



› DevOpsDays Fortaleza 2025

# CI em projetos open-source

Testando e entregando 85+  
libs mensalmente

Emídio Neto

# Quem sou eu

Maintainer OpenTelemetry Docs PT-BR

Approver OpenTelemetry Python SDK

Organizador do Cloud Native RN



Emídio Neto

# Agenda



- 1 Por que testar múltiplas versões ?
- 2 Como o Tox ajuda (ou não) + GHA
- 3 Releases + Backport
- 4 Desafios e lições aprendidas
- 5 Q&A

**Por que testar múltiplas  
versões ?**

**A integração contínua (CI) é o processo automatizado de build, teste e validação de cada alteração de código para garantir que o projeto permaneça estável e consistente ao longo do tempo.**

# Por que testar múltiplas versões ?

1

Garantir  
compatibilidade

2


Preparação para o  
futuro e detecção  
de depreciações

3


Requisitos  
específicos a  
depender do  
ambiente/cliente

# 2 monorepos

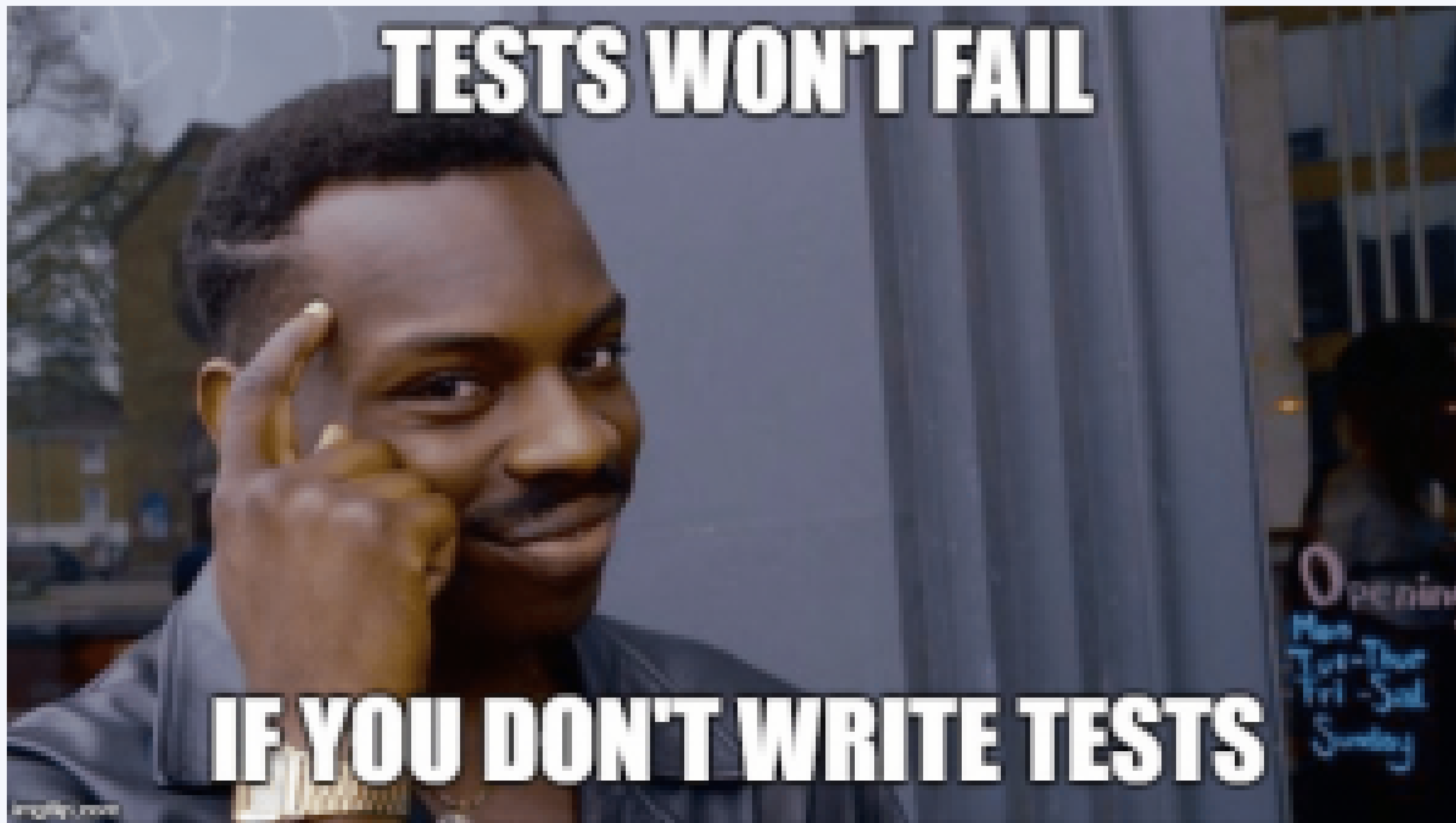
## Repo principal e repo de contrib (instrumentações específicas)

