

Passerelle - Development #9416

endpoint /check (ou autre) pour vérif minimale du bon fonctionnement du service distant

21 décembre 2015 19:23 - Frédéric Péters

Statut:	Fermé	Début:	21 décembre 2015
Priorité:	Normal	Echéance:	
Assigné à:	Frédéric Péters	% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		
Description			
On a pour certains connecteurs un /ping, qui vérifie un appel simple au service distant. Je pense qu'il faudrait généraliser et standardiser ça, qu'on puisse ensuite balancer un check_http nagios dessus, pour obtenir gratuitement l'info de downtime de ces services.			

Révisions associées

Révision 840693f8 - 02 janvier 2018 08:44 - Frédéric Péters

general: add service availability basics (#9416)

Révision 1d84c51f - 02 janvier 2018 08:58 - Frédéric Péters

tests: add tests for availability parts (#9416)

Historique

#1 - 21 décembre 2015 19:23 - Frédéric Péters

- Tracker changé de Bug à Development

#2 - 22 décembre 2015 14:22 - Thomas Noël

Moi j'aime bien /ping.

JSON de la réponse "tout va bien":

```
{
  "data": {
    "ping": "pong"
  },
  "err": 0
}
```

#3 - 07 juillet 2016 20:19 - Frédéric Péters

Je pense que dans le travail de factorisation actuel, ça serait pas mal de faire revivre cette demande, d'un endpoint ping/check généralisé. Je me dis aussi que même sans nagios, on pourrait avoir en interne dans passerelle un système de suivi de disponibilité; qui au-delà du ping/check pourrait être alimenté par les appels réussis/ratés (genre quand un appel distant échoue on appelle le code ping/check pour faire la part des choses entre un bug de l'appel et une indisponibilité du système distant, et on en profite pour enregistrer l'info).

#4 - 27 septembre 2016 11:16 - Frédéric Péters

Basiquement ça pourrait être un bloc supplémentaire affiché sur la page du connecteur, avec les dernières indisponibilités :

- down le 27 septembre 2016 de 10h04 à 10h32
- etc.

#5 - 27 septembre 2016 11:16 - Frédéric Péters

- Assigné à mis à Jean-Baptiste Jaillet

#6 - 28 septembre 2016 18:22 - Frédéric Péters

Il y a différentes parties :

- Avoir dans tous les connecteurs (qui fonctionnent avec une appli distante) une fonction minimale de vérification de bon fonctionnement de celle-ci; par exemple dans le connecteur Agora+ il y a un appel qui demande juste un token; dans le connecteur base adresse il pourrait y avoir

un appel vers "http://api-adresse.data.gouv.fr/search/?q=169%20rue%20du%20chateau,%20paris&limit=1" et vérification du résultat, dans le connecteur gdc récupération de la liste des communes, etc. Le seul truc important c'est que ça n'ait pas d'effet de bord, on ne va pas créer de fausses demandes pour vérifier le bon fonctionnement. Pour commencer ici, il est bien sûr suffisant de juste assurer ça pour un ou deux connecteurs, pas besoin de tous les faire. base_adresse et gdc sont de bons candidats dans la mesure où ils sont déjà plutôt nouveau style.

- Il faut un modèle global, qui contiendrait au moins la référence vers un connecteur, un timestamp de début d'indispo, un timestamp de fin d'indispo.
- Une commande de management, qui sera lancée par un cron, qui pose un lock et appelle pour tous les connecteurs la méthode de vérif, comme il va surtout y avoir de la latence réseau, ils peuvent être lancés en tas en parallèle; qui ajoute une ligne en cas de nouvelle erreur (et pose le timestamp de fin d'indispo si pas d'erreur).
- L'ajout d'un bloc "Indisponibilités récentes" dans le template service_view.html (affiché uniquement aux personnes qui ont perms.base.view_accessright, je dirais)

Et en fait ça peut très bien s'arrêter à ça.

À la suite, dans BaseResource, on peut ajouter un endpoint ping, qui appelle la fonction ajoutée au-dessus, mais ce n'est même pas important.

Aussi, ce qu'il pourrait y avoir en plus, c'est dans GenericEndpointView de quoi automatiquement appeler le check en cas d'erreur, pour identifier l'indisponibilité dès la première erreur (et dans l'autre sens, en cas d'appel réussi alors que le service est marqué indispo, marquer que c'est revenu).

