

[Презентация к уроку 6.3.3](#)

## Код

### Прикладная работа с Neo4j

#### Установка локальной базы в Docker

```
sudo docker pull neo4j:5.3.0
sudo docker run -p7687:7687 --name myneo4j --env NEO4J_AUTH=neo4j/testttest neo4j:5.3.0
sudo docker exec -it myneo4j bash
```

Заходим в контейнер и работаем с БД.

#### Использование драйвера Python Neo4j

```
pip install neo4j
if __name__ == "__main__":
    greeter = HelloWorldExample("bolt://localhost:7687", "neo4j", "password")
    greeter.print_greeting("hello, world")
    greeter.close()
```

Этот код использует подключение к БД Neo4j чтобы создать объект-подключение, напечатать приветствие и закрыть подключение.

Посмотрим как создан класс объекта-приветствие:



```

from neo4j import GraphDatabase

class HelloWorldExample:

    def __init__(self, uri, user, password):
        self.driver = GraphDatabase.driver(uri, auth=(user, password))

    def close(self):
        if self.driver is not None:
            self.driver.close()

    def print_greeting(self, message):
        with self.driver.session() as session:
            greeting = session.execute_write(self._create_and_return_greeting, message)
            print(greeting)

    @staticmethod
    def _create_and_return_greeting(tx, message):
        result = tx.run("CREATE (a:Greeting) "
                        "SET a.message = $message "
                        "RETURN a.message + ', from node ' + id(a)", message=message)
        return result.single()[0]

```

Текстовая расшифровка видео:

## ОБЗОР NEO4J

**План:**

- Neo4j.

### Neo4j

[Neo4j](#) позволяет работать с графовыми моделями данных, так как если бы мы хранили такие данные в реляционных базах, то задача бы усложнилась, потому что связи между данными тоже имеют свои свойства. Neo4j делает работу с ними тривиальнее.

Работаем с Neo4j. Для этого устанавливаем его в контейнере, скачиваем образ, pull, название образа. Как только образ скачается – перезапустим:

```

asya@asya-HP-Pavilion-dv6-Notebook-PC:~$ sudo docker pull neo4j:5.3.0
[sudo] password for asya:
5.3.0: Pulling from library/neo4j
8740c948ffd4: Pull complete
f12eef77f798: Pull complete
dacf190957de: Pull complete
f1c30508fa3b: Pull complete
6bf95ba3a6e2: Pull complete
Digest: sha256:84e9e5a1d82cee0253d512a8a4b449f0d26eaa82545b509d5e33fc9fd0e0a979
Status: Downloaded newer image for neo4j:5.3.0
docker.io/library/neo4j:5.3.0
docker.io/library/neo4j:5.3.0
asya@asya-HP-Pavilion-dv6-Notebook-PC:~$ sudo docker run -p7687:7687 --name myneo4j --env NEO4J_AUTH=neo4j/testte
st neo4j:5.3.0

```

Мы видим следующее:

- Порт по умолчанию – 7687;
- Имя контейнера – «myneo4j»;
- Переменные среды;
- Логин и пароль.

Открываем еще одно окошко терминала, проверяем работает ли контейнер и можно ли в него зайти:

```
asya@asya-HP-Pavilion-dv6-Notebook-PC:~$ sudo docker ps
[sudo] password for asya:
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
961b19492ea4  neo4j:5.3.0   "tini -g -- /startup..." 35 seconds ago Up 32 seconds 7473-
7->7687/tcp, :::7687->7687/tcp myneo4j
asya@asya-HP-Pavilion-dv6-Notebook-PC:~$
```

Далее, указываем ID контейнера, также можно указать его название. Мы зашли в контейнер, в нем лежат файлы:

```
asya@asya-HP-Pavilion-dv6-Notebook-PC:~$ sudo docker exec -it myneo4j bash
root@961b19492ea4:/var/lib/neo4j# ls
LICENSE.txt  NOTICE.txt  UPGRADE.txt  certificates  data  labs  licenses  packaging_info  run
LICENSES.txt  README.txt  bin          conf          import  lib  logs  plugins
root@961b19492ea4:/var/lib/neo4j#
```

Для того, чтобы работать с Neo4j, используем утилиту «**Cypher-shell**». Под юзером Neo4j запаролен testtest (это позволит писать на языке Cypher). Мы получили доступ к оболочке Cypher (имя и пароль те же, что мы указываем при создании контейнера). По умолчанию и логин, и пароль – «Neo4j»:

```
Connected to Neo4j using Bolt protocol version 5.0 at neo4j://localhost:7687 as user neo4j.
Type :help for a list of available commands or :exit to exit the shell.
Note that Cypher queries must end with a semicolon.
neo4j@neo4j>
```

Напишем запросы на языке Cypher.