**opentelemetry-python-contrib** Public  
OpenTelemetry instrumentation for Python modules

Python 908 Apache-2.0 781 540 (107 issues need help) 193 Updated 2 days ago

**opentelemetry-python** Public  
OpenTelemetry Python API and SDK

Python 2,174 Apache-2.0 748 340 (31 issues need help) 70 Updated 3 days ago





“

## PEP 585 – Built-in Generic Types (Python 3.9)

A partir do Python 3.9, você pode usar os tipos integrados diretamente (por exemplo, `list[int]`), o que pode quebrar o código que depende do estilo antigo.

snippet.py

```
def double_numbers(nums: list[int]) -> list[int]:  
    return [n * 2 for n in nums]
```

snippet

```
$ python3.8 snippet.py
```

```
Traceback (most recent call last):
```

```
  File "snippet.py", line 1, in <module>
```

```
    def double_numbers(nums: list[int]) -> list[int]:
```

```
TypeError: 'type' object is not subscriptable
```

“

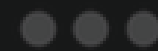
## PEP 604 – Allow writing union types as X | Y

No Python 3.9, embora você pudesse usar tipos integrados (como `list[int]`) graças ao PEP 585, os tipos union ainda precisam da sintaxe “`typing.Union`” se você quiser compatibilidade com versões anteriores (<3.10).



snippet

```
def get_items(data: list[int] | None) -> list[str] | None:
    if data is None:
        return None
    return [str(x) for x in data]
```



snippet

```
$ python3.9 snippet.py
Traceback (most recent call last):
  File "snippet.py", line 1, in <module>
    def get_items(data: list[int] | None) -> list[str] | None:
TypeError: unsupported operand type(s) for |: 'types.GenericAlias' and 'NoneType'
```

**Breaking changes que  
afetam o serviço em  
runtime não são legais**



Uma boa prática é  
rodar testes para todas  
versões Python que  
sua biblioteca suporta

Matriz de  
Testes

Static Type  
Checkers





python 3.9 | 3.10 | 3.11 | 3.12 | 3.13



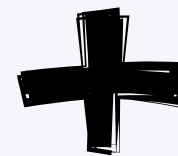
# Pacotes populares fazem isso





**opentelemetry-python** Public

18 Python Packages



**opentelemetry-python-contrib** Public

69 Python Packages



**opentelemetry-python** Public

~300+ Jobs na CI



**opentelemetry-python-contrib** Public

~700+ Jobs na CI



**All checks have passed**

1 skipped, 384 successful checks



Test 0 / opentelemetry-api 3.12 Ubu

+ 1000 Jobs

712 checks passed



instrumentation-openai-v2-latest 3.11 Ubuntu



instrumentation-urllib3-0 pypy-3.8 Ubuntu

# Testes

Considere a seguinte matriz de testes:

```
package: ["opentelemetry-api"]  
OS: ["Ubuntu"]  
python: ["3.8", "3.9", "3.10",  
"3.11", "3.12", "3.13"]
```

## Jobs

- ✓ opentelemetry-api 3.8 Ubuntu
- ✓ opentelemetry-api 3.9 Ubuntu
- ✓ opentelemetry-api 3.10 Ubuntu
- ✓ opentelemetry-api 3.11 Ubuntu
- ✓ opentelemetry-api 3.12 Ubuntu
- ✓ opentelemetry-api 3.13 Ubuntu



Para uma biblioteca parece OK  
mas e pra 80?



