

# RFC 7763 : The text/markdown Media Type

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 23 mars 2016

Date de publication du RFC : Mars 2016

<https://www.bortzmeyer.org/7763.html>

---

Le format Markdown est un excellent format léger de marquage de textes, permettant d'enrichir un texte, sans pour autant avoir à apprendre un gros langage de marquage comme LaTeX. Markdown est très bien adapté aux courts rapports, aux articles simples, aux fichiers README, à la génération de pages Web... Markdown a de nombreuses variantes et le but de ce nouveau RFC n'est pas de normaliser Markdown (une tâche probablement impossible) mais juste d'enregistrer un type MIME pour ce format, `text/markdown`. Donc, on décrit, on ne normalise pas.

Markdown est un langage de marquage du texte, comme ReST (utilisé dans le livre CNP3 <<https://www.bortzmeyer.org/cnp3.html>>) ou comme les langages des Wiki. On peut donc éditer du Markdown avec n'importe quel éditeur (mes lecteurs ne seront pas surpris d'apprendre que j'utilise Emacs). Le source Markdown est donc du texte lisible tel quel, sans logiciel spécifique, mais un logiciel peut le traiter pour produire une forme plus agréable et plus efficace, par exemple du HTML pour publication sur le Web.

Voici un exemple de document Markdown n'utilisant, sauf erreur, que les constructions communes à toutes les variantes : (en ligne sur <https://www.bortzmeyer.org/files/test1.md>). On peut le traiter, par exemple avec `discount` <<http://www.pell.portland.or.us/~orc/Code/discount/>>, une mise en œuvre simple de Markdown, en C, qui ne peut produire que du HTML :

```
% markdown test1.md
<h1>Début</h1>

<p>Un test très simple
en Markdown.</p>
...
<h2>Suite</h2>
...
```

Mais on peut aussi utiliser le bien plus riche Pandoc, qui a de nombreuses extensions à Markdown, et peut produire beaucoup de formats de sortie, ici du PDF, via LaTeX :

```
% pandoc --to latex --latex-engine=xelatex -o test1.pdf test1.md
% evince test1.pdf
...
```

Un point important de Markdown est qu'il n'y a **jamais d'erreur de syntaxe** (section 1 du RFC) : tout fichier texte est un fichier Markdown légal (même s'il ne donne pas toujours le résultat attendu par son auteur). C'est donc un monde très différent de LaTeX ou de XML (au passage, ce blog est écrit en XML <<https://www.bortzmeyer.org/blog-implementation.html>>). Markdown se veut un langage léger, imposant peu de contraintes et peu d'apprentissage. Avec Markdown, on a toujours un résultat, même suboptimal. Ce n'est pas considéré comme un problème. Si on veut du « sérieux », il faut utiliser LaTeX ou DocBook.

La « spécification » de Markdown est très grossière, et laisse le doute sur beaucoup de points mais le choix de l'IETF a été de ne pas essayer d'en écrire une correcte : Markdown est fait ainsi.

Un autre point important est le grand nombre de variantes, souvent incompatibles. L'auteur original, John Gruber, a toujours refusé de permettre des évolutions du langage et chacun a donc ajouté les siennes. Il y a eu plusieurs tentatives de normalisation (la principale étant CommonMark <<http://commonmark.org/>>), mais sans résultat clair. Au début du processus long et compliqué qui a abouti à ce RFC, il était question de faire une nouvelle tentative, sous la houlette de l'IETF cette fois. Mais le projet RFC s'est assez vite rabattu sur une ambition plus modeste : enregistrer le type MIME et documenter (partiellement) l'état de Markdown. Un autre RFC, le RFC 7764<sup>1</sup>, décrit les usages de Markdown. La discussion a été d'autant plus chaude que la « communauté » Markdown est largement extérieure à l'IETF.

