The Neo4j Graph Platform

Contents

About this module

The Neo4j Graph Platform enables developers to create applications that are best architected as graph-powered systems that are built upon the rich connectedness of data.
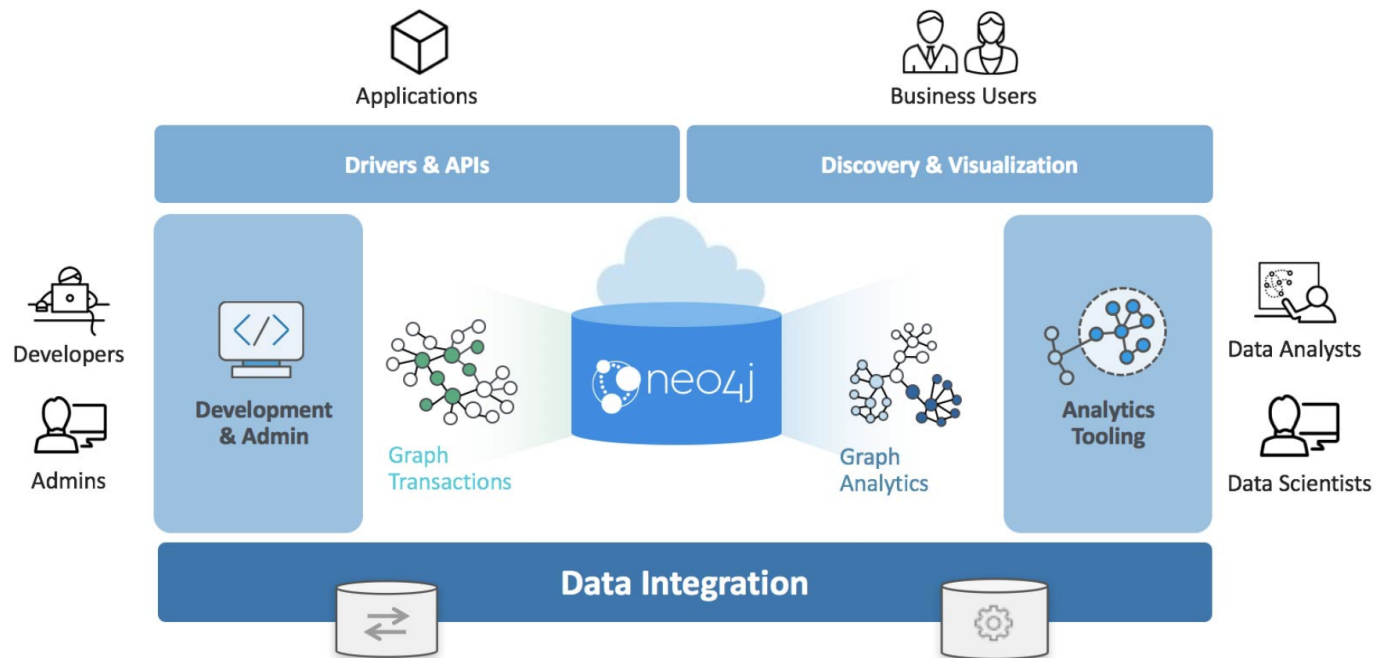
At the end of this module, you will be able to:

- Describe the components and benefits of the Neo4j Graph Platform.

- Access a Neo4j database using Neo4j Browser

Need help? **Ask in the Neo4j Community** ↗

Neo4j Graph Platform

The Neo4j Graph Platform includes components that enable you to develop your graph-enabled application.

It is used by developers, administrators, data analysts, and data scientists to access application data. Developers create the data in the graph by either importing it into the graph or using the Cypher language to implement the data model. In addition, developers are responsible for integrating the graph with other systems and database management system (DBMS) installations. Admins manage the processes and files related to the Neo4j installation. Data scientists and data analysts typically use a combination of Cypher queries, as well as tools to analyze the data. End-users typically use applications written by developers to access the graph data.

Components of the Neo4j Graph Platform

To better understand the Neo4j Graph Platform, you will learn about these components and the benefits they provide.

