

Distribution EOLE - Tâche #34515

Scénario # 34440 (Terminé (Sprint)): Sur Zéphir on a systématiquement "détection de 1 fichiers modifiés" si on ajouter un dépôt additionnelle en mode clé

Étude

22/08/2022 12:29 - Benjamin Bohard

Statut:	Fermé	Début:	25/07/2022
Priorité:	Normal	Echéance:	
Assigné à:	Benjamin Bohard	% réalisé:	100%
Version cible:	Prestation Cadoles MEN 2022 30-35 (été)	Temps estimé:	0.00 heure
		Temps passé:	0.00 heure
Description			

Historique

#1 - 22/08/2022 12:29 - Benjamin Bohard

- Statut changé de Nouveau à En cours

#2 - 22/08/2022 12:30 - Benjamin Bohard

Il semble que le problème réside plutôt dans la conversion des objets complexes (listes, dictionnaires) en chaînes de caractères.

Test avec une fonction qui convertit tous les objets complexes en chaînes de caractères en partant des éléments les plus petits, sur le modèle de la fonction dans zéphir backend utils.convert.to_bytes

```
##*- coding: utf-8 -*-
import cjson
import json
import codecs

def to_bytes(objet):
    """Transforme les objets str ou Binary contenus dans un objet en bytes
    """
    if isinstance(objet, tuple):
        l = []
        for item in objet:
            l.append(to_bytes(item))
        return '({})'.format(', '.join(l))
    if isinstance(objet, list):
        l = []
        for item in objet:
            l.append(to_bytes(item))
        return '[{}]'.format(', '.join(l))
    if isinstance(objet, dict):
        dico={}
        for cle in objet:
            dico[to_bytes(cle)] = to_bytes(objet[cle])
        return '{{{}}}'.format(', '.join(['{}: {}'.format(e1[0], e1[1]) for e1 in dico.items()]))
    if isinstance(objet, unicode):
        print(type(objet), objet)
        string = objet.encode('utf-8')
        return string
    if isinstance(objet, int):
        return str(int)
    if objet == None:
        return ''
    return objet

with open('/etc/eole/config.eol', 'r') as eol:
#with codecs.open('/etc/eole/config.eol', 'r', 'utf-8') as eol:
    cu = cjson.decode(eol.read(), all_unicode=True)
    eol.seek(0)
    cs = cjson.decode(eol.read())
    eol.seek(0)
    cj = json.loads(eol.read())
```

```

with open('/tmp/variables.eol', 'w') as v:
    v.write('\n'.join(['{}:{}'.format(el[0], to_bytes(el[1].get('val')) for el in sorted(cs.items()) if isinstance(el[1], dict))
    #v.write(u'\n'.join([u'{}:{}'.format(el[0].decode('utf-8'), to_bytes(el[1].get('val')).decode('utf-8')) for el in sorted(cu.items()) if isinstance(el[1], dict))
    #v.write(u'\n'.join([u'{}:{}'.format(el[0].decode('utf-8'), to_bytes(el[1].get('val')).decode('utf-8')) for el in sorted(cj.items()) if isinstance(el[1], dict))
    #v.write('\n'.join(['{}:{}'.format(el[0], el[1].get('val')) for el in sorted(cs.items()) if isinstance(el[1], dict))

#with codecs.open('/tmp/variables.eol', 'w', 'utf-8') as v:
    #v.write(u'\n'.join([u'{}:{}'.format(el[0].decode('utf-8'), to_bytes(el[1].get('val')).decode('utf-8')) for el in sorted(cs.items()) if isinstance(el[1], dict))
    #v.write(u'\n'.join([u'{}:{}'.format(el[0].decode('utf-8'), to_bytes(el[1].get('val')).decode('utf-8')) for el in sorted(cu.items()) if isinstance(el[1], dict))
    #v.write(u'\n'.join([u'{}:{}'.format(el[0].decode('utf-8'), to_bytes(el[1].get('val')).decode('utf-8')) for el in sorted(cj.items()) if isinstance(el[1], dict))

```

#3 - 22/08/2022 12:36 - Benjamin Bohard

Le problème suivant qui se pose est le traitement différencié, sur Zéphir, des objets complexes et des objets simples.

Exemple sur zéphir :

```

additional_repository_key_type:{'0': 'URL de la clé'}
additional_repository_key_url:{'0': 'https://vulcain.cadoles.com/cadoles.gpg'}
additional_repository_name:['cadoles']
additional_repository_source:{'0': 'deb https://vulcain.cadoles.com 2.7.0-unstable main'}
adresse_ip_dns:['192.168.0.1']
adresse_ip_eth0:192.168.0.5

```

Et avec la conversion en profondeur et l'utilisation du paramètre `all_unicode=True`, sur le seth :

```

additional_repository_key_type:{'0': 'URL de la clé'}
additional_repository_key_url:{'0': 'https://vulcain.cadoles.com/cadoles.gpg'}
additional_repository_name:['cadoles']
additional_repository_source:{'0': 'deb https://vulcain.cadoles.com 2.7.0-unstable main'}
adresse_ip_dns:['192.168.0.1']
adresse_ip_eth0:'192.168.0.5'

```

Enfin, avec la conversion en profondeur mais avec le paramètre `all_unicode=False`, sur le seth:

```

additional_repository_key_type:{0: 'URL de la clé'}
additional_repository_key_url:{0: https://vulcain.cadoles.com/cadoles.gpg}
additional_repository_name:['cadoles']
additional_repository_source:{0: deb https://vulcain.cadoles.com 2.7.0-unstable main}

```

adresse_ip_dns: [192.168.0.1]
adresse_ip_eth0: 192.168.0.5

Dans ce dernier cas, seule la variable `additional_repository_key_type` est convertie en unicode (c'est la seule qui le nécessite)

#4 - 22/08/2022 14:07 - Benjamin Bohard

- Statut changé de *En cours* à *À valider*

#5 - 22/08/2022 14:07 - Benjamin Bohard

- % réalisé changé de 0 à 100

#6 - 22/08/2022 14:24 - Benjamin Bohard

- Assigné à mis à Benjamin Bohard

#7 - 05/09/2022 13:40 - Ludwig Seys

- Statut changé de *À valider* à *Résolu*

#8 - 06/09/2022 16:35 - Joël Cuissinat

- Statut changé de *Résolu* à *Fermé*

- Restant à faire (heures) mis à 0.0