



Quick search

Go

Index

[Alphabetical](#)

[Tree](#)

Neo4j Python Driver 5.21

The Official Neo4j Driver for Python.

Neo4j versions supported:

- Neo4j 5.0 - 5.21
- Neo4j 4.4

Python versions supported:

- Python 3.12 (added in driver version 5.14.0)
- Python 3.11 (added in driver version 5.3.0)
- Python 3.10
- Python 3.9
- Python 3.8
- Python 3.7

Topics

This website uses cookies

We use cookies to offer you a better browsing experience, analyze site traffic, personalize content and serve targeted ads. Learn about how we use cookies and how you can control them in [Cookie Settings](#). By using our site, you consent to our use of cookies.

[Accept Cookies](#)

[Use necessary cookies only](#)

Installation

To install the latest stable release, use:

```
python -m pip install neo4j
```

To install the latest pre-release, use:

```
python -m pip install --pre neo4j
```

Note: `neo4j-driver` is the old name for this package. It is now deprecated and will receive no further updates starting with 6.0.0. Make sure to install `neo4j` as shown above.

Note: It is always recommended to install python packages for user space in a virtual environment.

Virtual Environment

To create a virtual environment named `sandbox`, use:

```
python -m venv sandbox
```

To activate the virtual environment named `sandbox`, use:

```
source sandbox/bin/activate
```

To deactivate the current active virtual environment, use:

```
deactivate
```

This website uses cookies

We use cookies to offer you a better browsing experience, analyze site traffic, personalize content and serve targeted ads. Learn about how we use cookies and how you can control them in [Cookie Settings](#). By using our site, you consent to our use of cookies.

[Accept Cookies](#)

[Use necessary cookies only](#)

- enable **DeprecationWarning** , which the driver emits if deprecated APIs are used.
- enable the driver's debug mode (this can also be achieved by setting the environment variable `PYTHONNEO4JDEBUG`):
 - **This is experimental.** It might be changed or removed any time even without prior notice.
 - the driver will raise an exception if non-concurrency-safe methods are used concurrently.
 - the driver will emit warnings if the server sends back notification (see also [warn_notification_severity](#)).

Added in version 5.15.

Changed in version 5.21: Added functionality to automatically emit warnings on server notifications.

Quick Example

```
from neo4j import GraphDatabase, RoutingControl

URI = "neo4j://localhost:7687"
AUTH = ("neo4j", "password")

def add_friend(driver, name, friend_name):
    driver.execute_query(
        "MERGE (a:Person {name: $name}) "
        "MERGE (friend:Person {name: $friend_name}) "
        "MERGE (a)-[:KNOWS]->(friend)",
        name=name, friend_name=friend_name, database_="neo4j",
    )

def print_friends(driver, name):
    records, _, _ = driver.execute_query(
        "MATCH (a:Person)-[:KNOWS]->(friend) WHERE a.name = $name "
        "RETURN friend.name ORDER BY friend.name",
        name=name, database_="neo4j", routing_=RoutingControl.READ,
    )
```

This website uses cookies

We use cookies to offer you a better browsing experience, analyze site traffic, personalize content and serve targeted ads. Learn about how we use cookies and how you can control them in [**Cookie Settings**](#). By using our site, you consent to our use of cookies.

[Accept Cookies](#)

[Use necessary cookies only](#)

