Passerelle - Bug #8727

csv: bogue quand le fichier commence par un BOM

21 octobre 2015 10:31 - Thomas Noël

Statut:	Fermé	**Début:**	21 octobre 2015
Priorité:	Normal	**Echéance:**	18 décembre 2015
Assigné à:	Thomas Noël	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	

Description

Quand le fichier CSV commence par un BOM (U+FEFF) celui ci est considéré comme appartenant au fichier.

Résultat, si la première ligne est 0,choix1, le 0 est rendu en JSON comme "id": "<U+FEFF>0".

Révisions associées

Révision b4454520 - 18 décembre 2015 15:33 - Thomas Noël

csv: remove BOM if exists (#8727)

Historique

#1 - 11 décembre 2015 11:16 - Thomas Noël

- Echéance mis à 18 décembre 2015

- Assigné à mis à Thomas Noël

- Priorité changé de Normal à Haut

Il faut donc lors de la lecture du CSV retirer le BOM s'il existe sur la première ligne.

(Il y a une recherche préalable à faire sur les possibilités du module python csv)

#3 - 18 décembre 2015 12:38 - Thomas Noël

- Fichier 0001-csv-remove-BOM-if-exists-8727.patch ajouté

- Statut changé de Nouveau à En cours

- Patch proposed changé de Non à Oui

Je veux bien l'avis de quelqu'un qui maitrise un peu mieux le sujet... Mais bon, dans l'optique de fichiers CSV en utf-8, ça semble ok.

#4 - 18 décembre 2015 13:21 - Benjamin Dauvergne

D'après ma lecture de https://docs.python.org/2/library/codecs.html:

> Without external information it's impossible to reliably determine which encoding was used for encoding a Unicode string. Each charmap encoding can decode any random byte sequence. However that's not possible with UTF-8, as UTF-8 byte sequences have a structure that doesn't allow arbitrary byte sequences. To increase the reliability with which a UTF-8 encoding can be detected, Microsoft invented a variant of UTF-8 (that Python 2.5 calls "utf-8-sig") for its Notepad program: Before any of the Unicode characters is written to the file, a UTF-8 encoded BOM (which looks like this as a byte sequence: 0xef, 0xbb, 0xbf) is written. As it's rather improbable that any charmap encoded file starts with these byte values (which would e.g. map to
>
> LATIN SMALL LETTER I WITH DIAERESIS
> RIGHT-POINTING DOUBLE ANGLE QUOTATION MARK
> INVERTED QUESTION MARK
>
> in iso-8859-1), this increases the probability that a utf-8-sig encoding can be correctly guessed from the byte sequence. So here the BOM is not used to be able to determine the byte order used for generating the byte sequence, but as a signature that helps in guessing the encoding. On encoding the utf-8-sig codec will write 0xef, 0xbb, 0xbf as the first three bytes to the file. On decoding utf-8-sig will skip those three bytes if they appear as the first three bytes in the file. In UTF-8, the use of the BOM is discouraged and should generally be avoided.

Ack.

#5 - 18 décembre 2015 13:22 - Benjamin Dauvergne

Mais bon on nous demandera forcément de gérer du latin1 ou un codepage à la con un jour.

#6 - 18 décembre 2015 15:34 - Thomas Noël

Ok... Allez, on va rester sur "CSV en utf-8 obligatoire". C'est un truc qui est géré même par windows (alors que pour le BOM, on peut pas décider).

```
commit b44545207c52a5eb1774057099cc9b883c9e0aca
Author: Thomas NOEL <tnoel@entrouvert.com>
Date:   Fri Dec 18 12:35:39 2015 +0100

    csv: remove BOM if exists (#8727)
```

#7 - 18 décembre 2015 15:34 - Thomas Noël

- Statut changé de En cours à Résolu (à déployer)

- Priorité changé de Haut à Normal

#8 - 04 août 2018 12:32 - Benjamin Dauvergne

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-csv-remove-BOM-if-exists-8727.patch	1,77 ko 18 décembre 2015	Thomas Noël