Peut-être, quand le service distant est marqué comme indisponible, imaginer que GenericEndpointView réponde plutôt par lui-même que le service est indispo, que le code du connecteur ne soit même pas appelé.

À voir aussi, faire en sorte que la fréquence de vérification augmente lors des périodes d'indisponibilités (genre ce serait toutes les dix minutes quand tout va a priori bien et toutes les minutes pour les connecteurs se déclarant en indispo).

En terme de vues, ensuite, un tas de trucs sont possibles aussi, afficher un graphe et/ou des stats de disponibilité (comme nagios, ex: <https://eudat.eu/sites/default/files/pictures/eudat3.png>).

À parler de nagios aussi, avoir une vue http /check générale, qui réponde simplement ok quand tous les services sont ok, et une erreur quand ce n'est pas le cas. (on n'est vraiment pas équipé pour le moment pour suivre dans nagios les connecteurs un par un, mais un check par passerelle, ça doit se faire).

Bref, pour ce ticket j'en resterais aux premiers points, le reste ça peut être des idées pour des tickets supplémentaires, après.

#7 - 29 septembre 2016 19:05 - Jean-Baptiste Jaillet

- Fichier 0001-endpoints-add-check-for-base_adresse-endpoint-9416.patch ajouté

Voilà. J'ai fait quelque chose pour base_adresse.

Je pose un patch maintenant pour savoir si c'est la bonne direction, et prendre maintenant les critiques pour le caler après pour gdc aussi.

#8 - 29 septembre 2016 19:07 - Jean-Baptiste Jaillet

je pense notamment à la requête envoyée pour tester le service.

Peut être faire (au lieu de juste != 200) un test si c'est 400 (indisponible) ou 500 (le serveur a merdé soit à cause de ce qu'on envoie, soit parce que le serveur fait de la merde)

#9 - 29 septembre 2016 19:07 - Jean-Baptiste Jaillet

enfin pour 400 indisponible ou l'adresse du service n'est plus à jour.

#10 - 29 septembre 2016 19:15 - Frédéric Péters

Les connecteurs en eux-mêmes ne doivent rien savoir de la classe EndpointCheckPing (qui paraît s'appeler ServiceAvailabilityReport, genre).

Dans le connecteur, la méthode elle doit juste vérifier la disponibilité et retourner vrai ou faux.

```
def availability_check(self):
    scheme, netloc, path, params, query, fragment = urlparse.urlparse(self.service_url)
    path = '/search/'
    query = urlencode({'q': 'rue du chateau', 'limit': 1})
    url = urlparse.urlunparse((scheme, netloc, path, params, query, fragment))
    result_response = self.requests.get(url)
    return len(result_response.json().get('features')) > 0
```

```
connector = models.CharField(max_length=50, verbose_name='Connector's slug')
```

Faut passer par ContentType, comme c'est fait dans le modèle AccessRight.

#11 - 29 septembre 2016 19:18 - Jean-Baptiste Jaillet

D'accord.

Du coup comment le modèle se met à jour (celui qui garde en base les derniers couacs debut / fin) ?

Il instancie un connecteur (paramètre du modèle) et lance la fonction availability?

#12 - 29 septembre 2016 20:07 - Frédéric Péters

→ Une commande de management, qui sera lancée par un cron, qui pose un lock et appelle pour tous les connecteurs la méthode de vérif [...]

cf la page d'accueil pour lister l'ensemble des connecteurs définis.

#13 - 02 octobre 2017 23:41 - Frédéric Péters

- Assigné à Jean-Baptiste Jaillet supprimé

#14 - 27 décembre 2017 11:26 - Frédéric Péters

- Fichier availability.png ajouté

- Fichier 0001-general-add-service-availability-basics-9416.patch ajouté

- Statut changé de Nouveau à En cours

- Assigné à mis à Frédéric Péters

- Patch proposed changé de Non à Oui

Reprise à zéro.

Ajout d'un suivi de la disponibilité d'un service, cron exécuté toutes les cinq minutes; pour le moment simplement utilisé pour afficher "down" sur la page du connecteur quand c'est le cas. Tests ajoutés aux connecteurs base adresse, fil atom/rss et opengis. (il s'agit simplement de poser une méthode qui lèvera une exception en cas d'erreur).