Example Application

```
import logging

from neo4j import GraphDatabase, RoutingControl
from neo4j.exceptions import DriverError, Neo4jError

class App:

    def __init__(self, uri, user, password, database=None):
        self.driver = GraphDatabase.driver(uri, auth=(user, password))
        self.database = database

    def close(self):
        # Don't forget to close the driver connection when you are finished
        # with it
        self.driver.close()

    def create_friendship(self, person1_name, person2_name):
        with self.driver.session() as session:
            # Write transactions allow the driver to handle retries and
            # transient errors
            result = self._create_and_return_friendship(
                person1_name, person2_name
            )
            print("Created friendship between: "
                  f"{result['p1']}, {result['p2']}")

    def _create_and_return_friendship(self, person1_name, person2_name):

        # To learn more about the Cypher syntax,
        # see https://neo4j.com/docs/cypher-manual/current/

        # The Cheat Sheet is also a good resource for keywords,
        # see https://neo4j.com/docs/cypher-cheat-sheet/

        query = (
            "CREATE (p1:Person { name: $person1_name }) "
            "CREATE (p2:Person { name: $person2_name }) "
            "CREATE (p1)-[:KNOWS]->(p2) "
            "RETURN p1.name, p2.name"
        )
        try:
            record = self.driver.execute_query(
                query, person1_name=person1_name, person2_name=person2_name,
```

This website uses cookies

We use cookies to offer you a better browsing experience, analyze site traffic, personalize content and serve targeted ads. Learn about how we use cookies and how you can control them in [Cookie Settings](#). By using our site, you consent to our use of cookies.

[Accept Cookies](#)

[Use necessary cookies only](#)

```

def _find_and_return_person(self, person_name):
    query = (
        "MATCH (p:Person) "
        "WHERE p.name = $person_name "
        "RETURN p.name AS name"
    )
    names = self.driver.execute_query(
        query, person_name=person_name,
        database_=self.database, routing_=RoutingControl.READ,
        result_transformer_=lambda r: r.value("name")
    )
    return names

if __name__ == "__main__":
    # For Aura specific connection URI,
    # see https://neo4j.com/developer/aura-connect-driver/ .
    scheme = "neo4j" # Connecting to Aura, use the "neo4j+s" URI scheme
    host_name = "example.com"
    port = 7687
    uri = f"{scheme}://{host_name}:{port}"
    user = "<Username for Neo4j database>"
    password = "<Password for Neo4j database>"
    database = "neo4j"
    app = App(uri, user, password, database)
    try:
        app.create_friendship("Alice", "David")
        app.find_person("Alice")
    finally:
        app.close()

```

Further Information

- [The Neo4j Operations Manual](#) (docs on how to run a Neo4j server)
- [The Neo4j Python Driver Manual](#) (good introduction to this driver)
- [Python Driver API Documentation](#) (full API documentation for this driver)
- [Neo4j Cypher Cheat Sheet](#) (summary of Cypher syntax - Neo4j's graph query language)
- [Example Project](#) (small web application using this driver)
- [GraphAcademy](#) (interactive, free online trainings for Neo4j)
- [Driver Wiki](#) (includes change logs)

This website uses cookies

We use cookies to offer you a better browsing experience, analyze site traffic, personalize content and serve targeted ads. Learn about how we use cookies and how you can control them in [Cookie Settings](#). By using our site, you consent to our use of cookies.

[Accept Cookies](#)

[Use necessary cookies only](#)





Neo4j[®], Neo Technology[®], Cypher[®], Neo4j[®]
Bloom[™] and Neo4j[®] Aura[™] are registered
trademarks of Neo4j, Inc. All other marks are
owned by their respective companies.

US: 1-855-636-4532
Sweden +46 171 480 113
UK: +44 20 3868 3223
France: +33 (0) 1 88 46 13 20

Learn

-  [Sandbox](#)
-  [Neo4j Community Site](#)
-  [Neo4j Developer Blog](#)
-  [Neo4j Videos](#)
-  [GraphAcademy](#)
-  [Neo4j Labs](#)

Social

-  [Twitter](#)
-  [Meetups](#)
-  [Github](#)
-  [Stack Overflow](#)

[Want to Speak?](#)

This website uses cookies

We use cookies to offer you a better browsing experience, analyze site traffic, personalize content and serve targeted ads. Learn about how we use cookies and how you can control them in [Cookie Settings](#). By using our site, you consent to our use of cookies.

[Accept Cookies](#)

[Use necessary cookies only](#)