

Hobo - Development #56991

UnpicklingError: pickle data was truncated

16 septembre 2021 11:38 - Sentry lo

Statut:	Fermé	Début:	16 septembre 2021
Priorité:	Normal	Echéance:	
Assigné à:	Emmanuel Cazenave	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposé:	Oui		
Description			
https://sentry.entrouvert.org/entrouvert/publik/issues/52156/			
UnpicklingError: pickle data was truncated File "uwsgidecorators.py", line 58, in manage_spool_request vars[k] = pickle.loads(vars.pop(k))			
Demandes liées:			
Lié à Hobo - Development #57019: implémenter notre propre uwsgidecorators		Fermé	16 septembre 2021

Révisions associées

Révision d84fd70d - 16 septembre 2021 18:39 - Emmanuel Cazenave

provisionning: pass data to spooler function in body parameter (#56991)

The `data` can be very long, by spooler files cannot be longer than 64k except for the body bytes parameter.

See <https://uwsgi-docs.readthedocs.io/en/latest/Spooler.html#spool-files>

Using the body parameters is only possible with pass_arguments=False (not documented).

Historique

#1 - 16 septembre 2021 11:39 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

#2 - 16 septembre 2021 11:41 - Benjamin Dauvergne

- Fichier 0001-provisionning-pass-data-to-spooler-function-in-body-.patch ajouté

- Tracker changé de Bug à Development

- Statut changé de Nouveau à Solution proposée

- Patch proposé changé de Non à Oui

#3 - 16 septembre 2021 11:43 - Benjamin Dauvergne

- Projet changé de Suivi des traces à Hobo

#4 - 16 septembre 2021 11:44 - Benjamin Dauvergne

Donc le souci à la base c'est la sérialisation des arguments lorsqu'on provisionne les messages HTTP depuis le spooler, ne peut pas dépasser 64k¹, et sur un provisionning full des rôles ou des usagers, ça dépassera souvent.

¹<https://uwsgi-docs.readthedocs.io/en/latest/Spooler.html#spool-files>

#5 - 16 septembre 2021 12:17 - Emmanuel Cazenave

Bien vu.

Ce serait plutôt provision_spooler.spool(**kwargs) si on veut rester sur ton idée.

Mais je trouve que ça complique la lecture d'introduire une nouvelle méthode provision_spooler, pourquoi ne pas faire body = pickle.dumps(data) dans le middleware, passer ça à spooler.py::provision où tu peux faire un data = pickle.loads(kwargs['body']) et voilà ?

#6 - 16 septembre 2021 12:26 - Benjamin Dauvergne

Emmanuel Cazenave a écrit :

Ce serait plutôt `provision_spooler.spool(**kwargs)` si on veut rester sur ton idée.

Non, ce n'est plus la bonne API depuis des années¹.

Mais je trouve que ça complique la lecture d'introduire une nouvelle méthode `provision_spooler`, pourquoi ne pas faire `body = pickle.dumps(data)` dans le middleware, passer ça à `spooler.py::provision` où tu peux faire un `data = pickle.loads(kwargs['body'])` et voilà ?

Je trouve que ça complique le middleware d'y voir des détails d'implémentation de uwsgi.

¹<https://github.com/unbit/uwsgi/commit/164069e1948c4d30552606dd572c21b90d61a65f>

#7 - 16 septembre 2021 13:51 - Emmanuel Cazenave

Benjamin Dauvergne a écrit :

Je trouve que ça complique le middleware d'y voir des détails d'implémentation de uwsgi.

On est déjà dans un `if 'uwsgi' in sys.modules ...`

A part ça ton patch ne marche pas :

```
[spooler /var/lib/passerelle/spooler pid: 33441] managing request uwsgi_spoolfile_on_cazino-laptop_33466_1_686
55914_1631792990_225889 ...
Traceback (most recent call last):
  File "/home/cazino/envs/publik-env-py3/lib/python3.9/site-packages/uwsgidecorators.py", line 60, in manage_s
pool_request
    vars = _decode_from_spooler(vars)
  File "/home/cazino/envs/publik-env-py3/lib/python3.9/site-packages/uwsgidecorators.py", line 41, in _decode_
from_spooler
    return dict((_decode1(K), _decode1(V)) for (K, V) in list(vars.items()))
  File "/home/cazino/envs/publik-env-py3/lib/python3.9/site-packages/uwsgidecorators.py", line 41, in <genexpr
>
    return dict((_decode1(K), _decode1(V)) for (K, V) in list(vars.items()))
  File "/home/cazino/envs/publik-env-py3/lib/python3.9/site-packages/uwsgidecorators.py", line 31, in _decode1
    return val.decode('utf-8')
UnicodeDecodeError: 'utf-8' codec can't decode byte 0x80 in position 0: invalid start byte
```

#8 - 16 septembre 2021 14:25 - Benjamin Dauvergne

- Assigné à Benjamin Dauvergne supprimé

Essayez avec json plutôt que pickle.

