Autrefois, il était courant d'utiliser des types expérimentaux préfixés par x-.
- Auto-attribués sans formalités.
- C'est (théoriquement) fini depuis le RFC 6648 en 2012.
- C'était prévu pour les expérimentations mais ça a été abusé.

On n'utilise plus x-

- Autrefois, il était courant d'utiliser des types expérimentaux préfixés par x-.
- Auto-attribués sans formalités.
- C'est (théoriquement) fini depuis le RFC 6648 en 2012.
- C'était prévu pour les expérimentations mais ça a été abusé.
- Et puis les expérimentatipons temporaires durent...

On ne fait pas de classification

On ne fait pas de classification

- Les types de média sont conçus pour une utilisation pratique : permettre au client Internet de savoir quel logiciel lancer.

On ne fait pas de classification

- Les types de média sont conçus pour une utilisation pratique : permettre au client Internet de savoir quel logiciel lancer.
- Ce n'est pas une vraie classification, au sens des bibliothécaires ou archivistes.

On ne fait pas de classification

- Les types de média sont conçus pour une utilisation pratique : permettre au client Internet de savoir quel logiciel lancer.
- Ce n'est pas une vraie classification, au sens des bibliothécaires ou archivistes.
- Par exemple, il n'y a pas de types de média pour les langages de programmation car on n'imagine pas le logiciel client lancer le compilateur : `text/plain` suffit.

On ne fait pas de classification

- Les types de média sont conçus pour une utilisation pratique : permettre au client Internet de savoir quel logiciel lancer.
- Ce n'est pas une vraie classification, au sens des bibliothécaires ou archivistes.
- Par exemple, il n'y a pas de types de média pour les langages de programmation car on n'imagine pas le logiciel client lancer le compilateur : `text/plain` suffit.
- Et si on faisait des types de média pour les langages de programmation (`text/c`, `text/zig`, `text/rust`, etc) se poserait la question des versions, pour les langages dont la norme évolue, comme C.