En perspective reprendre l'information sur la page d'accueil, avoir une vue détaillée du taux de disponibilité, intégrer à la gestion des erreurs (si un endpoint échoue, appeler le test de dispo pour voir dès ce moment si c'est pas juste tout le service qui est down, etc.). Tout cela dans de futurs tickets une fois l'infra en place.

#15 - 27 décembre 2017 19:35 - Thomas Noël

Je pense qu'on devrait lancer un self.availability() lors du save, car j'imagine le cas du service qui est "down" parce que l'URL est erronée, et on la modifie, mais il faut attendre 5 minutes avant que ça revienne (ou pas). Avec une vérification lors de la création/modification d'un connecteur, on éviterait ce cas (et on proposerait une technique pour "forcer" un check, pas trop cachée mais un peu quand même)

#16 - 31 décembre 2017 16:36 - Frédéric Péters

- Fichier 0001-general-add-service-availability-basics-9416.patch ajouté

J'ai préféré dupliquer ça au niveau des vues de création et d'édition, plutôt que faire ça dans le .save(), pour ne pas qu'une commande genre import_site se trouve aussi emmener toutes ces vérifications. (mais ça peut se discuter)

#17 - 31 décembre 2017 22:20 - Thomas Noël

Ça me va très bien, je n'aime pas trop les save surchargés, effets de bord effectivement à craindre.

Au niveau de l'utilisation dans le template :

```
{% with status=object.get_availability_status %}
  {% if status.down %}<span class="down" title="{{status.message}} {% trans 'since:' %} {{status.start_times
tamp|date:"SHORT_DATETIME_FORMAT"}} ">{% trans 'Down' %}</span>{% endif %}
  {% endwith %}
```

J'aimerais bien que tu ajoutes un if status qui rappelle qu'il n'y a pas forcément de status, soit parce qu'il n'a pas encore été joué après un import, soit parce que le connecteur n'a pas de testeur (c'est vraiment juste pour s'en rappeler, la lecture du code est assez explicite)

Et à penser à ça, est-ce qu'on ajouterait pas l'info "OK" quand status.up (et donc, on enregistrerait la date du dernier succès dans le availability()) ?

Sauf si tu as des idées sur ces points là, c'est déjà un ack ainsi -- on pourra faire évoluer cette base à l'usage.

#18 - 02 janvier 2018 08:46 - Frédéric Péters

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

J'aimerais bien que tu ajoutes un if status qui rappelle qu'il n'y a pas forcément de status, soit parce qu'il n'a pas encore été joué après un

import, soit parce que le connecteur n'a pas de testeur (c'est vraiment juste pour s'en rappeler, la lecture du code est assez explicite)

Ajouté.

Et à penser à ça, est-ce qu'on ajouterait pas l'info "OK" quand status.up

Je suis en fait encore hésitant, dans un premier temps je resterais à "down = on est sûr en face que c'est mort" et rien afficher autrement. Cela étant dans #20901 je me contredis déjà en affichant une marque sur la page d'accueil quand le service apparait comme disponible (mais je pense que je devrais la retirer).

(et donc, on enregistrerait la date du dernier succès dans le availability()) ?

Ça pourrait facilement évoluer pour enregistrer la date de dernière vérification mais ça me va bien pour le moment d'écrire dans la db uniquement en cas de changement.

Sauf si tu as des idées sur ces points là, c'est déjà un ack ainsi -- on pourra faire évoluer cette base à l'usage.

Je pousse ainsi et clairement ça pourra évoluer.

```
commit 840693f81972838ffd5f8627c04bc3b0791c1cb9
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Wed Dec 27 11:17:20 2017 +0100
```

```
general: add service availability basics (#9416)
```

#19 - 02 janvier 2018 08:59 - Frédéric Péters

Et du commit j'avais oublié le fichier avec les tests; commit séparé :

```
commit 1d84c51fb10a9efa8de903ec331e1a2ed8647e31
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Tue Jan 2 08:58:16 2018 +0100
```

```
tests: add tests for availability parts (#9416)
```

#20 - 04 août 2018 12:33 - Benjamin Dauvergne

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-endpoints-add-check-for-base_adresse-endpoint-9416.patch	4,02 ko	29 septembre 2016	Jean-Baptiste Jaillet
availability.png	35 ko	27 décembre 2017	Frédéric Péters
0001-general-add-service-availability-basics-9416.patch	9,94 ko	27 décembre 2017	Frédéric Péters
0001-general-add-service-availability-basics-9416.patch	10,9 ko	31 décembre 2017	Frédéric Péters