**All checks have passed**  
1 skipped, 711 successful checks

Test Name	Status	Duration	Required
Test 0 / instrumentation-falcon-4 3.13 Ubuntu (pull_request)	Successful	11s	
Test 0 / instrumentation-falcon-4 pypy-3.8 Ubuntu (pull_request)	Successful	30s	
Test 0 / instrumentation-fastapi 3.8 Ubuntu (pull_request)	Successful	18s	Required
Test 0 / instrumentation-fastapi 3.9 Ubuntu (pull_request)	Successful	15s	Required
Test 0 / instrumentation-fastapi 3.10 Ubuntu (pull_request)	Successful	13s	Required
Test 0 / instrumentation-fastapi 3.11 Ubuntu (pull_request)	Successful	13s	Required
Test 0 / instrumentation-fastapi 3.12 Ubuntu (pull_request)	Successful	12s	Required
Test 0 / instrumentation-fastapi 3.13 Ubuntu (pull_request)	Successful	12s	



**Como o Tox ajuda  
(ou não) + GHA**

“

Tox automatiza e  
padroniza testes em  
Python.  
Como um orquestrador

<https://tox.wiki>





**opentelemetry-python**

Public

18 Python Packages

X

6 versões Python  
+ pypy3 + lint

```
1  [tox]
2  isolated_build = True
3  skipsdist = True
4  skip_missing_interpreters = True
5  envlist =
6      ; Environments are organized by individual package, allowing
7      ; for specifying supported Python versions per package.
8
9      py3{8,9,10,11,12,13}-test-opentelemetry-api
10     pypy3-test-opentelemetry-api
11     lint-opentelemetry-api
12
13     py3{8,9,10,11,12,13}-test-opentelemetry-proto-protobuf5
14     pypy3-test-opentelemetry-proto-protobuf5
15     lint-opentelemetry-proto-protobuf5
16
17     py3{8,9,10,11,12,13}-test-opentelemetry-sdk
18     pypy3-test-opentelemetry-sdk
19     lint-opentelemetry-sdk
20     benchmark-opentelemetry-sdk
21
22     py3{8,9,10,11,12,13}-test-opentelemetry-semantic-conventions
23     pypy3-test-opentelemetry-semantic-conventions
24     lint-opentelemetry-semantic-conventions
```

```
[testenv]
deps =
    lint: -r dev-requirements.txt
    coverage: pytest
    coverage: pytest-cov

mypy,mypyinstalled: -r {toxiniidir}/mypy-requirements.txt

api: -r {toxiniidir}/opentelemetry-api/test-requirements.txt

sdk: -r {toxiniidir}/opentelemetry-sdk/test-requirements.txt
benchmark-opentelemetry-sdk: -r {toxiniidir}/opentelemetry-sdk/benchmark-requirements.txt

semantic-conventions: -r {toxiniidir}/opentelemetry-semantic-conventions/test-requirements.txt
```

Para cada pacote, instalar as dependências  
necessárias pro teste

```
commands =  
test-opentelemetry-api: pytest {toxinidir}/opentelemetry-api/tests {posargs}  
lint-opentelemetry-api: pylint {toxinidir}/opentelemetry-api  
  
test-opentelemetry-sdk: pytest {toxinidir}/opentelemetry-sdk/tests {posargs}  
lint-opentelemetry-sdk: pylint {toxinidir}/opentelemetry-sdk  
benchmark-opentelemetry-sdk: pytest {toxinidir}/opentelemetry-sdk/benchmarks --ben  
  
test-opentelemetry-proto-protobuf5: pytest {toxinidir}/opentelemetry-proto/tests {  
lint-opentelemetry-proto-protobuf5: pylint {toxinidir}/opentelemetry-proto  
  
test-opentelemetry-semantic-conventions: pytest {toxinidir}/opentelemetry-semantic  
lint-opentelemetry-semantic-conventions: pylint {toxinidir}/opentelemetry-semantic
```

Para cada pacote, rodar os testes, lint ou  
benchmark

# Só criar uma matriz dos 80 pacotes

...

# Até funcionou por um tempo, mas...

- **Job matrix** - A job matrix can generate a maximum of 256 jobs per workflow run. This limit applies to both GitHub-hosted and self-hosted runners.

