

# Chrono - Bug #89040

## Des tests en erreurs en local

03 avril 2024 16:28 - Yann Weber

<b>Statut:</b>	Solution déployée	<b>Début:</b>	03 avril 2024
<b>Priorité:</b>	Normal	<b>Echéance:</b>	
<b>Assigné à:</b>	Yann Weber	<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>	Oui		

### Description

Alors que sur jenkins tout va bien, sur mon devinst local (j'ai MàJ devinst avec un git pull & make install, et j'ai lancé les migrations avec chrono-manage migrate\_schemas), avec un nombre d'erreur/fails différents selon le nombre de processus (et sur des erreurs relatives à la DB a priori) :

```
tox -e py3-django32-codestyle-coverage
```

```
py3-django32-codestyle-coverage: commands[2]> py.test -v --dist loadfile --junitxml=junit-py3-djan
go32-codestyle-coverage.xml --cov-report xml --cov-report html --cov=chrono/ --cov-config .coverag
erc --numprocesses=1 tests/
```

```
===== test session starts =====
```

```
platform linux -- Python 3.11.8, pytest-8.1.1, pluggy-1.4.0 -- /tmp/tox-yann/chrono/py3-django32-c
odestyle-coverage/bin/python
```

```
cachedir: /tmp/tox-yann/chrono/py3-django32-codestyle-coverage/.pytest_cache
```

```
django: version: 3.2.25, settings: chrono.settings (from env)
```

```
rootdir: /home/yann/src/chrono
```

```
plugins: cov-5.0.0, django-webtest-1.9.11, django-4.8.0, xdist-3.5.0, freezegun-0.4.2
```

```
created: 1/1 worker
```

```
1 worker [1355 items]
```

```
scheduling tests via LoadFileScheduling
```

```
[...]
```

```
===== short test summ
ary info =====
```

```
FAILED tests/test_misc.py::test_translate_holidays_exceptions - django.db.utils.IntegrityError: du
plicate key value violates unique constraint "agendas_agenda_slug_9271fa33_uniq"
```

```
FAILED tests/test_misc.py::test_migration_convert_week_days - django.db.utils.IntegrityError: dupl
icate key value violates unique constraint "agendas_agenda_slug_9271fa33_uniq"
```

```
FAILED tests/test_misc.py::test_migration_booking_check_data - django.db.utils.IntegrityError: dup
licate key value violates unique constraint "agendas_agenda_slug_9271fa33_uniq"
```

```
FAILED tests/test_api_utils.py::test_err_desc_translation - django.db.utils.ProgrammingError: colu
mn agendas_agenda.snapshot_id does not exist
```

```
FAILED tests/api/fillslot/test_meetings_with_lock_code.py::test_meetings_agenda - django.db.utils.
ProgrammingError: column "snapshot_id" of relation "agendas_agenda" does not exist
```

```
FAILED tests/api/fillslot/test_meetings_with_lock_code.py::test_meetings_agenda_expiration - djang
o.db.utils.ProgrammingError: column "snapshot_id" of relation "agendas_agenda" does not exist
```

```
FAILED tests/api/fillslot/test_meetings_with_lock_code.py::test_meetings_agenda_with_resource_excl
usion - django.db.utils.ProgrammingError: column "snapshot_id" of relation "agendas_agenda" does n
ot exist
```

```
FAILED tests/api/fillslot/test_meetings_with_lock_code.py::test_virtual_agenda_with_external_user_
id_exclusion - django.db.utils.ProgrammingError: column "snapshot_id" of relation "agendas_agenda"
does not exist
```

