



Бесплатная электронная книга

# УЧУСЬ

---

## neo4j

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

#neo4j



.....	8
.....	8
.....	8
neo4jrestclient.....	8
<b>"db",</b> .....	9
.....	9
<b>:</b> .....	9
.....	10

---

# Около

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [neo4j](#)

It is an unofficial and free neo4j ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official neo4j.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

---

# глава 1: Начало работы с neo4j

## замечания

В этом разделе представлен обзор того, что такое neo4j, и почему разработчик может захотеть его использовать.

Следует также упомянуть о любых крупных предметах в neo4j и ссылаться на связанные темы. Поскольку документация для neo4j является новой, вам может потребоваться создать начальные версии этих связанных тем.

## Examples

### Установка или настройка

Перейдите к [Install Neo4j](#), который должен обнаружить платформу ОС через ваш веб-браузер, загрузить и выполнить обычные инструкции по установке для вашей ОС.

Neo4j был создан с Java, поэтому будет работать на любой платформе с установленной Java, однако команда Neo4j упростила установку, предоставив простые установочные пакеты для популярной платформы (например, .dmg для Mac, .deb для Debian и Ubuntu, .exe для Windows 64 и 32-разрядных платформ ...).

*Чтобы просмотреть другие версии и доступные платформы, см. [Раздел «Другие выпуски Neo4j»](#)*

### Установка Neo4j в качестве контейнера докеров:

```
## Required : Docker machine, docker cli

# Pull neo4j image from the docker hub
docker pull neo4j

# create the docker container
docker run \
  --publish=7474:7474 --publish=7687:7687 \
  --volume=$HOME/neo4j/data:/data \
  neo4j

# If you are running docker directly on the host (e.g ubuntu, RHEL, CentOS etc)
#   Access the neo4j console at http://localhost:7474
# If you are on OSX/ Windows
#   Access the neo4j console at http://<docker-machine-ip>:7474
```

### Установка и запуск сервера Neo4j

Предварительные шаги:

- Установите Java на свой компьютер
- Посетите [сайт neo4j](#) и нажмите ссылку «Скачать Community Edition» или посетите [ссылку для загрузки](#) .
- Разархивируйте загруженный файл .tar в свой домашний каталог.

---

## Запустите Neo4j с консоли (без головы, без веб-сервера)

- Перейдите в подкаталог `/bin` извлеченной папки и выполните в `./neo4j console`
- Теперь вы можете выполнять запросы neo4j в терминале

---

## Запустить веб-сервер Neo4j

- Перейдите в подкаталог `/bin` извлеченной папки и выполните в терминале `./neo4j start`
- Посетите сайт <http://localhost:7474/>
- Только в первый раз вам придется войти в систему со своей учетной записью по умолчанию и изменить пароль по умолчанию. По версии сообщества 3.0.3, по умолчанию имя пользователя и пароль - neo4j и neo4j.
- Теперь вы можете вставить запросы Neo4j в консоль, предоставленную в вашем веб-браузере, и визуальнo исследовать результаты каждого запроса.

---

## Запустить веб-сервер Neo4j

Каждый сервер Neo4j в настоящее время (в редакции сообщества) может размещать одну базу данных Neo4j, поэтому для настройки новой базы данных:

- Посетите подкаталог `/bin` и выполните `./neo4j stop` чтобы остановить сервер.
- Перейдите в подкаталог `/conf` и отредактируйте файл `neo4j.conf` , изменив значение параметра `dbms.active_database` на имя новой базы данных, которую вы хотите создать.
- Перейдите снова в подкаталог `/bin` и выполните `./neo4j start`
- Веб-сервер снова запустился с новой пустой базой данных. Вы можете снова посетить <http://localhost:7474/> для работы с новой базой данных.
- Созданная база данных находится в подкаталоге `/data/databases` под папкой с именем, указанным в параметре `dbms.active_database` .