Jinja2 Templates  
+ Automação em Python  
=  
Gerar os workflows

```
33     jobs:
34         {%- for job_data in job_datas %}
35
36         {{ job_data.name }}:
37             name: {{ job_data.ui_name }}
38             runs-on: {{ job_data.os }}
39             timeout-minutes: 30
40             steps:
41                 - name: Checkout repo @ SHA - ${% raw %}{{ github.sha }}{% endraw %}
42                   uses: actions/checkout@v4
43
44                 - name: Set up Python {{ job_data.python_version }}
45                   uses: actions/setup-python@v5
46                   with:
47                       python-version: "{{ job_data.python_version }}"
48
49                 - name: Install tox
50                   run: pip install tox-uv
51                 {%- if job_data.os == "windows-latest" %}
52
53                 - name: Configure git to support long filenames
54                   run: git config --system core.longpaths true
55                 {%- endif %}
56
57                 - name: Run tests
58                   run: tox -e {{ job_data.tox_env }} -- -ra
59         {%- endfor %}
```



```
35     py39-test-instrumentation-openai-v2-oldest_ubuntu-latest:
36         name: instrumentation-openai-v2-oldest 3.9 Ubuntu
37         runs-on: ubuntu-latest
38         timeout-minutes: 30
39         steps:
40             - name: Checkout repo @ SHA - ${ github.sha }
41               uses: actions/checkout@v4
42
43             - name: Set up Python 3.9
44               uses: actions/setup-python@v5
45               with:
46                 python-version: "3.9"
47
48             - name: Install tox
49               run: pip install tox-uv
50
51             - name: Run tests
52               run: tox -e py39-test-instrumentation-openai-v2-oldest -- -ra
53
```

```
203 generate-workflows:
204   name: generate-workflows
205   runs-on: ubuntu-latest
206   timeout-minutes: 30
207   if: |
208     !contains(github.event.pull_request.labels.*.name, 'Skip generate-workflows')
209     && github.event.pull_request.user.login != 'otelbot[bot]' && github.event_name == 'pull_request'
210   steps:
211     - name: Checkout repo @ SHA - ${github.sha}
212       uses: actions/checkout@v4
213
214     - name: Set up Python 3.11
215       uses: actions/setup-python@v5
216       with:
217         python-version: "3.11"
218
219     - name: Install tox
220       run: pip install tox-uv
221
222     - name: Run tests
223       run: tox -e generate-workflows
224
225     - name: Check github workflows are up to date
226       run: git diff --exit-code || (echo 'Generated workflows are out of date, run "tox -e generate-workflows" and commit the changes in this PR.' && exit 1)
227
```

También verifica se está tudo  
Up-to-date

**Releases + Backport**

## Opentelemetry Python → PEP 440

- Release: X.Y.Z → e.g., **1.37.0**
- Pre-releases: aN (alpha), bN (beta), rcN (release candidate) e.g., **0.58b0**
- Dev releases: .dev → e.g., **1.37.0.dev (fica apenas no git)**

e.g., opentelemetry-api → 1.34.0, 1.34.1 (patch release),

opentelemetry-instrumentation → 0.58b0, 0.58b1 (patch)

Automação via Github Actions cria uma branch de release por repositório: **e.g., release/v1.37.x-0.58bx**

2 PRs por repositório (automação via Github Actions)

- 1 para fazer bump da versão nos arquivos removendo o sufixo .dev para ser mergeado na branch de release (o que vai ser publicado no PyPi)
- 1 para fazer bump da versão mantendo o sufixo .dev para a branch main (manter o ciclo de desenvolvimento ativo)

# [release/v1.37.x-0.58bx] Prepare release 1.37.0/0.58b0 #4745

Merged

xrmx merged 2 commits into `release/v1.37.x-0.58bx` from `otelbot/prepare-release-1.37.0-0.58b0` last month

Conversation 0

Commits 2

Checks 369

Files changed 32



otelbot bot commented last month

Contributor ...

Prepare release 1.37.0/0.58b0.



opentelemetry-api/src/opentelemetry/version/\_\_init\_\_.py

@@ -12,4 +12,4 @@

12 # See the License for the specific language governing permissions and

13 # limitations under the License.

14

15 - \_\_version\_\_ = "1.37.0.dev"

12 # See the License for the specific language gov

13 # limitations under the License.