```
FAILED tests/api/test_statistics.py::test_statistics_list - django.db.utils.ProgrammingError: colu
mn agendas_agenda.snapshot_id does not exist
```

```
FAILED tests/api/test_statistics.py::test_statistics_bookings - django.db.utils.ProgrammingError:
column agendas_agenda.snapshot_id does not exist
```

```
FAILED tests/api/test_statistics.py::test_statistics_bookings_subfilters_list - django.db.utils.Pr
ogrammingError: column agendas_category.snapshot_id does not exist
```

```
FAILED tests/api/test_statistics.py::test_statistics_bookings_virtual - django.db.utils.Programmin
gError: column agendas_agenda.snapshot_id does not exist
```

```
FAILED tests/api/test_meetingtype.py::test_agendas_meetingtypes_api - django.db.utils.ProgrammingError: column agendas_agenda.snapshot_id does not exist
FAILED tests/api/test_meetingtype.py::test_agendas_meetingtype_api - django.db.utils.ProgrammingError: column agendas_agenda.snapshot_id does not exist
FAILED tests/api/test_meetingtype.py::test_virtual_agendas_meetingtypes_api - django.db.utils.ProgrammingError: column agendas_agenda.snapshot_id does not exist
FAILED tests/api/test_shared_custody.py::test_add_shared_custody_agenda_with_rules - django.db.utils.ProgrammingError: column agendas_unavailabilitycalendar.snapshot_id does not exist
FAILED tests/api/test_desk.py::test_agendas_desks_api - django.db.utils.ProgrammingError: column agendas_agenda.snapshot_id does not exist
FAILED tests/api/test_resource.py::test_agendas_resources_api - django.db.utils.ProgrammingError: column agendas_agenda.snapshot_id does not exist
FAILED tests/api/test_serializer.py::test_subscribed_with_dates - django.db.utils.ProgrammingError: column agendas_agenda.snapshot_id does not exist
FAILED tests/test_snapshot.py::test_clear_snapshot - django.db.utils.ProgrammingError: column "snapshot_id" of relation "agendas_agenda" does not exist
ERROR tests/test_misc.py::test_clean_time_period_exceptions - django.core.management.base.CommandError: Database chrono-test-gw0 couldn't be flushed. Possible reasons:
ERROR tests/test_misc.py::test_translate_holidays_exceptions - django.core.management.base.CommandError: Database chrono-test-gw0 couldn't be flushed. Possible reasons:
ERROR tests/test_misc.py::test_migration_convert_week_days - django.core.management.base.CommandError: Database chrono-test-gw0 couldn't be flushed. Possible reasons:
ERROR tests/test_misc.py::test_migration_booking_check_data - django.core.management.base.CommandError: Database chrono-test-gw0 couldn't be flushed. Possible reasons:
ERROR tests/ants_hub/test_commands.py::test_sync_ants_hub - django.db.utils.ProgrammingError: relation "ants_hub_city" does not exist
ERROR tests/ants_hub/test_models.py::test_export_to_push - django.db.utils.ProgrammingError: column "snapshot_id" of relation "agendas_agenda" does not exist
ERROR tests/test_metz.py::test_get_all_slots - django.db.utils.ProgrammingError: column agendas_agenda.snapshot_id does not exist
===== 20 failed, 1332 passed, 235 warnings,
7 errors in 174.19s (0:02:54) =====
```

