Why did Flask start failing with "ImportError: cannot import name 'url_quote' from 'werkzeug.urls'"?

Asked 3 months ago Modified 24 days ago Viewed 81k times

125

Environment:

```
Python 3.10.11
Flask==2.2.2
```

I run my Flask backend code in docker container, with BASE Image: `FROM pytorch/pytorch:2.0.1-cuda11.7-cudnn8-runtime`

But when I run the pytest with version `pytest 7.4.2`,

```
pip install pytest
pytest
```

it raised an Error, with logs:

```
=================================== ERRORS
===================================
_____________ ERROR collecting tests/test_fiftyone_utils_utils.py
_____________
ImportError while importing test module '/builds/kw/data-auto-analysis-toolkit-
backend/tests/test_fiftyone_utils_utils.py'.
Hint: make sure your test modules/packages have valid Python names.
Traceback:
/opt/conda/lib/python3.10/importlib/__init__.py:126: in import_module
    return _bootstrap._gcd_import(name[level:], package, level)
tests/test_fiftyone_utils_utils.py:2: in <module>
    import daat  # noqa: F401
/opt/conda/lib/python3.10/site-packages/daat-1.0.0-py3.10.egg/daat/__init__.py:
1: in <module>
    from daat.app import app
/opt/conda/lib/python3.10/site-packages/daat-1.0.0-py3.10.egg/daat/app/
__init__.py:6: in <module>
    from flask import Flask, jsonify, request
/opt/conda/lib/python3.10/site-packages/flask/__init__.py:5: in <module>
    from .app import Flask as Flask
/opt/conda/lib/python3.10/site-packages/flask/app.py:30: in <module>
    from werkzeug.urls import url_quote
E   ImportError: cannot import name 'url_quote' from 'werkzeug.urls' (/opt/
conda/lib/python3.10/site-packages/werkzeug/urls.py)
```

My codes works well when I directly run it with `python run.py`

`run.py` shown below

```
from daat import app

app.run(host='0.0.0.0')
```

I guess it should be the pytest versions issue, because it used to work well without changing any related code, and I use `pip install pytest` without defined a specific version.

And my backend runs well without pytest.

python flask pytest werkzeug

Share Follow

edited Nov 16, 2023 at 22:35

Peter Mortensen
30.8k 22 106 131

asked Oct 2, 2023 at 3:02

stevezkw
1,771 2 6 13

Depreciated. Use urllib.parse.urlsplit instead. – MethodMan Nov 9, 2023 at 6:06

1 I edited the title because it should be immediately clear that this is about a Flask installation problem, and not for people who are trying to use the Werkzeug API directly. – Karl Knechtel Nov 16, 2023 at 20:44

6 Answers

Sorted by: Highest score (default)

194

I had the same problem. It is because `Werkzeug 3.0.0` was released and Flask doesn't specify the dependency correctly (requirements says `Werkzeug>=2.2.0`). This is why, `Werkzeug 3.0.0` is still installed and `Flask 2.2.2` isn't made for `Werkzeug 3.0.0`.

Solution: Just set a fix version for Werkzeug such as `Werkzeug==2.2.2` in your `requirements.txt` and it should work.

Share Follow

edited Oct 17, 2023 at 9:57


XM01 - stands with Palestine
6,823 5 16 41

answered Oct 2, 2023 at 8:17

Scrashdemix
2,058 1 7 5

Alternatively, you could use Werkzeug==2.3.x as this was the last version used according to the version specified by flask. – Scrashdemix Oct 2, 2023 at 8:56

1 This is the error I got: `The conflict is caused by: The user requested Werkzeug==2.2; flask 2.2.2 depends on Werkzeug>=2.2.2` – IanS Oct 2, 2023 at 8:59

14 When using `flask==2.1.3`, it seems forcing `Werkzeug==2.3.7` solves the issue for us. According to our logs, that's the version of `Werkzeug` that still got resolved by `pip` as of end of last week. – Svend Oct 2, 2023 at 9:58

1 `Werkzeug>=2.2,<3.0` should also work – Tails86 Oct 2, 2023 at 14:12

10 Can also confirm that using `flask==2.2.2` and adding `Werkzeug==2.3.7` fixed this issue for us – dem Oct 3, 2023 at 16:58

14

The root cause of this is that `Werkzeug 3.0.0` removed previously deprecated code: https://werkzeug.palletsprojects.com/en/3.0.x/changes/#version-3-0-0

Please update your Flask version, `Flask 2.2.2` is unsupported: https://github.com/pallets/flask/releases

Anyway, you need to pin Werkzeug yourself then if you insist on using a deprecated version of `Flask`, or if your code is using `url_quote` directly then you can switch to the built-in `urllib`:

```
from urllib.parse import quote as url_quote
```

Share Follow

edited Dec 16, 2023 at 7:24

answered Oct 3, 2023 at 11:58

aude
1,562 17 21

This solution worked for us. The aforementioned one (pinning Werkzeug version) functions as well, but this one is IMHO the correct way. – jvleminc Oct 4, 2023 at 12:50

10

I started getting this error in an update I deployed today, even though I wasn't trying to import "url_quote". `Flask == 2.0.1`. Setting `Werkzeug==2.2.2` also worked for me.

Share Follow

edited Oct 17, 2023 at 9:58

XM01 - stands with Palestine
6,823 5 16 41

answered Oct 2, 2023 at 19:15

Eye Scream
101 4

5

Modify your requirements.txt to include:

`Werkzeug==2.2.x` or `Werkzeug==2.3.x`. Or use `Werkzeug==2.2.2` to be safe.

Share Follow

answered Oct 20, 2023 at 2:56

Liam Swayne
51 2

Seems pretty much the same as many answers already posted. – ggorlen Nov 12, 2023 at 4:34

4

```
connexion[swagger-ui]<3
flask>=2.0
Werkzeug>=2.0
gunicorn>=20.0
```

This combination worked for me and resolved into this:

```
Successfully installed Werkzeug-2.2.3 connexion-2.14.2 flask-2.2.5
```

Python 3.11/3.12

Share Follow

answered Nov 4, 2023 at 17:01

Henadzi Rabkin

6,923 3 33 39

0

If you're encountering an issue with the "url_quote" function in your `Flask` application, it's likely due to an incorrect import or a version conflict between `Flask` and `Werkzeug`.

To resolve this issue, follow the steps below:

- **Update Your Flask Version**

This step ensures that you have the latest Flask version :

```
pip install --upgrade Flask
```

- **Update pytest**

In some cases, the problem could be related to a pytest version conflict. You can try upgrading pytest to a version that is compatible with your environment using the following command :

```
pip install --upgrade pytest
```

- **Downgrade the Werkzeug Version**

If updating `Flask` and resolving package conflicts doesn't solve the problemconsider using `Werkzeug==2.3.x`, be aware of dependency constraints, and force `Werkzeug==2.3.7` with `Flask==2.1.3` if needed. Specifying a Werkzeug version range like `Werkzeug>=2.2,<3.0` is an adaptable option. Test with `Flask==2.2.2` and `Werkzeug==2.3.7` and verify `Flask` version compatibility. You can also specify the `Werkzeug` version in your `requirements.txt` file, e.g., `Werkzeug==2.3.6`. These steps should help manage version conflicts in your `Flask` application.

Share Follow

edited Oct 17, 2023 at 20:43

answered Oct 17, 2023 at 9:52

XM01 - stands with Palestine

6,823 5 16 41

Highly active question. Earn 10 reputation (not counting the association bonus) in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.