

## BiJoe - Development #50381

### Améliorations des performances

21 janvier 2021 09:25 - Mikaël Ates

<b>Statut:</b>	Nouveau	<b>Début:</b>	21 janvier 2021
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>		<b>% réalisé:</b>	0%
<b>Catégorie:</b>		<b>Temps estimé:</b>	0:00 heure
<b>Version cible:</b>		<b>Planning:</b>	Non
<b>Patch proposed:</b>	Non		
<b>Description</b>			
Les requêtes sur une base d'environ 100 000 demandes sont longues et rendent l'exploitation difficile. Peut-être que ce temps est tout à fait honorable considérant l'opération à conduire ou peut-être y a-t-il moyen de gagner en performance ?			
<ul style="list-style-type: none"><li>"Depuis l'origine, délai de traitement moyen par formulaire" : <a href="https://statistiques.mesdemarches06.fr/visualization/106/">https://statistiques.mesdemarches06.fr/visualization/106/</a> : 26s</li><li>"Depuis l'origine, nombre de demandes par formulaire" : <a href="https://statistiques.mesdemarches06.fr/visualization/105/">https://statistiques.mesdemarches06.fr/visualization/105/</a> : 32s</li></ul>			
<b>Demandes liées:</b>			
Lié à BiJoe - Development #51719: optimisation: ne pas générer/générer de man...		<b>Nouveau</b>	<b>05 mars 2021</b>

### Historique

#### #2 - 21 janvier 2021 13:20 - Benjamin Dauvergne

Oui ça doit être optimisable parce que fait à la main ça prend 70 millisecondes pour la première et encore moins pour la seconde :

```
bijoe=# select formdef_id, date_trunc('month', receipt_time) as month, avg(endpoint_delay) from formdata group
  by formdef_id, date_trunc('month', receipt_time) order by formdef_id, month;
Temps : 73,428 ms
```

```
bijoe=# select formdef_id, count(*) from formdata group by formdef_id;
Temps : 23,935 ms
```

Je crains que le temps ne soit pris en fait par autre chose, il faut regarder le postgres pour voir les requêtes lentes lors de l'affichage d'un rapport, mais ce n'est pas la requête principale pour les deux cas soumis. Sur des requêtes simples de ce genre on ne devrait pas voir de temps de plus de quelques secondes avant 1 bon million de demandes.