NUMPROCESSES=16 tox -e py3-django32-codestyle-coverage

```
[...]
tests/manager/test_all.py::test_meeting_agenda_lease_display
[gw0] [100%] PASSED tests/manager/test_all.py::test_meeting_agenda_lease_display /tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-packages/_pytest/main.py:339: Pluggy TeardownRaisedWarning: A plugin raised an exception during an old-style hookwrapper teardown.
Plugin: _cov, Hook: pytest_runtestloop
DataError: Couldn't use data file '/home/yann/src/chrono/.coverage.debian.231919.XrvvDAax': no such table: file
For more information see https://pluggy.readthedocs.io/en/stable/api_reference.html#pluggy.PluggyTeardownRaisedWarning
  config.hook.pytest_runtestloop(session=session)

INTERNALERROR> Traceback (most recent call last):
INTERNALERROR>   File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-packages/coverage/sqlitedb.py", line 114, in _execute
INTERNALERROR>     return self.con.execute(sql, parameters) # type: ignore[arg-type]
INTERNALERROR>     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
INTERNALERROR> sqlite3.OperationalError: no such table: file
INTERNALERROR>
INTERNALERROR> During handling of the above exception, another exception occurred:
INTERNALERROR>
INTERNALERROR> Traceback (most recent call last):
INTERNALERROR>   File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-packages/coverage/sqlitedb.py", line 119, in _execute
INTERNALERROR>     return self.con.execute(sql, parameters) # type: ignore[arg-type]
INTERNALERROR>     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
INTERNALERROR> sqlite3.OperationalError: no such table: file
INTERNALERROR>
INTERNALERROR> The above exception was the direct cause of the following exception:
INTERNALERROR>
INTERNALERROR> Traceback (most recent call last):
```

```
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/_pytest/main.py", line 285, in wrap_session
INTERNALERROR>     session.exitstatus = doit(config, session) or 0
INTERNALERROR>
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/_pytest/main.py", line 339, in _main
INTERNALERROR>     config.hook.pytest_runtestloop(session=session)
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/pluggy/_hooks.py", line 501, in __call__
INTERNALERROR>     return self._hookexec(self.name, self._hookimpls.copy(), kwargs, firstresult)
INTERNALERROR>     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/pluggy/_manager.py", line 119, in _hookexec
INTERNALERROR>     return self._inner_hookexec(hook_name, methods, kwargs, firstresult)
INTERNALERROR>     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/pluggy/_callers.py", line 155, in _multicall
INTERNALERROR>     teardown[0].send(outcome)
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/pytest_cov/plugin.py", line 339, in pytest_runtestloop
INTERNALERROR>     self.cov_controller.finish()
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/pytest_cov/engine.py", line 46, in ensure_topdir_wrapper
INTERNALERROR>     return meth(self, *args, **kwargs)
INTERNALERROR>     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/pytest_cov/engine.py", line 352, in finish
INTERNALERROR>     self.cov.save()
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/coverage/control.py", line 784, in save
INTERNALERROR>     data = self.get_data()
INTERNALERROR>     ^^^^^^^^^^^^^^^^^
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/coverage/control.py", line 865, in get_data
INTERNALERROR>     self._post_save_work()
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/coverage/control.py", line 896, in _post_save_work
INTERNALERROR>     self._data.touch_files(paths, plugin_name)
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/coverage/sqldata.py", line 621, in touch_files
INTERNALERROR>     self._file_id(filename, add=True)
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/coverage/sqldata.py", line 417, in _file_id
INTERNALERROR>     self._file_map[filename] = con.execute_for_rowid(
INTERNALERROR>     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/coverage/sqlitedb.py", line 170, in execute_for_rowid
INTERNALERROR>     with self.execute(sql, parameters) as cur:
INTERNALERROR> File "/usr/lib/python3.11/contextlib.py", line 137, in __enter__
INTERNALERROR>     return next(self.gen)
INTERNALERROR>     ^^^^^^^^^^^^^^^^^
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/coverage/sqlitedb.py", line 149, in execute
INTERNALERROR>     cur = self._execute(sql, parameters)
INTERNALERROR>     ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
INTERNALERROR> File "/tmp/tox-yann/chrono/py3-django32-codestyle-coverage/lib/python3.11/site-pa
ckages/coverage/sqlitedb.py", line 137, in _execute
INTERNALERROR>     raise DataError(f"Couldn't use data file {self.filename!r}: {msg}") from exc
INTERNALERROR> coverage.exceptions.DataError: Couldn't use data file '/home/yann/src/chrono/.cover
age.debian.231919.XrvvDAax': no such table: file

===== 3 failed, 1352 passed, 235 warn
ings, 4 errors in 53.41s =====
```