#9 - 16 septembre 2021 14:31 - Emmanuel Cazenave

- Statut changé de Solution proposée à En cours

- Assigné à mis à Emmanuel Cazenave

#10 - 16 septembre 2021 14:38 - Valentin Deniaud

Emmanuel Cazenave a écrit :

A part ça ton patch ne marche pas :
UnicodeDecodeError: 'utf-8' codec can't decode byte 0x80 in position 0: invalid start byte

(pour info ça correspond à un ticket catégorie bug côté uwsgi <https://github.com/unbit/django-uwsgi/issues/10>, j'étais déjà tombé dessus)

#11 - 16 septembre 2021 14:51 - Benjamin Dauvergne

Valentin Deniaud a écrit :

Emmanuel Cazenave a écrit :

A part ça ton patch ne marche pas :
UnicodeDecodeError: 'utf-8' codec can't decode byte 0x80 in position 0: invalid start byte

(pour info ça correspond à un ticket catégorie bug côté uwsgi <https://github.com/unbit/django-uwsgi/issues/10>, j'étais déjà tombé dessus)

C'est fou ça, en gros on a déjà de la chance que ça fonctionne dans le cas général, parce que kwarg c'est bien un pickle aussi.

#12 - 16 septembre 2021 16:25 - Emmanuel Cazenave

- Fichier 0001-provisionnement-pass-data-to-spooler-function-in-body-.patch ajouté

- Statut changé de En cours à Solution proposée

uwsgidecorators c'est vraiment limpide.

Déjà pour pouvoir utiliser 'body' il faut passer en mode pass_arguments=False, sinon pas d'accès à cette donnée, uwsgidecorators l'accepte en entrée mais ne la passe pas à la fonction ...

En mode pass_arguments=False, il faut passer tous les arguments en bytes mais uwsgidecorators s'amuse à tout décoder en utf-8, d'où effectivement un json.dumps plutôt que pickle. Voilà voilà.

#13 - 16 septembre 2021 16:41 - Thomas Noël

Pour la lisibilité, plutôt utiliser force_bytes que les .encode('utf-8') ? (si tant est que ça fait bien la même chose, moi et l'encodage sommes fâchés)

#14 - 16 septembre 2021 17:31 - Emmanuel Cazenave

- Fichier 0001-provisionnement-pass-data-to-spooler-function-in-body-.patch ajouté

Voilà.

#15 - 16 septembre 2021 17:32 - Benjamin Dauvergne

Emmanuel Cazenave a écrit :

uwsgidecorators c'est vraiment limpide.

Je vais proposer qu'on s'en passe carrément parce que c'est du caca.

#16 - 16 septembre 2021 17:46 - Thomas Noël

- Statut changé de Solution proposée à Solution validée

Benjamin Dauvergne a écrit :

Emmanuel Cazenave a écrit :

uwsgidecorators c'est vraiment limpide.

Je vais proposer qu'on s'en passe carrément parce que c'est du caca.

Le patch d'Emmanuel me va quand même pas trop mal, même si oui, c'est assez lourdingue au final. Ceci étant, on "suit" ainsi les recommandations de la doc uwsgi qui dit bien de faire attention aux bytes, au 64k, etc.

Bref, comme je sais que Manu a testé en vrai, je valide.

#17 - 16 septembre 2021 17:49 - Benjamin Dauvergne

Thomas Noël a écrit :

Je vais proposer qu'on s'en passe carrément parce que c'est du caca.

Oui désolé de mon intervention intempestive sur ce ticket, c'est juste pour dire que j'ai envie de proposer un remplacement à uwsgidecorators dans un autre ticket (genre un module hobo.util.uwsgi); c'était juste pour voir si il y avait un assentiment général sur le sujet ou si tout le monde s'en fout.

#18 - 16 septembre 2021 17:56 - Thomas Noël

Benjamin Dauvergne a écrit :

Thomas Noël a écrit :

Je vais proposer qu'on s'en passe carrément parce que c'est du caca.