- Neo4j DBMS

- Neo4j Aura

- Neo4j Sandbox

Need help? **Ask in the Neo4j Community** ↗

- Neo4j Desktop

- Neo4j graph applications

  - Neo4j Browser

  - Neo4j Bloom

- Neo4j libraries

- Neo4j drivers

- Neo4j integration

- Neo4j administration tools

  - cypher-shell

Neo4j DBMS

The heart of the Neo4j Graph Platform is the Neo4j DBMS. The Neo4j DBMS includes processes and resources needed to manage a single Neo4j instance or a set of Neo4j instances that form a cluster. A Neo4j instance is a single process that runs the Neo4j server code. A Neo4j instance at a minimum contains two databases, the system database and the default database, neo4j.
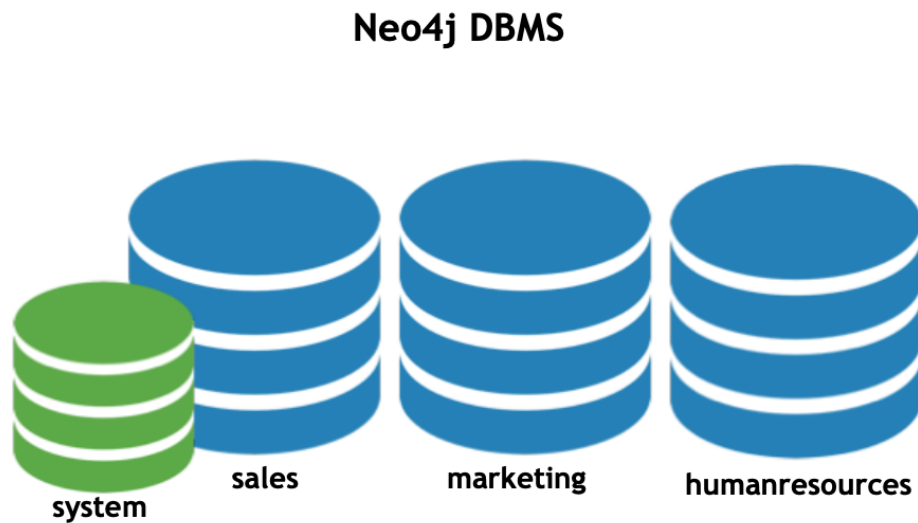
Neo4j DBMS

neo4j

system

The **system** database stores metadata about the databases for the installation, as well as security configuration. The **default** database (named neo4j by default) is the "user" database where you implement your graph data model.

Need help? [Ask in the Neo4j Community](#) ↗

Neo4j 4.x DBMS

In Neo4j Enterprise Edition 4.x, you may have more than one "user" database.

Here we have three "user" databases that hold the application data. You specify one of the databases as the default database.

Next, you will learn about some features of Neo4j DBMS that make it different from the traditional relational database management system (RDBMS).

Index-free adjacency

One of the key features that makes Neo4j graph databases different from an RDBMS is that Neo4j implements **index-free adjacency**.

To better understand the benefit of index-free adjacency, let's look at how a query executes in an RDBMS. Suppose you have this table in the RDBMS:

Primary Key   Foreign Key

ID	GROUP_NAME	PARENT_ID
1	Top group	1
2	Some group	1
3	Other group	2

Need help? Ask in the Neo4j Community ↗

You execute this SQL query to find the third-degree parents of the group with the ID of 3:

```sql
SELECT PARENT_ID
FROM GROUPS
WHERE ID = (SELECT PARENT_ID
    FROM GROUPS
    WHERE ID = (SELECT PARENT_ID
        FROM GROUPS
        WHERE ID = 3))
```

The result of this query is 1, but in order to determine this result, the query engine needed to:

1. Locate the innermost clause.

2. Build the query plan for the subclause.

3. Execute the query plan for the subclause.

4. Locate the next innermost clause.

5. Repeat Steps 2-4.

Resulting in:

- 3 planning cycles

- 3 index lookups

- 3 DB reads

Neo4j DBMS: Index-free adjacency (IFA)

With index-free adjacency, Neo4j stores nodes and relationships as objects that are linked to each other. Conceptually, the graph looks as follows:
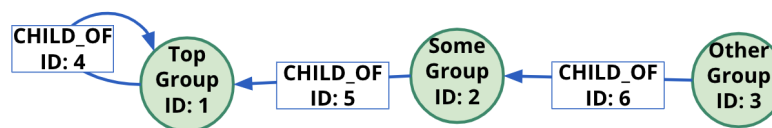

These nodes and relationships are stored as follows:

Need help? Ask in the Neo4j Community ↗

	1	2	3	4	5	6
	Top Group	**Some Group**	**Other Group**	**CHILD_OF**	**CHILD_OF**	**CHILD_OF**
	Links 4, 5	Links 5, 6	Links 6	Links 1,1	Links 2,1	Links 3,2