## Historique

#1 - 03 avril 2024 16:43 - Thomas Noël

Pour info, tox ne se base pas sur devinst, il est sensé construire un environnement "propre". Parfois il faut le remettre à jour avec un bon vieux "tox -rv ...".

Ici tu es sur des erreurs sqlite3 qui dénotent un foirage complet de l'affaire (nos tests doivent tourner sur du psql, devinst aide à ce niveau en construisant les accès nécessaires je crois bien)

#### #2 - 03 avril 2024 16:44 - Benjamin Dauvergne

Vérifie si t'as pas un autre process arrêté avec une connection ouverte sur la base chrono-test-.

#### #3 - 03 avril 2024 16:45 - Benjamin Dauvergne

Thomas Noël a écrit :

Pour info, tox ne se base pas sur devinst, il est sensé construire un environnement "propre". Parfois il faut le remettre à jour avec un bon vieux "tox -rv ...".

Là je penche plutôt pour une base qui n'est pas recréé de zéro.

Ici tu es sur des erreurs sqlite3 qui dénotent un foirage complet de l'affaire (nos tests doivent tourner sur du psql, devinst aide à ce niveau en construisant les accès nécessaires je crois bien)

Les erreurs sqlite3 viennent de coverage, pas de rapport, fils unique.

#### #4 - 03 avril 2024 16:54 - Yann Weber

Thomas Noël a écrit :

Pour info, tox ne se base pas sur devinst, il est sensé construire un environnement "propre". Parfois il faut le remettre à jour avec un bon vieux "tox -rv ...".

Merci, c'est bien ce qui me semblait, mais en désespoir de cause j'ai essayé de relancer les scripts devinst ;)

J'ai tenté plusieurs fois de supprimer le dossier /tmp/tox-yann/ avec les venv de tox mais ça ne change rien (et par acquis de conscience j'ai lancé avec tox -rv mais idem...)

Ici tu es sur des erreurs sqlite3 qui dénotent un foirage complet de l'affaire (nos tests doivent tourner sur du psql, devinst aide à ce niveau en construisant les accès nécessaires je crois bien)

Ok ! J'avoue que ça m'a fait douté (j'avais aussi compris que les tests étaient censé tourner sur psql). Donc un foirage total sur 7/1352 (ou 27/1352) tests : c'est envisageable que les tests passent de psql à sqlite "au bout d'un moment" ? Vu que j'ai vérifié : en stoppant psql j'ai bien l'essentiel des tests en erreurs.

#### #5 - 03 avril 2024 16:55 - Yann Weber

Benjamin Dauvergne a écrit :

Vérifie si t'as pas un autre process arrêté avec une connection ouverte sur la base chrono-test-.

Ok, je fais ça, merci !

#### #6 - 03 avril 2024 16:57 - Benjamin Dauvergne

Yann Weber a écrit :

Ok, je fais ça, merci !

Et sinon si la base existe sans personne connecté dessus, dropdb chrono-test-.

#### #7 - 03 avril 2024 17:09 - Yann Weber

Benjamin Dauvergne a écrit :

Et sinon si la base existe sans personne connecté dessus, dropdb chrono-test-.

Alors : j'ai vérifié, il n'y avait aucune connexion ouverte (avec `SELECT * FROM pg_stat_activity WHERE state='active'`). J'ai reboot (même si j'ai un peu honte d'en arriver là je l'ai fais). J'ai re-vérifié, toujours pas de connexion ouverte. J'ai DROP la db chrono-test- qui existait effectivement encore. J'ai relancé les tests et ça m'a donné strictement le même résultat (3 failed, 1352 passed, 235 warnings, 4 errors in 52.67s avec NUMPROCESSES=16)

Une fois les tests ayant échoués j'ai revérifié : pas de connexion ouverte, la DB chrono-test- n'existe plus.

#### #8 - 03 avril 2024 17:17 - Thomas Noël

Benjamin Dauvergne a écrit :

Les erreurs sqlite3 viennent de coverage, pas de rapport, fils unique.

J'ai une soeur.

#### #9 - 03 avril 2024 17:44 - Yann Weber

Valentin a trouvé que la fixture transactional\_db pose problème sur les tests qu'il lance en local.

Et effectivement, si je commente les 4 tests qui utilisent cette fixture dans tests/test\_misc.py je n'ai plus un seul test en erreur, quelque-soit la valeur de NUMPROCESSES (même si je conserve la stacktrace lié a coverage quand NUMPROCESSES=16).