#### #3 - 21 janvier 2021 13:24 - Benjamin Dauvergne

Le temps ne vient pas des requêtes mais du code de l'application, le log de toutes les requêtes pour visualiser <https://statistiques.mesdemarches06.fr/visualization/106/> :

```
2021-01-21 13:22:02.687 CET [3950] bijoe@bijoe LOG: duration: 138.125 ms statement: select distinct jsonb_ob
ject_keys(json_data) as a from formdata order by a
2021-01-21 13:22:02.794 CET [3953] bijoe@bijoe LOG: duration: 67.284 ms statement: SELECT channel.id AS valu
e, channel.label::text AS label FROM formdata RIGHT OUTER JOIN channel AS "channel" ON formdata.channel_id =
"channel".id GROUP BY channel.id, channel.label ORDER BY (channel.id)
2021-01-21 13:22:02.891 CET [3954] bijoe@bijoe LOG: duration: 76.507 ms statement: SELECT category.ref AS va
lue, category.label::text AS label FROM formdata RIGHT OUTER JOIN (formdef AS "formdef" RIGHT OUTER JOIN cat
egory AS "category" ON formdef.category_id = "category".id) ON formdata.formdef_id = "formdef".id GROUP BY cat
egory.ref, category.label ORDER BY (category.label)
2021-01-21 13:22:02.988 CET [3955] bijoe@bijoe LOG: duration: 76.747 ms statement: SELECT formdef.ref AS val
ue, formdef.label::text AS label FROM formdata RIGHT OUTER JOIN formdef AS "formdef" ON formdata.formdef_id =
"formdef".id GROUP BY formdef.ref, formdef.label ORDER BY (formdef.label)
2021-01-21 13:22:03.075 CET [3956] bijoe@bijoe LOG: duration: 65.802 ms statement: SELECT generic_status.id
AS value, generic_status.label::text AS label FROM formdata RIGHT OUTER JOIN status AS "generic_status" ON fo
rmdata.generic_status_id = "generic_status".id GROUP BY generic_status.id, generic_status.label ORDER BY (gene
ric_status.id)
2021-01-21 13:22:04.056 CET [3971] bijoe@bijoe LOG: duration: 65.093 ms statement: SELECT DISTINCT json_data
->>'activite_1_classe_college' AS v, json_data->>'activite_1_classe_college' AS v FROM formdata WHERE (json_da
ta->>'activite_1_classe_college') IS NOT NULL ORDER BY v
2021-01-21 13:22:08.296 CET [4038] bijoe@bijoe LOG: duration: 60.688 ms statement: SELECT DISTINCT json_data
->>'association_ville' AS v, json_data->>'association_ville' AS v FROM formdata WHERE (json_data->>'associatio
n_ville') IS NOT NULL ORDER BY v
2021-01-21 13:22:09.743 CET [4069] bijoe@bijoe LOG: duration: 60.127 ms statement: SELECT DISTINCT json_data
->>'CategorieSituation' AS v, json_data->>'CategorieSituation' AS v FROM formdata WHERE (json_data->>'Categori
eSituation') IS NOT NULL ORDER BY v
```

```

2021-01-21 13:22:10.068 CET [4076] bijoe@bijoe LOG: duration: 64.855 ms statement: SELECT DISTINCT json_data
->>'civilite' AS v, json_data->>'civilite' AS v FROM formdata WHERE (json_data->>'civilite') IS NOT NULL ORDER
BY v
2021-01-21 13:22:16.438 CET [4190] bijoe@bijoe LOG: duration: 85.870 ms statement: SELECT DISTINCT json_data
->>'distribution_traitement' AS v, json_data->>'distribution_traitement' AS v FROM formdata WHERE (json_data->
>'distribution_traitement') IS NOT NULL ORDER BY v
2021-01-21 13:22:17.960 CET [4234] bijoe@bijoe LOG: duration: 66.122 ms statement: SELECT DISTINCT json_data
->>'effectif_lycee' AS v, json_data->>'effectif_lycee' AS v FROM formdata WHERE (json_data->>'effectif_lycee')
IS NOT NULL ORDER BY v
2021-01-21 13:22:18.291 CET [4251] bijoe@bijoe LOG: duration: 64.126 ms statement: SELECT DISTINCT json_data
->>'effectif_moyen_exo_en_cours_autres' AS v, json_data->>'effectif_moyen_exo_en_cours_autres' AS v FROM formd
ata WHERE (json_data->>'effectif_moyen_exo_en_cours_autres') IS NOT NULL ORDER BY v
2021-01-21 13:22:20.079 CET [4287] bijoe@bijoe LOG: duration: 63.042 ms statement: SELECT DISTINCT json_data
->>'id_rdv' AS v, json_data->>'id_rdv' AS v FROM formdata WHERE (json_data->>'id_rdv') IS NOT NULL ORDER BY v
2021-01-21 13:22:21.494 CET [4333] bijoe@bijoe LOG: duration: 60.557 ms statement: SELECT DISTINCT json_data
->>'LocalisationNidBtqAdresse' AS v, json_data->>'LocalisationNidBtqAdresse' AS v FROM formdata WHERE (json_da
ta->>'LocalisationNidBtqAdresse') IS NOT NULL ORDER BY v
2021-01-21 13:22:25.684 CET [4433] bijoe@bijoe LOG: duration: 78.789 ms statement: SELECT DISTINCT json_data
->>'objet' AS v, json_data->>'objet' AS v FROM formdata WHERE (json_data->>'objet') IS NOT NULL ORDER BY v
2021-01-21 13:22:31.483 CET [4528] bijoe@bijoe LOG: duration: 62.368 ms statement: SELECT DISTINCT json_data
->>'tu' AS v, json_data->>'tu' AS v FROM formdata WHERE (json_data->>'tu') IS NOT NULL ORDER BY v
2021-01-21 13:22:31.709 CET [4531] bijoe@bijoe LOG: duration: 60.876 ms statement: SELECT DISTINCT json_data
->>'type_activite' AS v, json_data->>'type_activite' AS v FROM formdata WHERE (json_data->>'type_activite') IS
NOT NULL ORDER BY v
2021-01-21 13:22:38.528 CET [4582] bijoe@bijoe LOG: duration: 74.005 ms statement: SELECT formdef.ref AS val
ue, formdef.label::text AS label FROM formdata RIGHT OUTER JOIN formdef AS "formdef" ON formdata.formdef_id =
"formdef".id GROUP BY formdef.ref, formdef.label ORDER BY (formdef.label)

```

Rien au dessus de 100 ms et 18 requêtes, ça devrait répondre en moins de 3/4 secondes si le code djang bijoe fait son boulot correctement.

PS: c'est lent même quand tout est en cache.

#### #4 - 04 mars 2021 19:44 - Frédéric Péters

et 18 requêtes

Il a été noté plus tard que seules les requêtes "lentes" étaient logguées (log\_min\_duration\_statement = 60).

#### #5 - 05 mars 2021 12:59 - Benjamin Dauvergne

Frédéric Péters a écrit :

et 18 requêtes

Il a été noté plus tard que seules les requêtes "lentes" étaient logguées (log\_min\_duration\_statement = 60).

Ça ne change pas grand chose au diagnostic, le temps est pris dans la présentation des données, certainement un mix de calcul dans l'appli et d'un trop grand nombre de requêtes. L'obtention des données elle même est rapide, c'est plutôt du côté du cache où de l'optimisation de la récupération des méta-données (liste des valeurs pour les dimensions) qu'il faut regarder.

À regarder le cas de Metz ça utilise la vieille méthode par iframe, ça irait peut-être mieux en passant par un graphique intégré à combo vue que la vue iframe calcule le formulaire de la sidebar pour rien. Ça rejoint aussi les demandes de passe par un sélecteur JSONP pour les champs de filtre ayant beaucoup de valeurs.

#### #6 - 05 mars 2021 14:36 - Benjamin Dauvergne

- Lié à Development #51719: optimisation: ne pas générer/générer de manière paresseuse le formulaire du cube dans les vues iframe ajouté

#### #7 - 13 avril 2021 18:34 - Emmanuel Cazenave

Benjamin Dauvergne a écrit :

À regarder le cas de Metz ça utilise la vieille méthode par iframe, ça irait peut-être mieux en passant par un graphique intégré à combo vue que la vue iframe calcule le formulaire de la sidebar pour rien.

Je viens de faire cette tentative et timeout aussi.