

Chrono - Bug #74001

api fillslot → DataError: integer out of range

31 janvier 2023 09:53 - Sentry Io

Statut:	Fermé	Début:	31 janvier 2023
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	Non
Patch proposé:	Non		
Description			
https://sentry.entrouvert.org/entrouvert/publik/issues/104999/			
NumericValueOutOfRangeException: integer out of range			
Traceback:			
File "/usr/lib/python3/dist-packages/django/db/backends/utils.py" in _execute 84. return self.cursor.execute(sql, params)			
The above exception (integer out of range) was the direct cause of the following exception:			
File "/usr/lib/python3/dist-packages/django/core/handlers/exception.py" in inner 34. response = get_response(request)			
File "/usr/lib/python3/dist-packages/django/core/handlers/base.py" in _get_response 115. response = self.process_exception_by_middleware(e, request)			
File "/usr/lib/python3/dist-packages/django/core/handlers/base.py" in _get_response 113. response = wrapped_callback(request, *callback_args, **callback_kwargs)			
File "/usr/lib/python3/dist-packages/django/views/decorators/csrf.py" in wrapped_view 54. return view_func(*args, **kwargs)			
File "/usr/lib/python3/dist-packages/django/views/generic/base.py" in view 71. return self.dispatch(request, *args, **kwargs)			
File "/usr/lib/python3/dist-packages/rest_framework/views.py" in dispatch 509. response = self.handle_exception(exc)			
File "/usr/lib/python3/dist-packages/rest_framework/views.py" in handle_exception 469. self.raise_uncaught_exception(exc)			
File "/usr/lib/python3/dist-packages/rest_framework/views.py" in raise_uncaught_exception 480. raise exc			
File "/usr/lib/python3/dist-packages/rest_framework/views.py" in dispatch 506. response = handler(request, *args, **kwargs)			
File "/usr/lib/python3/dist-packages/chrono/api/views.py" in post 1671. return self.fillslot()			
File "/usr/lib/python3/dist-packages/chrono/api/views.py" in fillslot 1574. event.save()			
File "/usr/lib/python3/dist-packages/chrono/agendas/models.py" in save 1514. return super().save(*args, **kwargs)			
File "/usr/lib/python3/dist-packages/django/db/models/base.py" in save 743. self.save_base(using=using, force_insert=force_insert,			

```

File "/usr/lib/python3/dist-packages/django/db/models/base.py" in save_base
780.             updated = self._save_table(

File "/usr/lib/python3/dist-packages/django/db/models/base.py" in _save_table
873.             result = self._do_insert(cls._base_manager, using, fields, update_pk, raw)

File "/usr/lib/python3/dist-packages/django/db/models/base.py" in _do_insert
910.             return manager._insert([self], fields=fields, return_id=update_pk,

File "/usr/lib/python3/dist-packages/django/db/models/manager.py" in manager_method
82.             return getattr(self.get_queryset(), name)(*args, **kwargs)

File "/usr/lib/python3/dist-packages/django/db/models/query.py" in _insert
1186.            return query.get_compiler(using=using).execute_sql(return_id)

File "/usr/lib/python3/dist-packages/django/db/models/sql/compiler.py" in execute_sql
1377.            cursor.execute(sql, params)

File "/usr/lib/python3/dist-packages/django/db/backends/utils.py" in execute
99.            return super().execute(sql, params)

File "/usr/lib/python3/dist-packages/sentry_sdk/integrations/django/__init__.py" in execute
446.            return real_execute(self, sql, params)

File "/usr/lib/python3/dist-packages/django/db/backends/utils.py" in execute
67.            return self._execute_with_wrappers(sql, params, many=False, executor=self._execute)

File "/usr/lib/python3/dist-packages/django/db/backends/utils.py" in _execute_with_wrappers
76.            return executor(sql, params, many, context)

File "/usr/lib/python3/dist-packages/django/db/backends/utils.py" in _execute
84.            return self.cursor.execute(sql, params)

File "/usr/lib/python3/dist-packages/django/db/utils.py" in __exit__
89.            raise dj_exc_value.with_traceback(traceback) from exc_value

File "/usr/lib/python3/dist-packages/django/db/backends/utils.py" in _execute
84.            return self.cursor.execute(sql, params)

Exception Type: DataError at /api/agenda/retrait-pieces-identite-hdv-03-2023/fillslot/retrait-2-pe
rsonnes:2023-02-15-1530/
Exception Value: integer out of range