Для начала посмотрим, какие есть узлы:

```
root@961b19492ea4:/var/lib/neo4j# cypher-shell -u neo4j -p testtest
Connected to Neo4j using Bolt protocol version 5.0 at neo4j://localhost:7687 as user neo4j.
Type :help for a list of available commands or :exit to exit the shell.
Note that Cypher queries must end with a semicolon.
neo4j@neo4j> MATCH (n) RETURN count(n);
+-----+
| count(n) |
+-----+
| 0         |
+-----+

1 row
ready to start consuming query after 690 ms, results consumed after another 11 ms
neo4j@neo4j>
```

Видим:

- Match (n);
- Return count (n).

Это логично, так как в начале никаких узлов нет.

Для того, чтобы выйти, используем команду «:exit». Мы вышли из cypher-shell, но все еще находимся в контейнере:

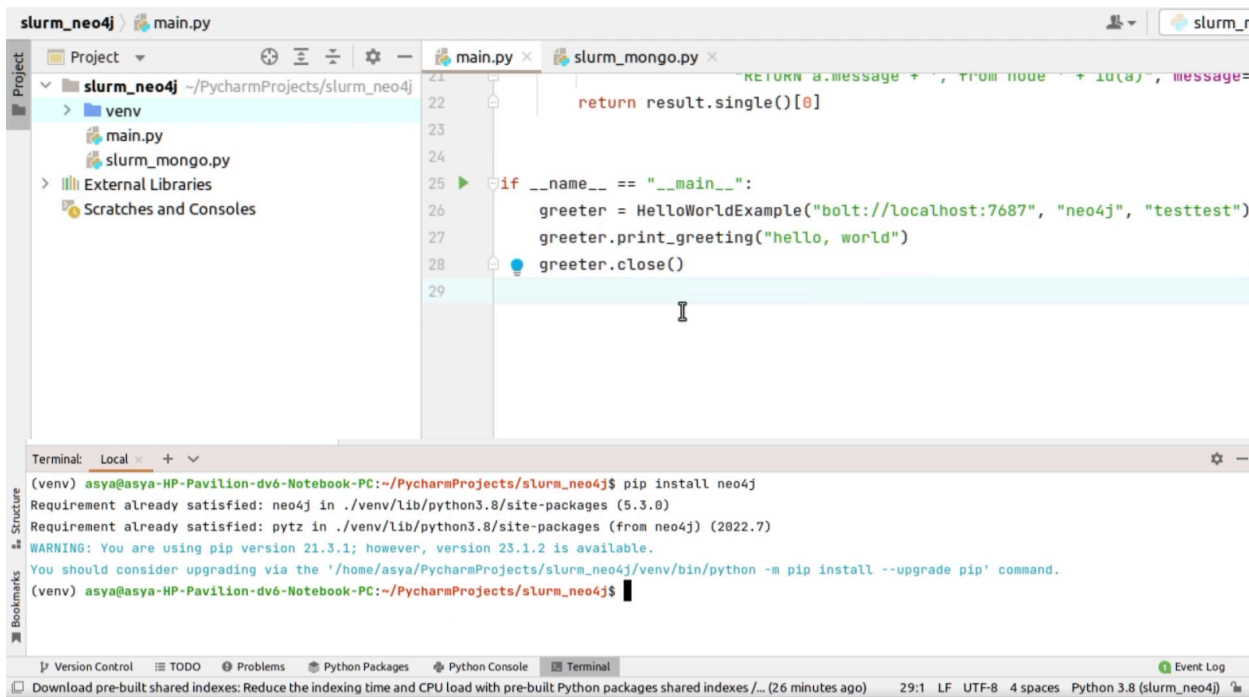
```
[sudo] password for asya:
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
961b19492ea4  neo4j:5.3.0   "tini -g -- /startup..." 35 seconds ago Up 32 seconds 7473-
7->7687/tcp, :::7687->7687/tcp myneo4j
asya@asya-HP-Pavilion-dv6-Notebook-PC:~$ sudo docker exec -it myneo4j bash
root@961b19492ea4:/var/lib/neo4j# ls
LICENSE.txt  NOTICE.txt  UPGRADE.txt  certificates  data  labs  licenses  packaging_info  run
LICENSES.txt  README.txt  bin          conf          import  lib  logs  plugins
root@961b19492ea4:/var/lib/neo4j# cypher-shell -u neo4j -p testtest
Connected to Neo4j using Bolt protocol version 5.0 at neo4j://localhost:7687 as user neo4j.
Type :help for a list of available commands or :exit to exit the shell.
Note that Cypher queries must end with a semicolon.
neo4j@neo4j> MATCH (n) RETURN count(n);
+-----+
| count(n) |
+-----+
| 0         |
+-----+

1 row
ready to start consuming query after 690 ms, results consumed after another 11 ms
neo4j@neo4j> :exit
Bye!
root@961b19492ea4:/var/lib/neo4j#
```

По аналогии БД в контейнере напоминают базу от облачного поставщика.

