

Publik - Development #28439

Pouvoir utiliser du JSON path sur des données JSON

28 novembre 2018 17:35 - Benjamin Dauvergne

Statut: Nouveau	Début: 28 novembre 2018
Priorité: Normal	Echéance:
Assigné à:	% réalisé: 0%
Catégorie:	Temps estimé: 0:00 heure
Version cible:	Club: Non
Patch proposed: Non	
Planning:	

Description

J'ai eu le besoin récemment par mon CPF (connecteur ATOS-Genesys, [#28381](#)) de transformer les données d'un web-service pour les besoins d'un template de

```
[{'xxx': yyy, ..}, {'xxx': zzz, ..}]
```

vers

```
{'yyy': [{'xxx': yy, ...}], 'zzz': [{'xxx': zzz, ...}]}
```

pour permettre de traiter les données de type yyy séparément des données de type zzz, notamment pour le CD06 de les compter

Ça me semble être des manipulations courante coté template et formulaires et je propose qu'on ajoute de quoi manipuler ces structures via JSONPath pour éviter un travail un peu inutile et répétitif coté connecteur (une fois les données exposées le travail devrait être terminé).

Pour cela je propose deux choses:

- pour les templates combo et les conditions Django de w.c.s., un filtre de template de la forme

```
|jsonpath:"expression"
```

- pour les conditions python dans w.c.s. une bête fonction

```
jsonpath(data, expression)
```

Sur cet exemple la solution serait

```
{% if data|jsonpath:"${?(@.xxx == "yyy")}"|length %}...{% endif %}
```

Demandes liées:

Lié à Passerelle - Autre #4193: Possibilité de filtrer une source json par JM...

Nouveau

04 janvier 2014

Lié à Passerelle - Development #9211: connecteur json in/json out

Nouveau

04 décembre 2015

Historique

#1 - 28 novembre 2018 20:21 - Benjamin Dauvergne

- Description mis à jour

#2 - 28 novembre 2018 20:23 - Benjamin Dauvergne

- Description mis à jour

#3 - 28 novembre 2018 20:41 - Benjamin Dauvergne

jmespath (<http://jmespath.org/>) semble plus avancé et bien supporté (présent dans stretch), avec jmespath la solution serait:

[?xxx == 'yyy']

#4 - 28 novembre 2018 20:49 - Frédéric Péters

Ça me semble être des manipulations courante coté template et formulaires

Avant l'introduction d'un nouveau langage, ça m'intéresserait d'avoir de solides/nombreux exemples pour le justifier.

Ça remplacerait quelles constructions aujourd'hui mises en œuvre ?

#5 - 28 novembre 2018 20:55 - Frédéric Péters

- Lié à Autre #4193: Possibilité de filtrer une source json par JMESPath ajouté

#6 - 28 novembre 2018 20:56 - Frédéric Péters

- Lié à Development #9211: connecteur json in/json out ajouté

#7 - 28 novembre 2018 21:04 - Frédéric Péters

J'ai mis en lien deux vieux tickets Passerelle, parce qu'instinctivement je préférerais voir les évolutions là, que Passerelle gagne des capacités de transformation de données, qu'on puisse, pour tous les connecteurs et pas seulement le connecteur tableur, définir des requêtes, etc.

Ça n'invite pas dans w.c.s. une nouvelle façon d'accéder aux attributs, ça me semble moins sujet aux débordements de fantaisies.

#8 - 29 novembre 2018 10:05 - Benjamin Dauvergne

Frédéric Péters a écrit :

Ça me semble être des manipulations courante coté template et formulaires

Avant l'introduction d'un nouveau langage, ça m'intéresserait d'avoir de solides/nombreux exemples pour le justifier.

Ça remplacerait quelles constructions aujourd'hui mises en œuvre ?

C'est fait pour remédier au manque de possibilité de parcourir les données dans Django, on peut seulement suivre un chemin simple (x.y.0.j.etc..) mais pas extraire des choses profondes sous forme de listes, principalement (c'est l'exemple ici). Pour ne pas n'avoir qu'un seul exemple j'inviter justement les CPF à rapporter des choses qui leur semblaient proche, Stéphane a évoqué Vénissieux mais il n'a pas encore rapporté le cas concret ici.

#9 - 29 novembre 2018 10:22 - Benjamin Dauvergne

Frédéric Péters a écrit :

J'ai mis en lien deux vieux tickets Passerelle, parce qu'instinctivement je préférerais voir les évolutions là, que Passerelle gagne des capacités de transformation de données, qu'on puisse, pour tous les connecteurs et pas seulement le connecteur tableur, définir des requêtes, etc.

Aucun des deux tickets ne donne de solution au problème que j'ai eu; à voir pour le problème de Stéphane à Vénissieux.

Ça n'invite pas dans w.c.s. une nouvelle façon d'accéder aux attributs, ça me semble moins sujet aux débordements de fantaisies.