Suppose we had this query in Cypher:

Cypher

```
MATCH (n) <-- (:Group) <-- (:Group) <-- (:Group {id: 3})
RETURN n.id
```

Using IFA, the Neo4j query engine starts with the anchor of the query which is the Group node with the id of 3. Then it uses the links stored in the relationship and node objects to traverse the graph pattern.
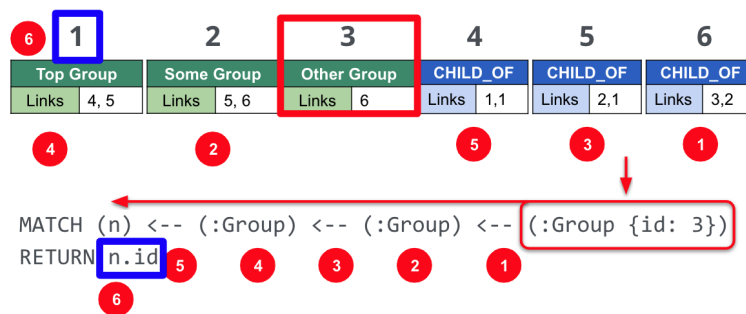
To perform this query, the Neo4j query engine needed to:

- Plan the query based upon the anchor specified.

- Use an index to retrieve the anchor node.

- Follow pointers to retrieve the desired result node.

Using IFA, there are:

- Much fewer index lookups or table scans

- Reduced duplication of foreign keys

Neo4j DBMS: Graph engine

Need help? [Ask in the Neo4j Community](#) ↗

The heart of the Neo4j Graph Platform is the Neo4j Database.

The Neo4j graph engine interprets Cypher statements to:

- Execute kernel-level code for optimal performance.
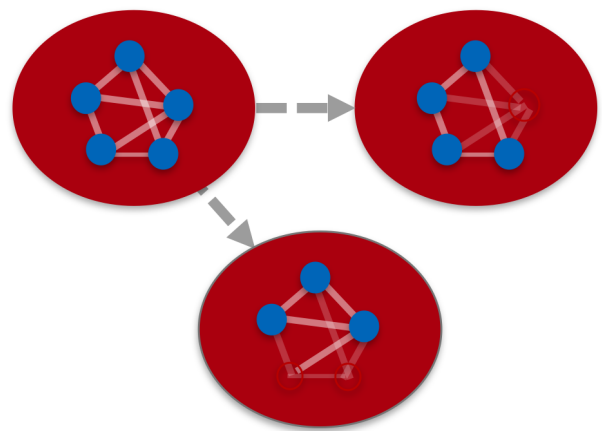
- Optimize data access on disk and cached in memory.

The graph engine has been improved with every release of Neo4j to provide the most efficient access to an application's graph data. There are many ways that you can tune the performance of the engine to suit your particular application needs.

Neo4j DBMS: ACID transactions
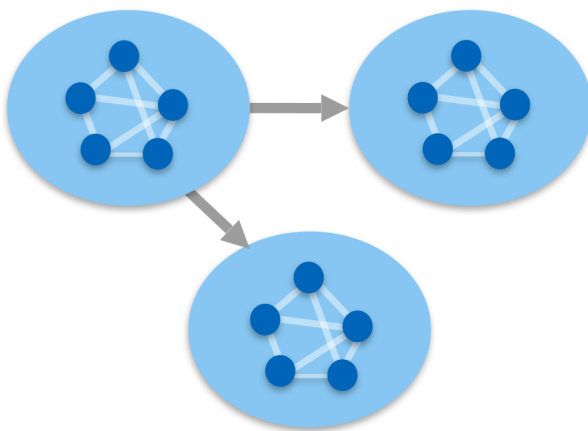

- Atomicity

- Consistency

- Isolation

- Durability

Transactionality is very important for robust applications that require an ACID (atomicity, consistency, isolation, and durability) guarantee for their data. If a relationship between nodes is created, not only is the relationship created, but the nodes are updated as connected. All of these updates to the database must all succeed or fail.