Oui désolé de mon intervention intempestive sur ce ticket, c'est juste pour dire que j'ai envie de proposer un remplacement à uwsgidecorators dans un autre ticket (genre un module hobo.utils.uwsgi); c'était juste pour voir si il y avait un assentiment général sur le sujet ou si tout le monde s'en fout.

Pas trop d'avis sur le sujet, mais c'est vrai que uwsgidecorators est assez loufoque. Ceci dit, si "ça marche"...

#19 - 16 septembre 2021 18:36 - Benjamin Dauvergne

Thomas Noël a écrit :

Pas trop d'avis sur le sujet, mais c'est vrai que uwsgidecorators est assez loufoque. Ceci dit, si "ça marche"...

Ben ça me fait un peu peur sur tous les autres projets où s'en sert sans voir que ça va planter un jour, on utilise pickle via pass_argument=True partout.

#20 - 16 septembre 2021 18:40 - Emmanuel Cazenave

- Statut changé de Solution validée à Résolu (à déployer)

```
commit d84fd70defb71fe59fd500ffd445cb71d01b121b
Author: Emmanuel Cazenave <ecazenave@entrouvert.com>
Date: Thu Sep 16 15:58:16 2021 +0200
```

```
provisionning: pass data to spooler function in body parameter (#56991)
```

```
The `data` can be very long, by spooler files cannot be longer than 64k
except for the body bytes parameter.
```

```
See https://uwsgi-docs.readthedocs.io/en/latest/Spooler.html#spool-files
```

```
Using the body parameters is only possible with pass_arguments=False (not documented).
```

#21 - 16 septembre 2021 18:51 - Emmanuel Cazenave

Si une bonne âme veut bien tagger/déployer demain ...

Et si ça merde encore ça me de reverter pour revenir à l'état provisionning HTTP synchrone.

Benjamin Dauvergne a écrit :

Je vais proposer qu'on s'en passe carrément parce que c'est du caca.

Je suis d'accord que ça inspire pas du tout confiance, mais probablement pas besoin de le virer complètement, s'en servir comme le fait wcs de façon très minimaliste, en passant juste un nom de tenant et un identifiant de job, avec un objet job qui peut se sérialiser/désérialiser complètement en dehors du magma de uwsgidecorators.

#22 - 16 septembre 2021 18:52 - Emmanuel Cazenave

Emmanuel Cazenave a écrit :

~~ça me de reverter~~

ça me va de reverter

#23 - 17 septembre 2021 01:15 - Thomas Noël

- Statut changé de Résolu (à déployer) à Solution déployée

Emmanuel Cazenave a écrit :

Si une bonne âme veut bien tagger/déployer demain ...

Taggé, déployé, passerelle redémarrée, le provisionning en http pour CD24 créé un spoolfile de 98k octets (versus 7k avant) qui est pris en charge aussitôt et effacé.

```
sept. 17 01:12:36 passerelle uwsgi[8477]: [spooler] written 98628 bytes to file /srv/nfs/lib/passerelle/spooler/392be64fae804959bd3ecc955e6656df/uwsgi_spoolfile_on_passerelle_8520_2_2060598390_1631833956_772310
sept. 17 01:12:36 passerelle uwsgi[8477]: [spooler /srv/nfs/lib/passerelle/spooler/392be64fae804959bd3ecc955e6656df pid: 8505] managing request uwsgi_spoolfile_on_passerelle_8520_2_2060598390_1631833956_772310 ...
sept. 17 01:12:37 passerelle uwsgi[8477]: [spooler /srv/nfs/lib/passerelle/spooler/392be64fae804959bd3ecc955e6656df pid: 8505] done with task uwsgi_spoolfile_on_passerelle_8520_2_2060598390_1631833956_772310 after 1 seconds
```

Parfait donc !

#24 - 02 décembre 2021 11:15 - Thomas Noël

- Lié à *Development #57019: implémenter notre propre uwsgidecorators ajouté*

Fichiers

0001-provisionnement-pass-data-to-spooler-function-in-body-.patch	2,25 ko	16 septembre 2021	Benjamin Dauvergne
0001-provisionnement-pass-data-to-spooler-function-in-body-.patch	3,16 ko	16 septembre 2021	Emmanuel Cazenave
0001-provisionnement-pass-data-to-spooler-function-in-body-.patch	3,59 ko	16 septembre 2021	Emmanuel Cazenave