Sinon, quand je lance les tests uniquement sur le fichier tests/test\_misc.py j'ai ces erreurs :

```
===== short test summary info =====
FAILED tests/test_misc.py::test_translate_holidays_exceptions - django.db.utils.IntegrityError: duplicate key value violates unique constraint "agendas_agenda_slug_9271fa33_uniq"
FAILED tests/test_misc.py::test_migration_convert_week_days - django.db.utils.IntegrityError: duplicate key value violates unique constraint "agendas_agenda_slug_9271fa33_uniq"
FAILED tests/test_misc.py::test_migration_booking_check_data - django.db.utils.IntegrityError: duplicate key value violates unique constraint "agendas_agenda_slug_9271fa33_uniq"
ERROR tests/test_misc.py::test_clean_time_period_exceptions - django.core.management.base.CommandError: Database chrono-test- couldn't be flushed. Possible reasons:
ERROR tests/test_misc.py::test_translate_holidays_exceptions - django.core.management.base.CommandError: Database chrono-test- couldn't be flushed. Possible reasons:
ERROR tests/test_misc.py::test_migration_convert_week_days - django.core.management.base.CommandError: Database chrono-test- couldn't be flushed. Possible reasons:
ERROR tests/test_misc.py::test_migration_booking_check_data - django.core.management.base.CommandError: Database chrono-test- couldn't be flushed. Possible reasons:
===== 3 failed, 4 passed, 2 warnings, 4 errors in 12.67s =====
```

Je ne sais pas si c'est un hasard, mais c'est le même nombre de failure/errors qu'avec NUMPROCESSES=16

#### #10 - 03 avril 2024 17:48 - Thomas Noël

Juste pour dire que je n'avais pas vu le NUMPROCESSES, et quand je lance tox avec NUMPROCESSES=16 j'ai bien exactement le même soucis (mais je ne fais jamais ça, ça fait trop de bruit).

#### #11 - 03 avril 2024 18:18 - Benjamin Dauvergne

Plus haut je vois numprocesses=1, le souci c'est avec 1 ou 16 ? ou les deux ? et en faisant tox -e py3-django32 -- tests (donc dans xdist) le souci persiste ?

#### #12 - 03 avril 2024 19:09 - Yann Weber

Benjamin Dauvergne a écrit :

Plus haut je vois numprocesses=1, le souci c'est avec 1 ou 16 ? ou les deux ?

Les deux (avec un nombre d'erreur différents selon le cas)

et en faisant tox -e py3-django32 -- tests (donc dans xdist) le souci persiste ?

Le souci persiste, avec le même nombre d'erreur qu'avec NUMPROCESSES=16 :

tox -e py3-django32 -- tests

```
===== short test summary info =====
FAILED tests/test_misc.py::test_translate_holidays_exceptions - django.db.utils.IntegrityError: duplicate key value violates unique constraint "agendas_agenda_slug_9271fa33_uniq"
FAILED tests/test_misc.py::test_migration_convert_week_days - django.db.utils.IntegrityError: duplicate key value violates unique constraint "agendas_agenda_slug_9271fa33_uniq"
```

```
FAILED tests/test_misc.py::test_migration_booking_check_data - django.db.utils.IntegrityError: duplicate key value violates unique constraint "agendas_agenda_slug_9271fa33_uniq"
ERROR tests/test_misc.py::test_clean_time_period_exceptions - django.core.management.base.CommandError: Database chrono-test- couldn't be flushed. Possible reasons:
ERROR tests/test_misc.py::test_translate_holidays_exceptions - django.core.management.base.CommandError: Database chrono-test- couldn't be flushed. Possible reasons:
ERROR tests/test_misc.py::test_migration_convert_week_days - django.core.management.base.CommandError: Database chrono-test- couldn't be flushed. Possible reasons:
ERROR tests/test_misc.py::test_migration_booking_check_data - django.core.management.base.CommandError: Database chrono-test- couldn't be flushed. Possible reasons:
===== 3 failed, 1352 passed, 235 warnings, 4 errors in 119.05s (0:01:59) =====
```

### #13 - 03 avril 2024 23:29 - Benjamin Dauvergne

- Fichier `fix_migration_tests.patch` ajouté

- Statut changé de Nouveau à Solution proposée

- Patch proposed changé de Non à Oui

J'ai ajouté `pytest-random` pour exécuter les tests dans un ordre quelconque et ça montre que ce sont les tests de migration qui mettent le boxon, ils ne remettent pas en place la structure de la base en sortie, il y a une fixture qui fait ça correctement dans `tests/confest.py` d'`authentic`, il faut appeler `migrate` systématiquement après les tests de migration.

J'attache un patch minimal.