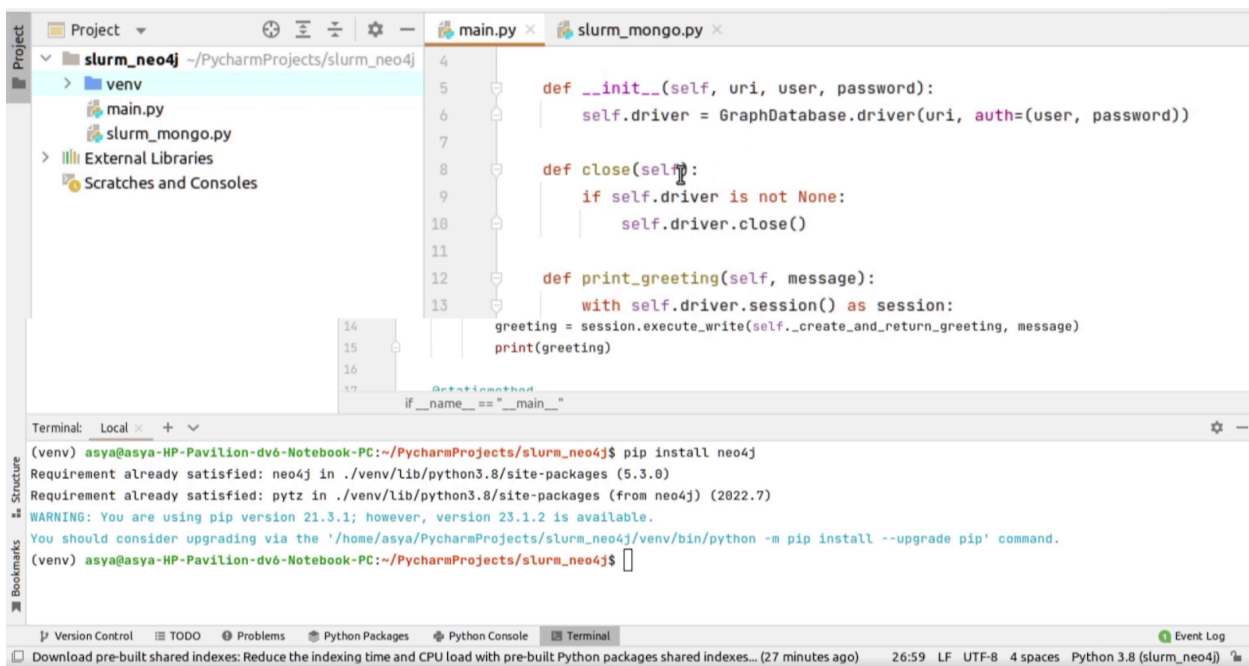
**Далее, поработаем через драйвер Python.**

Для работы нам необходимо установить библиотеку «Neo4j», после чего посмотрим в код:



```
slurm_neo4j main.py
Project
  slurm_neo4j ~/PycharmProjects/slurm_neo4j
  venv
  main.py
  slurm_mongo.py
  External Libraries
  Scratches and Consoles
Terminal
Local
(venv) asya@asya-HP-Pavilion-dv6-Notebook-PC:~/PycharmProjects/slurm_neo4j$ pip install neo4j
Requirement already satisfied: neo4j in ./venv/lib/python3.8/site-packages (5.3.0)
Requirement already satisfied: pytz in ./venv/lib/python3.8/site-packages (from neo4j) (2022.7)
WARNING: You are using pip version 21.3.1; however, version 23.1.2 is available.
You should consider upgrading via the '/home/asya/PycharmProjects/slurm_neo4j/venv/bin/python -m pip install --upgrade pip' command.
(venv) asya@asya-HP-Pavilion-dv6-Notebook-PC:~/PycharmProjects/slurm_neo4j$
```

