RFC 5564 : Linguistic Guidelines for the Use of Arabic Characters in Internet Domains

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 13 février 2010. Dernière mise à jour le 18 juin 2010

Date de publication du RFC : Février 2010

https://www.bortzmeyer.org/5564.html

La norme IDN, décrite dans les RFC 5890 [1] et suivants <https://www.bortzmeyer.org/idnabis.html>, permet d'écrire des noms de domaine en Unicode, c'est-à-dire avec toutes les écritures du monde. Unicode est très riche et comprend une variété de caractères qui peut être jugée excessive dans certains cas d'utilisation. Voilà pourquoi, dans le cas d'IDN, il est parfois préférable de réduire le jeu de caractères utilisable, de le limiter à certains caractères. C'est ce que ce RFC fait pour l'écriture arabe.

Cette écriture a la particularité de s'écrire de droite à gauche. Cela signifie qu'un nom de domaine partiellement en Unicode (avec, par exemple, le TLD .com à droite) n'a pas grand sens et cela explique en partie le peu de déploiement des IDN en écriture arabe jusqu'à présent, puisque l'ICANN bloque toujours leur déploiement dans la racine du DNS. Mais il y a aussi d'autres raisons, comme un certain conservatisme dans les registres nationaux des pays arabes.

L'écriture arabe est utilisée par bien d'autres langues que l'arabe, par exemple en Iran pour le persan et au Pakistan pour l'ourdou. Les registres non-arabophones sont d'ailleurs parfois les plus dynamiques comme l'a montré celui du .ir, premier à enregistrer des noms de domaines en écriture arabe. Malheureusement, le RFC se focalise exclusivement sur la langue arabe.

Maintenant, pour un registre qui veut permettre cet enregistrement, faut-il autoriser tous les caractères « arabes » d'Unicode ? Il y en a beaucoup et certains peuvent ne pas correspondre à un réel usage. Leur présence dans la liste pourrait donc dérouter. Il est sans doute souhaitable que les registres des pays utilisant l'écriture arabe se mettent d'accord sur une liste de caractères utiles, et c'est le but de ce RFC.

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc5890.txt

La section 1 décrit le cadre général de ce document. Il a été écrit par l'AWG-ADN ("*Arabic Working Group on Arabic Domain Names*"), une émanation de la Ligue arabe. C'est dire si son origine est très « gouvernementale » et reflète souvent les points de vue de gouvernements autoritaires, voire dictatoriaux, facilement effrayés par la nouveauté. Ainsi, l'idée de « bannir » des caractères « non officiels » de la liste correspond bien à la mentalité de contrôle et de restriction de beaucoup de gouvernements arabes.

Les sections 2.1 et 2.3 du RFC listent les particularités de l'arabe en ce qui concerne les IDN et les conséquences à en tirer. Par exemple (section 2.1.1), les « accents » comme [Caractère Unicode non montré [2]] (`U+0651`, le Chadda) sont rejetés pour les noms de domaine en écriture arabe. Autre cas (section 2.1.3), des caractères différentes peuvent parfois être confondus par certains programmes. C'est le cas, par exemple le [Caractère Unicode non montré] (`U+0629`, le Té' Marbouta), souvent remplacé par le [Caractère Unicode non montré] (`U+0647`, le H[Caractère Unicode non montré]). Une telle confusion ne respecte pas les règles de la langue (le RFC parle même, avec emphase, de violation de l'éthique) et doit donc être évitée.

La section 2.3.1 couvre le difficile problème des chiffres. Il existe deux séries de chiffres en arabe, les chiffres que les francophones appellent « arabes », bien qu'ils soient d'origine indienne (0, 1, 2, 3, ..., `U+0030` à `U+0039`) et les chiffres indiens d'origine ([Caractère Unicode non montré], [Caractère Unicode non montré], [Caractère Unicode non montré] ..., `U+0660` à `U+0669`). Les premiers sont plutôt utilisés au Maghreb, les seconds au Moyen-Orient. Si les premières versions de ce document n'autorisaient que le premier jeu, celui compatible avec ASCII, le RFC accepte finalement les deux séries, prohibant uniquement leur mélange dans un même nom de domaine.

Compte-tenu de ces points, une liste de caractères autorisés est définie, en section 2.2. Elle comprend 60 caractères en tout. Voici un exemple de code Python pour tester (les chiffres indiens d'origine ont été exclus). Pour tester un composant d'un nom de domaine, il faut appeler `check(lenom)` :

```
def and_couple(l, r):
    return l and r

def and_tuple(t):
    return reduce(and_couple, t, True)

def check_ch(c):
    # Les chiffres indo-arabes, de U+0660 à U+0669, sont exclus
    return c in [unichr(0x0621), unichr(0x0622),unichr(0x0623),unichr(0x0624),unichr(0x0625),
                 unichr(0x0626),unichr(0x0627),unichr(0x0628),unichr(0x0629),unichr(0x062A),
                 unichr(0x062B),unichr(0x062C),unichr(0x062D),unichr(0x062E),unichr(0x062F),
                 unichr(0x0630),unichr(0x0631),unichr(0x0632),unichr(0x0633),unichr(0x0634),
                 unichr(0x0635),unichr(0x0636),unichr(0x0637),unichr(0x0638),unichr(0x0639),
                 unichr(0x063A),unichr(0x0641),unichr(0x0642),unichr(0x0643),unichr(0x0644),
                 unichr(0x0645),unichr(0x0646),unichr(0x0647),unichr(0x0648),unichr(0x0649),
                 unichr(0x064A),
                 '0', '1', '2', '3', '4', '5', '6', '7', '8', '9']

def check(name):
    return and_tuple(map(check_ch, name))
```

2. Car trop difficile à faire afficher par LaTeX