On pourrait dire que Markdown n'est pas un langage, mais une famille de langages, ayant des éléments communs. Un bon exemple de la variété de Markdown est, par exemple, le cas des métadonnées. Pandoc permet d'écrire en début de fichier :

```
% Document title
% Document author
% Document date
```

Mais d'autres variantes de Markdown ne vont pas comprendre ces métadonnées, les laissant telles quelles. Autre exemple de variété, les commentaires, pour lesquels il n'existe pas de solution standard (il y a un bon article sur StackOverflow <<http://stackoverflow.com/questions/4823468/comments-in-markdown>> 2088598 qui discute les différentes possibilités.)

Cet enregistrement du type MIME de Markdown s'inspire de celui de `text/troff`, un type pour un autre langage de marquage (RFC 4263). Après tout, beaucoup de gens considèrent que Markdown est « le troff d'aujourd'hui ».

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc7764.txt>

---

La section 2 de notre RFC décrit formellement l'enregistrement du type MIME `text/markdown` et de l'extension associée, `.md` (`.markdown` est également accepté). Ce type a plusieurs paramètres possibles (les paramètres sont mis après un point-virgule, par exemple `text/markdown; charset=UTF-8; variant=Original` va désigner du Markdown à la syntaxe originale de Gruber, encodé en UTF-8).

Premier paramètre, et obligatoire (RFC 6838, section 4.2.1), `charset`, qui désigne l'encodage du texte. Les autres paramètres sont facultatifs.

Le plus important est sans doute `variant` qui désigne le dialecte Markdown particulier utilisé. Ce fut l'une des plus chaudes discussions à l'IETF avant la sortie de ce RFC. Notamment, fallait-il un registre IANA des variantes (une sorte de catalogue des dialectes) ou bien laisser les auteurs mettre ce qu'ils veulent, ce qui correspondait mieux au côté très peu organisé de la « communauté » Markdown, très informelle? (Différentes versions du "draft" qui a mené à ce RFC faisaient des choix différents.) Le paramètre `variant` a finalement un registre `<https://www.iana.org/assignments/markdown-variants/markdown-variants.xml>` (section 6.1) mais l'émetteur n'est pas obligé de se limiter aux valeurs de ce registre. Il faut considérer la valeur de ce paramètre comme une simple indication. Parmi les valeurs actuellement enregistrées : `Original` (le Markdown des débuts, celui de Gruber), `GFM` (celui de GitHub), `pandoc` (celui de ce logiciel), `CommonMark` (le candidat à la normalisation `<http://www.commonmark.org>`) etc. Notez que ce paramètre `variant`, au gré des évolutions du "draft" à l'IETF, s'est nommé `syntax`, `flavor`, `processor`...

Des valeurs sont interdites dans ce registre : `Standard`, `Common` et `Markdown` ne peuvent être enregistrés comme variantes. C'est parce que « une partie de la communauté » (en réalité Gruber seul) a protesté contre la volonté de déclarer une variante particulière comme étant « standard » ou « officielle ».

Et si on veut ajouter une entrée à ce registre des variantes de Markdown? La politique d'enregistrement est le « Premier Arrivé, Premier Servi » du RFC 5226, donc très légère. Il suffit d'indiquer deux-trois trucs sur la variante et roulez, jeunesse. Dans des versions initiales de ce RFC, le gabarit d'enregistrement d'une variante était bien plus complexe, bien trop détaillé, avec des infos très volatiles comme "*currently maintained*" ou "*anticipated output types*" mais il est maintenant réduit à seulement cinq champs obligatoires.

Un des projets qui avaient été lancés était celui d'enregistrer non pas les noms des variantes mais leurs caractéristiques (« permet les notes de base de page », « permet la création automatique de liens hypertexte »). À la réunion IETF 90 `<http://www.ietf.org/meeting/90/index.html>`, la proposition était de pouvoir écrire des choses comme `text/markdown; variations=footnotes, line_blocks` pour désigner du texte Markdown utilisant les notes de bas de page et les blocs de lignes considérés comme un paragraphe. Une telle approche aurait nécessité la constitution d'un catalogue compliqué, et aurait imposé aux auteurs de garder trace de quelles extensions ils utilisent. Elle n'a finalement pas été retenue.