14

15 + \_\_version\_\_ = "1.37.0"

Release principal: Automação via GHA para realizar o build dos pacotes python e publicar no pypi a partir do que tiver na branch de release

# Update version to 1.38.0.dev/0.59b0.dev #4744

Merged

xrmx merged 3 commits into `main` from `otelbot/update-version-to-1.38.0.dev-0.59b0.dev` last month

Conversation 0

Commits 3

Checks 1,112

Files changed 31



otelbot bot commented last month

Contributor ...

Update version to 1.38.0.dev/0.59b0.dev .

opentelemetry-api/src/opentelemetry/version/\_\_init\_\_.py

@@ -12,4 +12,4 @@

12 # See the License for the specific language governing permissions and

13 # limitations under the License.

14

15 - \_\_version\_\_ = "1.37.0.dev"

12 # See the License for the specific

13 # limitations under the License.

14

15 + \_\_version\_\_ = "1.38.0.dev"

PR para atualizar a main com a versão de dev (próxima versão)

## release

succeeded last month in 2m 14s

- > ☒ Build wheels
- > ☒ Install twine
- > ☒ Publish to TestPyPI
- ▼ ☒ Publish to PyPI

```
1 ▶ Run twine upload --skip-existing --verbose dist/*
16 Uploading distributions to https://upload.pypi.org/legacy/
17 INFO dist/opentelemetry_api-1.37.0-py3-none-any.whl (64.2 KB)
18 INFO dist/opentelemetry_exporter_opencensus-0.58b0-py3-none-any.whl (11.1
19 KB)
20 INFO dist/opentelemetry_exporter_otlp-1.37.0-py3-none-any.whl (6.9 KB)
21 INFO dist/opentelemetry_exporter_otlp_proto_common-1.37.0-py3-none-any.whl
22 (17.9 KB)
23 INFO dist/opentelemetry_exporter_otlp_proto_grpc-1.37.0-py3-none-any.whl
24 (18.9 KB)
25 INFO dist/opentelemetry_exporter_otlp_proto_http-1.37.0-py3-none-any.whl
26 (19.1 KB)
27 INFO dist/opentelemetry_exporter_prometheus-0.58b0-py3-none-any.whl (12.7
28 KB)
29 INFO dist/opentelemetry_exporter_zipkin-1.37.0-py3-none-any.whl (6.8 KB)
30 INFO dist/opentelemetry_exporter_zipkin_json-1.37.0-py3-none-any.whl (16.0
31 KB)
32 INFO dist/opentelemetry_exporter_zipkin_proto_http-1.37.0-py3-none-any.whl
```

## release

succeeded last month in 2m 14s

- > ☒ Set up job
- > ☒ Run if [[ \$GITHUB\_REF\_NAME != release/\* ]]; then
- > ☒ Run actions/checkout@v4
- > ☒ Install toml
- > ☒ Set environment variables
- > ☒ Run if [[ -z \$PRIOR\_VERSION\_WHEN\_PATCH ]]; then
- > ☒ Run actions/checkout@v4
- > ☒ Run actions/checkout@v4
- > ☒ Run actions/setup-python@v5
- > ☒ Build wheels
- > ☒ Install twine
- > ☒ Publish to TestPyPI
- > ☒ Publish to PyPI
- > ☒ Generate release notes
- > ☒ Create GitHub release

# Release principal







## Fluxo de aplicar patches:

1. Abrir PR com o fix apontando pra main
2. Approve + Merge do PR com o fix, adicionar changelog etc
3. Fazer git cherry-pick do commit na branch de release e abrir um novo PR para branch de release (backport)
4. Approve + Merge do PR na branch de release

Para patch release: Trigar workflow de prepare Patch release via Github Actions + Merge o PR gerado + Trigar workflow de release (mesmo usado para release principal)

# [release/v1.31.x-0.52bx] api: revert catching BaseException in trace (#4494) #4495

 Merged xrmx merged 3 commits into `open-telemetry:release/v1.31.x-0.52bx` from `emdneto:cherrypick-4494` on Mar 20

 Conversation 0  Commits 3  Checks 383  Files changed 8



**emdneto** commented on Mar 19





Member ...

Clean cherry-pick of [#4494](#) to the [release/v1.31.x-0.52bx](#) branch.



## [release/v1.31.x-0.52bx] Prepare release 1.31.1/0.52b1 #4496

 Merged xrmx merged 1 commit into `release/v1.31.x-0.52bx` from `opentelemetrybot/prepare-release-1.31.1-0.52b1` on Mar 20

 Conversation 0  Commits 1  Checks 383  Files changed 31



**opentelemetrybot** commented on Mar 20

Member ...

