

4. API REST - EXISTING / NOT EXISTING FIELDS

[NGL-2931](#) :

Ekrame Jacoby a ajouté un commentaire - 30/avr./20 2:49 PM

Il y a effectivement deux façons dans NGL pour filtrer la présence/absence d'une propriété et les deux sont correctes.

Dans SQ (Container, Sample) : un seul champ pour tester la présence=true ou absence=false de la propriété

```
Map<String, Boolean> existingFields
```

Ce qui permet d'avoir des filtres d'existence dans les filteringConfig du type :

```
form["existingFields[propriété]"]
```

qui peut avoir la valeur true et false.

Dans BI et Project (Project, ReadSet, Analyses, Run) : deux champs pour tester la présence (existingFields) ou absence (notExistingFields) des propriétés

```
Set<String> existingFields, notExistingFields
```

Utilisé généralement par les bioinfos pour rechercher des ressources qui présente ou ne présente pas une liste de propriété. Mais le champ existingField ne peut pas être utilisé dans des filtres dynamique avec choix valeur=true/false d'où l'utilisation de la map dans SQ.

Donc je pense que c'est historique, au départ c'était codé comme dans BI et Project puis évolution dans SQ au vu des besoins dans FilteringConfig.

existingFields et **notExistingFields** sont mis à disposition depuis un moment, on ne peut pas changer leur signature sans risquer d'impacter les bioinfos. Je propose de créer un nouveau champs mapExistingFields pour permettre de créer des filtres sur l'existence de la propriété dans Project. Et si le besoin est exprimé sur BI alors on fera de la même manière.

Ce serait bien de faire une doc xwiki sur les reportingConfig/filteringConfig afin de noter toutes ces infos et surtout de lister les règles que l'on met en place

URL directement ds NGL-BI

exemples :

[http://ngl-bi.genoscope.cns.fr/readsets/search/home?
sampleCategoryCodes=RNA&sampleCategoryCodes=cDNA](http://ngl-bi.genoscope.cns.fr/readsets/search/home?sampleCategoryCodes=RNA&sampleCategoryCodes=cDNA)

<http://appuat.genoscope.cns.fr:9004/runs/search/home?categoryCode=illumina>

<http://appuat.genoscope.cns.fr:9004/runs/search/home?treatmentProperties%5BminknowMetrichor.default%5D%5BmetrichorWorkflowName%5D=2D%20Basecalling%20RNN%20for%20NSK007>

pour rechercher sur la propriété : **treatments.minknowMetrichor.default.metrichorWorkflowName=...**

treatmentProperties[minknowMetrichor.default][metrichorWorkflowName]=2D Basecalling RNN for NSK007

%20 : espace

%5B : [

%5D :]

URL EXISTING / not existing fields dans BI :

exemples :

[readsets/search/home?
existingFields=treatments.duplicatesRaw¬ExistingFields=treatments.duplicatesRaw.read1.estimateDuplicatedReadsNTimes.value](http://ngl-bi.genoscope.cns.fr/readsets/search/home?existingFields=treatments.duplicatesRaw¬ExistingFields=treatments.duplicatesRaw.read1.estimateDuplicatedReadsNTimes.value)

<http://appuat.genoscope.cns.fr:9004/readsets/search/home?existingFields=treatments.ngsrg.default.1DReverse>

4. API REST - EXISTING / NOT EXISTING FIELDS

<http://appuat.genoscope.cns.fr:9004/runs/search/home?existingFields=treatments.ngsrg.default.1DReverse>

NB : existing field utilisé (autrement) dans SQ recherche containers par exemple dans les filtres supplémentaires :

```
{
  "property": "contents.properties.depletionMethod",
  "html": "<label>Déplétion ARNr ? <input type='checkbox' ng-
model='searchService.form[\"existingFields[contents.properties.depletionMethod]\"]' ng-true-value='true' ng-false-
value='\"'/></label>",
},
{
  "property": "properties.limsCode",
  "html": "<div class='form-control' bt-select ng-
model='searchService.form[\"existingFields[properties.limsCode]\"]' placeholder='Importé du lims ?' bt-
options='v.code as v.name for v in searchService.lists.get(\"booleans\")'></div>",
},
}
```