

Passerelle - Development #73953

revoir par connecteur/endpoint le timeout des requêtes ?

29 janvier 2023 11:04 - Frédéric Péters

Statut:	En cours	Début:	29 janvier 2023
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposed:	Non		
Description			
Il y a possibilité après #59783 de définir au niveau du connecteur une valeur de timeout différente de settings.REQUESTS_TIMEOUT. C'est uniquement utilisé pour le connecteur BAN. Ça pourrait être utilisé davantage, ou on pourrait, peut-être plus rapide, réduire settings.REQUESTS_TIMEOUT (actuellement 25 secondes) et mettre une valeur plus élevée seulement là où ça se révèle nécessaire.			
Peut-être qu'il manque une possibilité d'override via les settings, par connecteur. (?)			

Historique

#2 - 30 janvier 2023 12:22 - Robot Gitea

- Statut changé de Nouveau à Solution proposée
- Assigné à mis à Thomas Noël

Thomas NOEL (tnoel) a ouvert une pull request sur Gitea concernant cette demande :

- URL : <https://gitea.entrouvert.org/entrouvert/passerelle/pulls/66>
- Titre : decrease requests default timeout to 9s ([#73953](#))
- Modifications : <https://gitea.entrouvert.org/entrouvert/passerelle/pulls/66/files>

#3 - 30 janvier 2023 12:24 - Thomas Noël

- Assigné à changé de Thomas Noël à Benjamin Dauvergne

J'ai toujours pensé ça aussi, trop secrètement sans doute.

Je propose sur la PR liée de passer à 9 secondes par défaut (ce qui permet aussi de faire 2 voire 3 appels sur une seule requête wcs ou combo).

Mais est-ce que le plan serait d'avoir un settings planqué façon settings.REQUESTS_SUBSTITUTION ([#73805](#)) ?

#4 - 30 janvier 2023 14:18 - Frédéric Péters

Je pense que c'est ma question :

Peut-être qu'il manque une possibilité d'override via les settings, par connecteur. (?)

Là si on passe ça et qu'en production on se rend compte d'un problème on a comme possibilité de remettre un timeout plus élevé qui s'appliquera partout, faire le changement pour le monter sur le code du connecteur problématique, mais il faut penser à retirer le paramétrage une fois que c'est déployé en production deux semaines plus tard, je crains qu'on oublie régulièrement.

#5 - 30 janvier 2023 15:49 - Thomas Noël

- Statut changé de Solution proposée à En cours
- Assigné à Benjamin Dauvergne supprimé

Frédéric Péters a écrit :

Je pense que c'est ma question :

Peut-être qu'il manque une possibilité d'override via les settings, par connecteur. (?)

Là si on passe ça et qu'en production on se rend compte d'un problème on a comme possibilité de remettre un timeout plus élevé qui

s'appliquera partout, faire le changement pour le monter sur le code du connecteur problématique, mais il faut penser à retirer le paramétrage une fois que c'est déployé en production deux semaines plus tard, je crains qu'on oublie régulièrement.

Ok, donc plutôt s'inspirer donc de la mécanique dans [#73805](#) avec un

```
REQUESTS_TIMEOUTS = {  
  'cmis/test': 5  
}
```

(noter le S sur TIMEOUTS, pour ne pas confondre avec le REQUEST_TIMEOUT déjà existant)

En même temps, ajouter un requests_timeout sur HTTPResource, qui donnera une UI pour paramétrer l'affaire sur les connecteurs utilisant HTTPResource.

(je ne m'assigne pas la chose, d'autres sujets en cours)

#6 - 31 janvier 2023 16:19 - Thomas Noël

[#73805](#) met finalement en place un settings.CONNECTORS_SETTINGS dans lequel on peut imaginer ajouter un requests_timeout.

On ajouterait donc ici cette possibilité :

```
CONNECTORS_SETTINGS = {  
  'cmis/test': {  
    'requests_timeout': 20,  
  }  
}
```

ou bien un peu de folie (mais ça pourrait venir dans un second temps) :

```
CONNECTORS_SETTINGS = {  
  'cmis/test': {  
    'requests_timeout': {  
      'https://www.example.net/slow-webservice': 20,  
      'https://www.example.net/slow-post': {'POST': 20},  
      'https://www.example.net/speedy': 1,  
    }  
  }  
}
```

en plus de passer le timeout par défaut à 5 secondes (c'est bien assez long, mais sinon 9s c'est joli car disruptif).