---

## Удалите одну из баз данных

- Убедитесь, что сервер Neo4j не запущен; перейдите в подкаталог `/bin` и выполните

```
./neo4j status . Если выходное сообщение показывает, что сервер запущен, также выполните ./neo4j stop .
```

- Затем перейдите в подкаталог / данные / базы данных и удалите папку базы данных, которую вы хотите удалить.

## Язык запросов Cypher

Это язык запросов Cypher, Neo4j. Во многих отношениях Cypher похож на SQL, если вы знакомы с ним, за исключением того, что SQL ссылается на элементы, хранящиеся в таблице, а Cypher - на элементы, хранящиеся на графике.

Во-первых, мы должны начать с изучения того, как создать график и добавить отношения, поскольку это по существу то, о чем говорит Neo4j.

```
CREATE (ab:Object { age: 30, destination: "England", weight: 99 })
```

- Вы используете **CREATE** для создания данных
- Чтобы указать узел, вы используете круглые скобки: ()
- Часть ab: Object может быть разбита следующим образом: переменная 'ab' и метка 'Object' для нового узла. Обратите внимание, что переменная может быть любой, но вы должны быть последовательны в строке Cypher Query
- Чтобы добавить свойства к узлу, используйте скобки: {} скобки

Затем мы узнаем о поиске MATCHes

```
MATCH (abc:Object) WHERE abc.destination = "England" RETURN abc;
```

MATCH указывает, что вы хотите найти определенный шаблон узла / отношения (abc: Object), относится к одному узлу Pattern (с меткой Object), который хранит совпадения в переменной abc. Вы можете думать о всей этой линии как о следующем

```
abc = find the matches that is an Object WHERE the destination is England.
```

В этом случае WHERE добавляет ограничение, которое является целью назначения в Англии. Вы должны включить возвращение в конце для всех запросов MATCH (neo4j не будет принимать только совпадение ... ваш запрос должен всегда возвращать некоторое значение [это также зависит от того, какой тип запроса вы пишете ... мы будем говорить больше об этом позже, когда мы вводим другие типы запросов, которые вы можете сделать]).

Следующая строка будет объяснена в будущем, после того как мы перейдем к некоторым элементам языка запросов Cypher. Это поможет вам понять, что мы можем сделать с этим языком! Ниже вы найдете пример, в который попадает фильм, название которого начинается с 'T'

```
MATCH (actor:Person)-[:ACTED_IN]->(movie:Movie)
WHERE movie.title STARTS WITH "T"
RETURN movie.title AS title, collect(actor.name) AS cast
ORDER BY title ASC LIMIT 10;
```

Полный список команд и их синтаксис можно найти на официальной [справочной карте Neo4j Cypher](#) .

## База данных отчетов VDG для РСУБД

RDBMS	Диаграмма базы данных
таблицы	диаграммы
Ряды	Вершины
Столбцы и данные	Свойства и его значения
Ограничения	Отношения
присоединяется	пересечение

Прочитайте Начало работы с neo4j онлайн: <https://riptutorial.com/ru/neo4j/topic/945/начало-работы-с-neo4j>

---

# глава 2: зашифровывать

## Вступление

Cypher - это язык запросов, используемый Neo4j. Вы используете Cypher для выполнения задач и совпадений с графиком Neo4j.

Cypher «вдохновлен SQL» и разработан с помощью интуитивного в том, как вы описываете отношения, т. Е. Обычно рисунок шаблона будет похож на представление Cypher шаблона.

## Examples

### Создание

---

## Создать узел

```
CREATE (neo:Company) //create node with label 'Company'

CREATE (neo:Company {name: 'Neo4j', hq: 'San Mateo'}) //create node with properties
```

---

## Создайте отношения

```
CREATE (beginning_node)-[:edge_name{Attribute:1, Attribute:'two'}]->(ending_node)
```

---

## Шаблоны запросов

Запуск neo4j локально, в GUI браузера (по умолчанию: [http://localhost:7474 / browser /](http://localhost:7474/browser/)), вы можете запустить следующую команду, чтобы получить палитру запросов.

```
:play query template
```

Это поможет вам приступить к созданию и объединению узлов и связей путем ввода запросов.