```

Demandes liées:

Lié à Chrono - Development #74008: Event: changer le model pour avoir un id B...	Fermé	31 janvier 2023
Lié à Chrono - Development #74037: ne pas actualiser les récurrences toutes l...	Rejeté	31 janvier 2023

Historique

#1 - 31 janvier 2023 09:53 - Frédéric Péters

- *Projet changé de Suivi des traces à Chrono*

#8 - 31 janvier 2023 11:30 - Lauréline Guérin

- *Lié à Development #74008: Event: changer le model pour avoir un id BigAutoField ajouté*

#12 - 31 janvier 2023 17:33 - Pierre Ducroquet

Alors les ids s'emballent a priori à cause du comportement de chrono avec django.

La requête utilisée pour insérer les occurrences est, grosso modo, INSERT INTO agendas_event (...) VALUES (...) ON CONFLICT DO NOTHING;

Côté PG, cette requête est exécutée comme suit:

- 1) analyse du VALUES pour remplir en mémoire une structure correspondant à agendas_event, en valorisant ce qui doit l'être à l'aide des DEFAULT de la table
- 2) insertion dans la table et l'index, et en cas de conflit ne rien faire.

Sauf que la séquence est manipulée à la première étape, et elle n'est jamais remise à zéro derrière (principe d'une séquence).

Démo sur un exemple tout simple, une table avec une colonne id serial et une colonne a unique:

```
test=> insert into test_serial (a) values (3) on conflict do nothing; select * from test_serial_id_seq ;
INSERT 0 0
 last_value | log_cnt | is_called
-----+-----+-----
          25 |        16 | t
(1 row)
```

```
test=> insert into test_serial (a) values (4) on conflict do nothing; select * from test_serial_id_seq ;
INSERT 0 1
 last_value | log_cnt | is_called
-----+-----+-----
          26 |        15 | t
(1 row)
```

```
test=> insert into test_serial (a) values (4) on conflict do nothing; select * from test_serial_id_seq ;
INSERT 0 0
 last_value | log_cnt | is_called
-----+-----+-----
          27 |        14 | t
(1 row)
```

```
test=> insert into test_serial (a) values (4) on conflict do nothing; select * from test_serial_id_seq ;
INSERT 0 0
 last_value | log_cnt | is_called
-----+-----+-----
          28 |        13 | t
(1 row)
```

Le serial est incrémenté sans insertion.

Je n'ai pas d'autre explication pour les sauts d'ids. Je suppose que les ids ont augmenté progressivement au fil des tentatives d'insertions de récurrences (je ne sais à quelle fréquence), grâce au bug de l'an 2099...

#13 - 31 janvier 2023 17:54 - Frédéric Péters

En pratique,

```
chrono=# select id from agendas_event order by id desc limit 5;
 id
-----
2148976798
2148970261
2148944116
2148944115
2148937578
(5 lignes)
```

on a une possibilité, sur l'instance en question, pour revenir à des itérations qui ne soient pas des incréments de milliers ou plus ?

#14 - 31 janvier 2023 18:12 - Frédéric Péters

- Lié à Development #74037: ne pas actualiser les récurrences toutes les cinq minutes ajouté

#15 - 15 février 2023 18:15 - Valentin Deniaud

- Statut changé de Nouveau à Fermé

Résolu via [#74105](#).