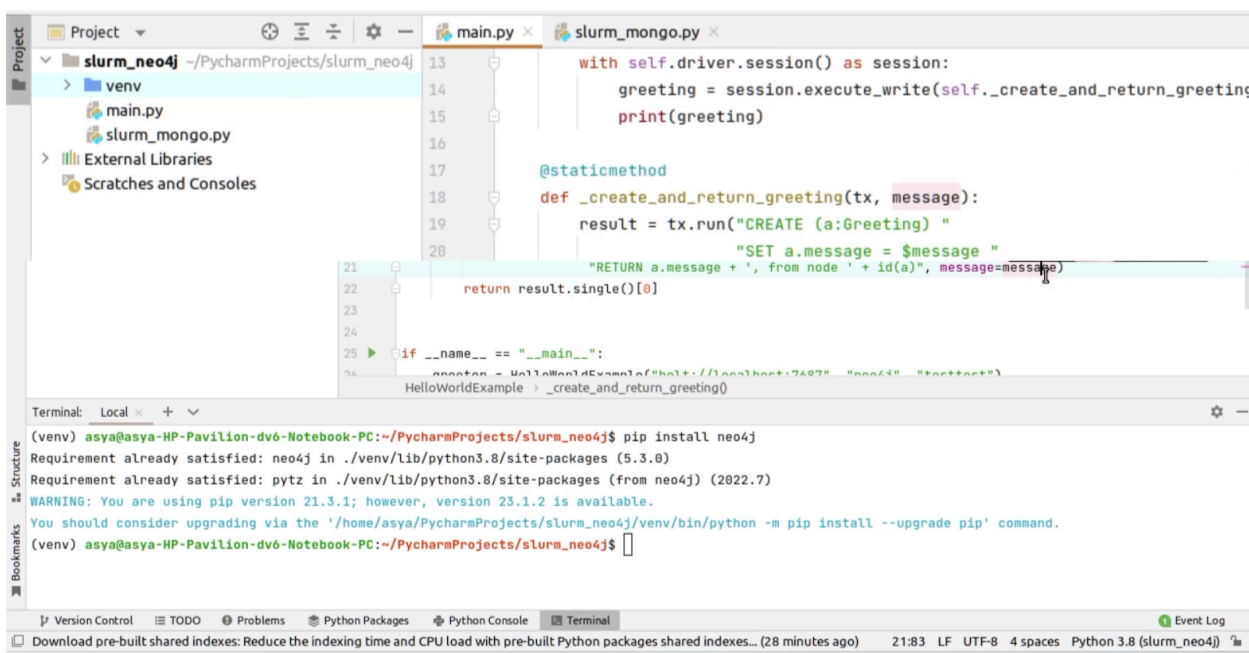
В «if \_\_name\_\_ == "\_\_main\_\_"» пишем greeter, который отправляет «HelloWord» и закрывает соединение. Вначале подключение создается, отправляется «HelloWord», после чего подключение закрывается:



```
Project
  slurm_neo4j ~/PycharmProjects/slurm_neo4j
  venv
  main.py
  slurm_mongo.py
  External Libraries
  Scratches and Consoles
Terminal
Local
(venv) asya@asya-HP-Pavilion-dv6-Notebook-PC:~/PycharmProjects/slurm_neo4j$ pip install neo4j
Requirement already satisfied: neo4j in ./venv/lib/python3.8/site-packages (5.3.0)
Requirement already satisfied: pytz in ./venv/lib/python3.8/site-packages (from neo4j) (2022.7)
WARNING: You are using pip version 21.3.1; however, version 23.1.2 is available.
You should consider upgrading via the '/home/asya/PycharmProjects/slurm_neo4j/venv/bin/python -m pip install --upgrade pip' command.
(venv) asya@asya-HP-Pavilion-dv6-Notebook-PC:~/PycharmProjects/slurm_neo4j$
```

В данном случае описан класс, инициализируется объект подключения, он импортируется из Neo4j в GraphDatabase.driver, передается user, password и метод «Close». Если существует открытое соединение, то его нужно закрыть. Этот метод тоже наследуется, и мы печатаем приветствие. Если есть подключение, то оно создается на основе message и печатает это приветствие.

Есть статический метод, который не использует self. В этой зоне мы отправляем запрос на сайфер, где создаем приветствие:



```
Project
  slurm_neo4j ~/PycharmProjects/slurm_neo4j
  venv
  main.py
  slurm_mongo.py
  External Libraries
  Scratches and Consoles
Terminal
Local
(venv) asya@asya-HP-Pavilion-dv6-Notebook-PC:~/PycharmProjects/slurm_neo4j$ pip install neo4j
Requirement already satisfied: neo4j in ./venv/lib/python3.8/site-packages (5.3.0)
Requirement already satisfied: pytz in ./venv/lib/python3.8/site-packages (from neo4j) (2022.7)
WARNING: You are using pip version 21.3.1; however, version 23.1.2 is available.
You should consider upgrading via the '/home/asya/PycharmProjects/slurm_neo4j/venv/bin/python -m pip install --upgrade pip' command.
(venv) asya@asya-HP-Pavilion-dv6-Notebook-PC:~/PycharmProjects/slurm_neo4j$
```