### #14 - 04 avril 2024 09:17 - Yann Weber

- Assigné à mis à Benjamin Dauvergne

Merci pour le patch ! Malheureusement ça ne semble pas résoudre totalement le problème, voici le résultat de `tox -e py3-django32-codestyle-coverage -- tests`

```
===== short test summary info =====
FAILED tests/test_misc.py::test_translate_holidays_exceptions - django.db.utils.IntegrityError: duplicate key value violates unique constraint "agendas_agenda_slug_9271fa33_uniq"
FAILED tests/test_misc.py::test_migration_convert_week_days - django.db.utils.IntegrityError: duplicate key value violates unique constraint "agendas_desk_agenda_id_slug_c02a2181_uniq"
FAILED tests/test_misc.py::test_migration_booking_check_data - django.db.utils.IntegrityError: duplicate key value violates unique constraint "agendas_desk_agenda_id_slug_c02a2181_uniq"
ERROR tests/test_misc.py::test_clean_time_period_exceptions - django.core.management.base.CommandError: Database chrono-test- couldn't be flushed. Possible reasons:
ERROR tests/test_misc.py::test_translate_holidays_exceptions - ExceptionGroup: errors while tearing down <Function test_translate_holidays_exceptions> (2 sub-exceptions)
ERROR tests/test_misc.py::test_migration_convert_week_days - ExceptionGroup: errors while tearing down <Function test_migration_convert_week_days> (2 sub-exceptions)
ERROR tests/test_misc.py::test_migration_booking_check_data - ExceptionGroup: errors while tearing down <Function test_migration_booking_check_data> (2 sub-exceptions)
===== 3 failed, 1352 passed, 235 warnings, 4 errors in 116.22s (0:01:56) =====
```

### #15 - 04 avril 2024 09:51 - Yann Weber

- Fichier `fix_migration_tests2.patch` ajouté

- Assigné à changé de Benjamin Dauvergne à Yann Weber

J'ai refait un patch sur ton patch qui semble résoudre le problème (de mon côté en tout cas).

Si j'ai bien compris : j'ai enlevé la fixture `transactional_db` (à priori c'est elle qui s'emmêlait les pinceaux lors des flush etc.), et j'ai enlevé le `yield None` à ta fixture `migrate_on_teardown` (j'imagine que du coup le `migrate` est lancé au `setup` du test ?)

Je vais en faire une PR voir si ça passe aussi sur Jenkins.

### #16 - 04 avril 2024 09:53 - Robot Gitea

- Statut changé de Solution proposée à En cours

Yann Weber (yweber) a ouvert une pull request sur Gitea concernant cette demande :

- URL : <https://git.entrouvert.org/entrouvert/chrono/pulls/241>
- Titre : WIP: tests: fix migrations tests (#89040)

- Modifications : <https://git.entrouvert.org/entrouvert/chrono/pulls/241/files>

#### #17 - 04 avril 2024 10:12 - Benjamin Dauvergne

Yann Weber a écrit :

J'ai refait un patch sur ton patch qui semble résoudre le problème (de mon côté en tout cas).

Si j'ai bien compris : j'ai enlevé la fixture transactional\_db (à priori c'est elle qui s'emmêlait les pinceaux lors des flush etc.), et j'ai enlevé le yield None à ta fixture migrate\_on\_teardown (j'imagine que du coup le migrate est lancé au setup du test ?)

Je vais en faire une PR voir si ça passe aussi sur Jenkins.

Ça ne va pas marcher tout le temps, le migrate doit être lancé après le test pour remettre le schéma de la base en ordre, pas avant (avant il est considéré comme toujours bon, sinon c'est un bug). Là si un autre test arrive après un test de migration (ce que j'arrive à faire en utilisant pytest-random) ça va bugger. Ce cas n'est pas un truc rare pour les gens bizarres qui utilisent pytest-random comme moi, ça peut arriver par exemple si on relance uniquement les tests en erreur (via pytest --last-failed) ou n'importe quoi qui change l'ordre des tests.

#### #18 - 04 avril 2024 10:28 - Yann Weber

Benjamin Dauvergne a écrit :

Ça ne va pas marcher tout le temps, le migrate doit être lancé après le test pour remettre le schéma de la base en ordre, pas avant (avant il est considéré comme toujours bon, sinon c'est un bug). Là si un autre test arrive après un test de migration (ce que j'arrive à faire en utilisant pytest-random) ça va bugger. Ce cas n'est pas un truc rare pour les gens bizarres qui utilisent pytest-random comme moi, ça peut arriver par exemple si on relance uniquement les tests en erreur (via pytest --last-failed) ou n'importe quoi qui change l'ordre des tests.