### Создать грань

```
CREATE (beginning_node)-[:edge_name{Attribute:1, Attribute:'two'}]->(ending_node)
```

### делеция

---

# Удалить все узлы

```
MATCH (n)
DETACH DELETE n
```

DETACH не работает в более старых версиях (менее 2,3), для предыдущих версий

```
MATCH (n)
OPTIONAL MATCH (n)-[r]-()
DELETE n, r
```

---

# Удалить все узлы определенной метки

```
MATCH (n:Book)
DELETE n
```

## Совпадение (группа захвата) и узлы с привязкой к ссылкам

```
Match (node_name:node_type {}), (node_name_two:node_type_two {})
CREATE (node_name)-[:edge_name{}]->(node_name_two)
```

## Обновить узел

```
MATCH (n)
WHERE n.some_attribute = "some identifier"
SET n.other_attribute = "a new value"
```

## Удалить все сиротские узлы

Сиротские узлы / вершины - это те, у кого отсутствуют все отношения / ребра.

```
MATCH (n)
WHERE NOT (n)--()
DELETE n
```

Прочитайте зашифровывать онлайн: <https://riptutorial.com/ru/neo4j/topic/3669/зашифровывать>

---

## глава 3: ПИТОН

### Examples

#### Установить neo4jrestclient

```
pip install neo4jrestclient
```

#### Подключиться к neo4j

```
from neo4jrestclient.client import GraphDatabase
db = GraphDatabase("http://localhost:7474", username="neo4j", password="mypass")
```

#### Создание некоторых узлов с метками

```
user = db.labels.create("User")
u1 = db.nodes.create(name="user1")
user.add(u1)
u2 = db.nodes.create(name="user2")
user.add(u2)
```

---

## Вы можете связать ярлык со многими узлами за один раз

```
Language = db.labels.create("Language")
b1 = db.nodes.create(name="C++")
b2 = db.nodes.create(name="Python")
beer.add(b1, b2)
```

#### Создать отношения

```
u1.relationships.create("likes", b1)
u1.relationships.create("likes", b2)
u2.relationships.create("likes", b1)
```

---

## Двунаправленные отношения

```
u1.relationships.create("friends", u2)
```

#### Сопоставление с использованием neo4jrestclient

```
from neo4jrestclient import client

q = 'MATCH (u:User)-[r:likes]->(m:language) WHERE u.name="Marco" RETURN u, type(r), m'
```

---

## "db", как определено выше

```
results = db.query(q, returns=(client.Node, str, client.Node))
```

---

## Результаты поиска

```
for r in results:
    print("(%s)-[%s]->(%s)" % (r[0]["name"], r[1], r[2]["name"]))
```

---

## Выход:

```
(Marco)-[likes]->(C++)
(Marco)-[likes]->(Python)
```

Прочитайте питон онлайн: <https://riptutorial.com/ru/neo4j/topic/5990/питон>

---

## кредиты

S. No	Главы	Contributors
1	Начало работы с neo4j	<a href="#">Andrew Lank</a> , <a href="#">Community</a> , <a href="#">cs_user2017</a> , <a href="#">Dimos</a> , <a href="#">Frank Pavageau</a> , <a href="#">Prosen Ghosh</a> , <a href="#">TinkerBotFoo</a> , <a href="#">wintersolider</a>
2	зашифровывать	<a href="#">frant.hartm</a> , <a href="#">Govind Singh</a> , <a href="#">JOG</a> , <a href="#">Liam</a> , <a href="#">Nicole White</a> , <a href="#">Renato Dinhani</a> , <a href="#">SerialDev</a> , <a href="#">TJ Walker</a>
3	питон	<a href="#">SerialDev</a>