Ça me paraît beaucoup moins pratique pour plusieurs raisons:

1. les connecteurs peuvent exposer des données sur différents endpoint, ça demanderait à avoir une vision structurée de ces endpoints (bon oui on a @generic_endpoint) et à stocker pour ces différents endpoint une liste de transformations à appliquer (c'est déjà beaucoup plus compliqué que de juste extraire via un tag dans un template); ça c'est si on est d'accord que c'est quelque chose de dynamique, défini par l'administrateur; sinon ça passerait par un connecteur proxy qui appellerait une autre connecteur pour y appliquer des transformations, je ne trouve pas ça pratique non plus et un peu usine à gaz,
2. si ta vision est plutôt d'avoir un outil en plus à utiliser dans le code des connecteurs (comme on a la couche soap/zeep, les logger, etc..) pour nous simplifier la vie, je n'en vois pas l'utilité immédiate, ce n'est pas une grande amélioration de notre productivité de passer de compréhension de liste ou de boucle for à du jmespath (pour les relectures c'est peut-être un gain); mais si je prends l'exemple en question, pour simplifier la vie de Mike j'ai créé un dico indexé par une des clés des dicos de demandes/droits chose que jmespath ne sait pas faire de toute façon;
3. ça remet en cause une séparation assez simple entre connecteurs domaine des CPT et template/workflow domaine des CPF et autres administrateurs fonctionnels chez nos clients; mais je peux comprendre aussi que dans les connecteurs CSV ça amène une deuxième façon de faire les choses: plutôt que de filtrer dans une query, on peut aussi tout récupérer et filtrer via jsonpath. Pour moi c'est une question de documentation que d'expliquer les usages à préférer;
4. la configuration des connecteurs deviendra un fourre-tout où on trouvera toute sorte de transformations destinées à des usages en aval sur lesquels on aura pas de visibilité (pourquoi on fait ça? ha oui c'est utilisé pour la génération d'un document ODT ailleurs), niveau "separation of concerns" ce n'est pas terrible

Pour moi c'est mieux si la transformation reste au plus près des consommateurs, et ne remonte que si vraiment c'est générique; après on pourra toujours trouver des cas ou c'est faux mais je pense qu'on les traite déjà. Par exemple un endpoint qui en appelle plusieurs coté application métier et présente une vue uniforme des choses, ce genre de transformation a toute sa place dans un connecteur.

#10 - 29 novembre 2018 10:29 - Frédéric Péters

mais si je prends l'exemple en question, pour simplifier la vie de Mike j'ai créé un dico indexé par une des clés des dicos de demandes/droits chose que jmespath ne sait pas faire de toute façon;

Mais donc l'exemple donné pour justifier l'idée n'est même pas réel ?

~~

Ce n'est pas nouveau, je suis pour que les connecteurs développés ne le soient pas en vase clos, qu'ils soient développés avec l'usage Publik en tête, qu'ils exposent ainsi des données directement et facilement exploitables dans nos briques.

#11 - 29 novembre 2018 13:52 - Benjamin Dauvergne

Frédéric Péters a écrit :

mais si je prends l'exemple en question, pour simplifier la vie de Mike j'ai créé un dico indexé par une des clés des dicos de demandes/droits chose que jmespath ne sait pas faire de toute façon;

Mais donc l'exemple donné pour justifier l'idée n'est même pas réel ?

Si c'est l'usage réel, c'est juste que coté connecteur le plus simple a été de séparer les demandes et droits selon leur type parce qu'il était impossible de faire ce filtrage coté template pour ensuite vérifier si il était vide.

Ce n'est pas nouveau, je suis pour que les connecteurs développés ne le soient pas en vase clos, qu'ils soient développés avec l'usage Publik en tête, qu'ils exposent ainsi des données directement et facilement exploitables dans nos briques.

Là j'ai orienté le connecteur pour séparer demandes et droit APA et PH, si on revient sur cette décision et on décide d'afficher la totalité des demandes sans distinction APA et PH et de les trier par date ou je ne sais quel critère ce ne sera pas possible, il faudra encore modifier le connecteur. Ce que tu demandes n'est pas possible en fait, on ne peut pas écrire un connecteur qui marchera tout le temps pour tout le monde avec les outils en place coté consommation (notamment les templates Django).

Mais bon on peut discuter sans fin de manière générale, je ne ferai de toute façon rien sans plus d'exemples.

#12 - 03 décembre 2018 10:22 - Benjamin Dauvergne

Encore un besoin qui serait résolu par json/jmes-path, toujours sur CD06, <https://dev.entrouvert.org/issues/28393#note-14>.

```
{% if json|jmespath:"data.dossier.DEMANDES[?etape == "Décision d'accord"]|length %}@
```

#13 - 28 février 2019 15:04 - Benjamin Dauvergne

- *Tracker changé de Support à Development*