Prepare release 1.31.1/0.52b1.



```
58
59     - name: Update version
60       run: .github/scripts/update-version-patch.sh $STABLE_VERSION $UNSTABLE_VERSION $STABLE_VERSION_PREV $UNSTABLE_VERSION_PREV
61
62     - name: Set up Python 3.9
63       uses: actions/setup-python@v5
64       with:
65         python-version: 3.9
66
67     - name: Install tox
68       run: pip install tox
69
70     - name: run tox
71       run: tox -e generate
72
73     - name: Update the change log with the approximate release date
74       run: |
75         date=$(date "+%Y-%m-%d")
76         sed -Ei "s/^## Unreleased$/## Version ${STABLE_VERSION}\/${UNSTABLE_VERSION} ($date)/" CHANGELOG.md
77
78     - name: Use CLA approved github bot
79       run: .github/scripts/use-cla-approved-github-bot.sh
80
81     - uses: actions/create-github-app-token@df432ceedc7162793a195dd1713ff69aefc7379e # v2.0.6
82       id: otelbot-token
83       with:
84         app-id: ${vars.OTELBOT_APP_ID}
85         private-key: ${secrets.OTELBOT_PRIVATE_KEY}
86
87     - name: Create pull request
88       id: create_pr
89       env:
90         # not using secrets.GITHUB_TOKEN since pull requests from that token do not run workflows
91         GITHUB_TOKEN: ${steps.otelbot-token.outputs.token}
92       run: |
93         message="Prepare release ${STABLE_VERSION}/${UNSTABLE_VERSION}"
94         branch="otelbot/prepare-release-${STABLE_VERSION}-${UNSTABLE_VERSION}"
95
96         git commit -a -m "$message"
97         git push origin HEAD:$branch
98         pr_url=$(gh pr create --title "[${GITHUB_REF_NAME}] $message" \
99           --body "$message." \
100           --head $branch \
101           --base $GITHUB_REF_NAME)
```

# Desafios e Lições Aprendidas

Total minutes


751,895

Total minutes across all workflows in this organization for last year

Total job runs

623,532

Total job runs across all workflows in this organization for last year

 **opentelemetry-python** Public

Total minutes

2,402,779

Total minutes across all workflows in this organization for last year

Total job runs

2,325,445

Total job runs across all workflows in this organization for last year

 **opentelemetry-python-contrib** Public

Avg job run time

30s

Average run time of jobs in this organization for last month

Avg job queue time

2s

Average queue time of jobs in this organization for last month

Job failure rate


3%

Failure rate across jobs in this organization for last month

Failed job usage

1,541

Total minutes used across failed jobs in this organization for last month

 **opentelemetry-python** Public

Avg job run time

21s

Average run time of jobs in this organization for last month

Avg job queue time

5s

Average queue time of jobs in this organization for last month

Job failure rate

1%

Failure rate across jobs in this organization for last month

Failed job usage

3,815

Total minutes used across failed jobs in this organization for last month

 **opentelemetry-python-contrib** Public

52.577h

1. Arquivos tox.ini imensos -- Chegando a ~1040 linhas
2. Difícil manter sincronizado o que está no tox.ini VS o que roda na CI
3. Instalação de pacotes lenta em alguns testes
4. CI de Lint demorava séculos para rodar
5. Testar um mesmo pacote em múltiplas versões do Python só que também para múltiplas versões de uma dependência específica

1. Matriz no Github Action tem limite de 250 Jobs -- sem chances
2. Testar o “básico” do repo -contrib em todo PR no repositório core
3. Testes no -contrib apontam para branch main no -core

```
[testenv]
test_deps =
    opentelemetry-api@{env:CORE_REPO}\#egg=opentelemetry-api&subdirectory=opentelemetry-api
    opentelemetry-semantic-conventions@{env:CORE_REPO}\#egg=opentelemetry-semantic-convention
    opentelemetry-sdk@{env:CORE_REPO}\#egg=opentelemetry-sdk&subdirectory=opentelemetry-sdk
    opentelemetry-test-utils@{env:CORE_REPO}\#egg=opentelemetry-test-utils&subdirectory=tests
```



# Obrigado!



@emdnetto