То есть мы создаем приветствие, устанавливаем у этого узла параметр «Message» и возвращаем сообщением:

```
"RETURN a.message + ', from node ' + id(a)", message=message)
```

Выполним код. Устанавливается соединение с базой в контейнере, передаются данные:

```

File Edit View Navigate Code Refactor Run Tools VCS Window Help
slurm_neo4j main.py
Project
  slurm_neo4j ~/PycharmProjects/slurm_neo4j
    venv
    main.py
    slurm_mongo.py
    External Libraries
    Scratches and Consoles
main.py
12 def print_greeting(self, message):
13     with self.driver.session() as session:
14         greeting = session.execute_write(self._create_and_return_greeting)
15         print(greeting)
16
17 @staticmethod
18 def _create_and_return_greeting(tx, message):
19     result = tx.run("CREATE (a:Greeting) "
20                   "SET a.message = $message "
21                   "RETURN a.message + ', from node ' + id(a)", message=message)
22     return result.single()[0]
23
24
Run: main
/home/asya/PycharmProjects/slurm_neo4j/venv/bin/python /home/asya/PycharmProjects/slurm_neo4j/main.py
hello, world, from node 0
Process finished with exit code 0

```

- Hello, word, from node 0 – это message;
- From node – это текст, который должен печататься;
- 0 – это айдишник ноды.

Посмотрим на это в контейнере. Заходим в контейнер, открываем утилиту «Cypher-shell» и повторяем запрос, который отправляли ранее:

```

961b19492ea4 neo4j:5.3.0 "tini -g -- /startup..." 11 minutes ago Up About a minute
0:7687->7687/tcp, :::7687->7687/tcp myneo4j
a5a6651a93ee redis:latest "docker-entrypoint.s..." 8 days ago Created
smartintergrator-redis-1
684e49c6b237 postgres:14 "docker-entrypoint.s..." 8 days ago Created
smartintergrator-postgres-1
asya@asya-HP-Pavilion-dv6-Notebook-PC:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS
NAMES
961b19492ea4 neo4j:5.3.0 "tini -g -- /startup..." 11 minutes ago Up About a minute 7473-7474/tcp, 0.0.0.0
:7687->7687/tcp, :::7687->7687/tcp myneo4j
asya@asya-HP-Pavilion-dv6-Notebook-PC:~$ sudo docker exec -it myneo4j bash
root@961b19492ea4:/var/lib/neo4j# cypher-shell -u neo4j -p testtest
Connected to Neo4j using Bolt protocol version 5.0 at neo4j://localhost:7687 as user neo4j.
Type :help for a list of available commands or :exit to exit the shell.
Note that Cypher queries must end with a semicolon.
neo4j@neo4j> MATCH (n) RETURN count(n);
+-----+
| count(n) |
+-----+
| 1 |
+-----+

1 row
ready to start consuming query after 317 ms, results consumed after another 2 ms
neo4j@neo4j>

```

Теперь у нас не «0», а один узел. Если мы хотим получить содержимое, то вместо count пишем «n» и приветствие с сообщением «Hello, word».

Если запустить код еще раз, то есть создать еще одно приветствие, то меняется айдишник ноды (он увеличивается на единицу):

```

Run: main
/home/asya/PycharmProjects/slurm_neo4j/venv/bin/python /home/asya/PycharmProjects/slurm_neo4j/main.py
hello, world, from node 1
Process finished with exit code 0

```

Также добавляется еще одна строка:

```
+-----+
| (:Greeting {message: "hello, world"}) |
+-----+

1 row
ready to start consuming query after 181 ms, results consumed after another 15 ms
neo4j@neo4j> MATCH (n) RETURN n;
+-----+
| n |
+-----+
| (:Greeting {message: "hello, world"}) |
+-----+

1 row
ready to start consuming query after 13 ms, results consumed after another 5 ms
neo4j@neo4j> MATCH (n) RETURN n;
+-----+
| n |
+-----+
| (:Greeting {message: "hello, world"}) |
| (:Greeting {message: "hello, world"}) |
+-----+

2 rows
ready to start consuming query after 3 ms, results consumed after another 6 ms
```

Две строчки с одинаковым содержимым, но с разным айдишником.

Мы попробовали поработать с Neo4j. Теперь можно выйти из утилиты и контейнера и завершить работу контейнера через стоп.

Как вам урок?



Изучил, далее >