L'enregistrement d'un type MIME nécessite une analyse de ses risques de sécurité. Rien d'extraordinaire ici, le RFC note juste que du texte avec du marquage ne présente guère de risque (contrairement à un format comme TrueType qui inclut un langage de Turing). Attention toutefois : ce que le processeur Markdown ne comprend pas, il l'envoie verbatim dans le format de sortie (ce truc est souvent utilisé pour mettre du HTML spécifique, ou du JavaScript, lorsqu'on utilise Markdown pour produire des pages Web). Du logiciel malveillant peut donc être copié ainsi.

Et la section sur l'interopérabilité rappelle ce qui a été dit plus haut : les différents Markdown vont donner des résultats différents, d'autant plus différents qu'ils utilisent des techniques spécifiques d'une variante.

Et si on veut indiquer un point précis d'un document Markdown? Contrairement à HTML, Markdown n'a pas d'identificateurs de fragments d'un texte (section 3 du RFC). Mais on peut réutiliser une partie de la syntaxe du RFC 5147 : en terminant un URI par `#lines=N`, on accède (si le logiciel client connaît le RFC 5147) à la Nième ligne du document Markdown.

À noter qu'on peut écrire des RFC en Markdown : c'est expliqué dans le RFC 7328 et c'est utilisé, par exemple, pour ceux du groupe "*Human Rights Protocol Considerations*" <<https://github.com/n11z/IRTF-HRPC>> de l'IRTF (ce groupe est typiquement moins geek que la moyenne de l'IETF/IRTF).

Un exemple plus important que l'exemple un peu artificiel (en ligne sur <https://www.bortzmeyer.org/files/test1.md>) que je donnais au début est mon rapport sur la panne DNS d'Oleane (en ligne sur <https://www.bortzmeyer.org/files/panne-dns-oleane-2016.md>). On peut le traiter avec Pandoc et un Makefile comme :

```
TARGETS=panne-dns-oleane-2016.pdf panne-dns-oleane-2016.html

all: ${TARGETS}

%.pdf: %.md
pandoc --latex-engine=xelatex -o $@ $^

%.html: %.md
pandoc -o $@ $^

clean:
rm -f ${TARGETS}
```

Markdown est souvent présent dans les « micro-éditeurs » qui permettent de remplir des documents via le Web. C'est le cas à GitHub où les "*gists*" dont le nom se termine par `.md` sont automatiquement convertis. C'est ainsi que j'ai fait un gist de ce rapport <<https://gist.github.com/bortzmeyer/616541997376c1380de6>> (cliquez sur le bouton "*Raw*" pour avoir le source).

Quelques lectures sur Markdown :

- La syntaxe originale <<http://daringfireball.net/projects/markdown/>> ,
- Le fameux article de Jeff Atwood <<http://blog.codinghorror.com/is-html-a-humane-markup-language/>> sur les langages de marquage et leur « humanité »,
- Un très bon article sur la normalisation de Markdown <<http://lwn.net/Articles/610975/>> et sur les exigences de Gruber.
- Un bon article <<http://arstechnica.com/information-technology/2014/10/markdown-throwdown/>> sur l'affaire Standard Markdown,
- Une comparaison <<https://css-tricks.com/choosing-right-markdown-parser/>> des différentes variantes de Markdown,
- Une autre façon d'étudier les différentes variantes est de les tester en ligne <[http://johnmacfarlane.net/babelmark2/?text=Here%27s+a+list+of+stuff%3A%0A\\*+item+one%0A\\*+item+two%0A\\*+item+three%0A](http://johnmacfarlane.net/babelmark2/?text=Here%27s+a+list+of+stuff%3A%0A*+item+one%0A*+item+two%0A*+item+three%0A)> ,
- Un bon article <<http://programminghistorian.org/lessons/sustainable-authorship-in-pla>> sur l'utilisation de Pandoc pour des documents compliqués, par exemple des articles scientifiques dans les sciences humaines et sociales.