Oui, ça me semblait aussi très étrange cette histoire de devoir migrate au setup ! Au final, en enlevant la fixture transactional\_db (et aucune autre modif) les tests passent sur ma machine et sur jenkins :D (par contre 63 tests sont en erreurs sur jenkins avec pytest-random <https://jenkins.entrouvert.org/job/gitea/job/chrono/job/wip%252F89040-fix-tests-db-migrations/2/> j'imagine que ça mérite un autre ticket ?)

#### #19 - 04 avril 2024 10:32 - Robot Gitea

- Statut changé de En cours à Solution proposée

#### #20 - 04 avril 2024 10:46 - Robot Gitea

- Statut changé de Solution proposée à Solution validée

Benjamin Dauvergne (bdauvergne) a approuvé une pull request sur Gitea concernant cette demande :

- URL : <https://git.entrouvert.org/entrouvert/chrono/pulls/241>

#### #21 - 04 avril 2024 10:47 - Benjamin Dauvergne

Yann Weber a écrit :

Benjamin Dauvergne a écrit :

Ça ne va pas marcher tout le temps, le migrate doit être lancé après le test pour remettre le schéma de la base en ordre, pas avant (avant il est considéré comme toujours bon, sinon c'est un bug). Là si un autre test arrive après un test de migration (ce que j'arrive à faire en utilisant pytest-random) ça va bugger. Ce cas n'est pas un truc rare pour les gens bizarres qui utilisent pytest-random comme moi, ça peut arriver par exemple si on relance uniquement les tests en erreur (via pytest --last-failed) ou n'importe quoi qui change l'ordre des tests.

Oui, ça me semblait aussi très étrange cette histoire de devoir migrate au setup ! Au final, en enlevant la fixture transactional\_db (et aucune autre modif) les tests passent sur ma machine et sur jenkins :D (par contre 63 tests sont en erreurs sur jenkins avec pytest-random <https://jenkins.entrouvert.org/job/gitea/job/chrono/job/wip%252F89040-fix-tests-db-migrations/2/> j'imagine que ça mérite un autre ticket ?)

transactional\_db n'est effectivement pas toujours nécessaire ou plus toujours nécessaire, dans le passé certains changement au schéma ne passaient pas dans une transaction ou dans un savepoint (le test étant dans une transaction, les transaction entre chaque migration se retrouve dans un savepoint), tant mieux si tout marche sans maintenant.

random n'est pas utilisé dans ce build (il est juste chargé, mais sans le paramètre --random à pytest ça ne fait rien).

#### #22 - 04 avril 2024 10:56 - Yann Weber

Benjamin Dauvergne a écrit :

random n'est pas utilisé dans ce build (il est juste chargé, mais sans le paramètre --random à pytest ça ne fait rien).

Oui, j'ai compris ça au bout d'un moment ;) Et en fait, il y a des tests en erreurs quand les tests sont lancés dans un ordre aléatoire dans différents workers.

Soit quand on enlève --dist loadfile (c'était le cas avec les build en erreurs sur Jenkins), soit en utilisant pytest-random-order (qui permet de lancer les tests dans un ordre aléatoire avec plusieurs processus, contrairement à pytest-random apparemment)

### #23 - 04 avril 2024 10:57 - Robot Gitea

- Statut changé de Solution validée à Résolu (à déployer)

Yann Weber (yweber) a mergé une pull request sur Gitea concernant cette demande :

- URL : <https://git.entrouvert.org/entrouvert/chrono/pulls/241>
- Titre : tests: remove transactional\_db fixture for migrations tests (#89040)
- Modifications : <https://git.entrouvert.org/entrouvert/chrono/pulls/241/files>

### #24 - 04 avril 2024 16:18 - Transition automatique

- Statut changé de Résolu (à déployer) à Solution déployée

#### Fichiers

---

fix_migration_tests.patch	2,79 ko	03 avril 2024	Benjamin Dauvergne
fix_migration_tests2.patch	1,73 ko	04 avril 2024	Yann Weber