Neo4j DBMS: Clusters
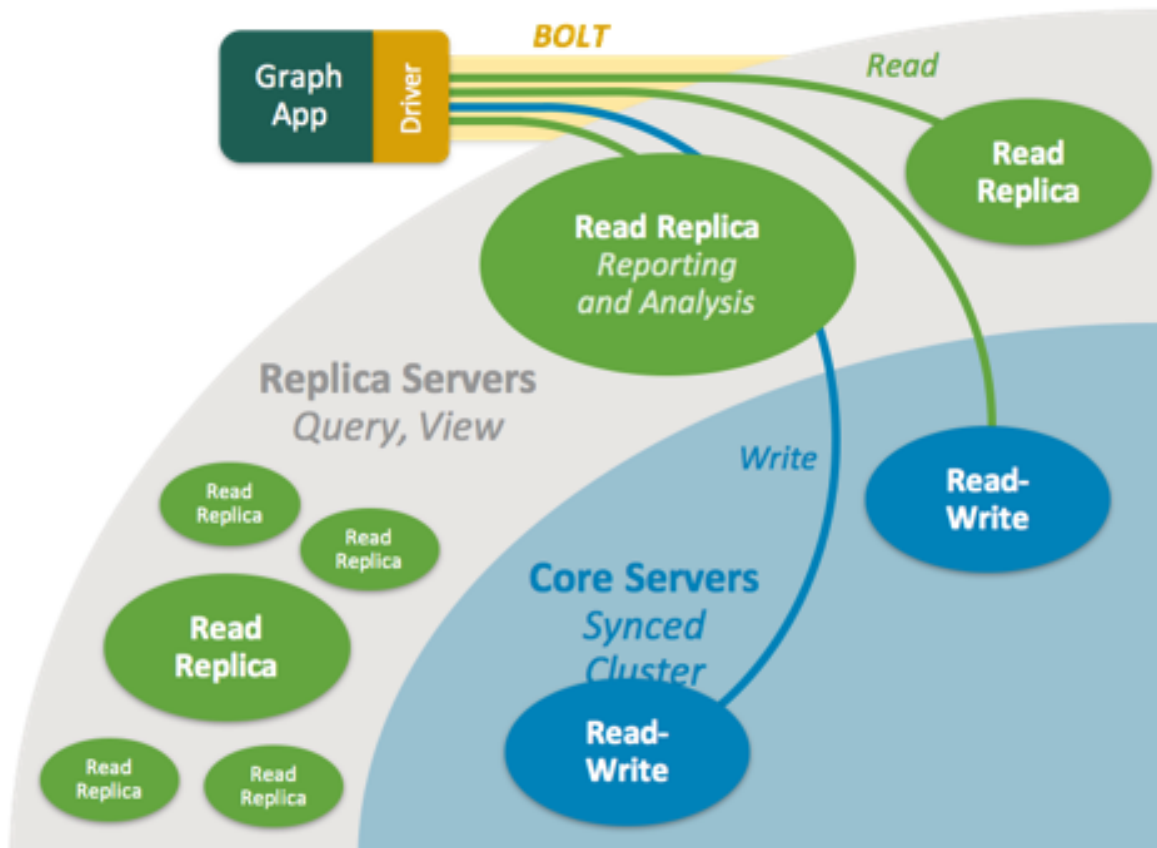
Neo4j supports clusters that provide high availability, scalability for read access to the data, and failover which is important to many enterprises. Neo4j clusters also maintain ACID transactions across all locations. Neo4j clusters are only available with Neo4j Enterprise Edition.

Neo4j Aura

Need help? Ask in the Neo4j Community ↗

Neo4j Aura is the simplest way to run the Neo4j DBMS in the cloud.

- The Neo4j team manages the administration of Neo4j.

- Developers focus on creating Neo4j applications.

Completely automated and fully-managed, Neo4j Aura delivers the world's most flexible, reliable and developer-friendly graph database as a service. With Neo4j Aura, you leave the day-to-day management of your database to the same engineers who built Neo4j, freeing you to focus on building rich graph-powered applications. Backups are done automatically for you and the database is available 24x7. In addition, the Neo4j Aura team will ensure that the database instance is always up-to-date with the latest version of Neo4j. To use Neo4j Aura, you must pay a monthly subscription fee which is based upon the size of your graph.

Once you create a Neo4j Database at the Neo4j Aura site ↗ , it will be managed by Neo4j.

Here is a short video that shows how to create a database in Neo4j Aura:

Need help? Ask in the Neo4j Community ↗

Neo4j Sandbox

The Neo4j Sandbox is a way that you can begin development with Neo4j. It is a free, temporary, and cloud-based instance of a Neo4j Server with its associated graph that you can access from any Web browser. The database in a Sandbox may be empty or it may be pre-populated. It is started automatically for you when you create the Sandbox.
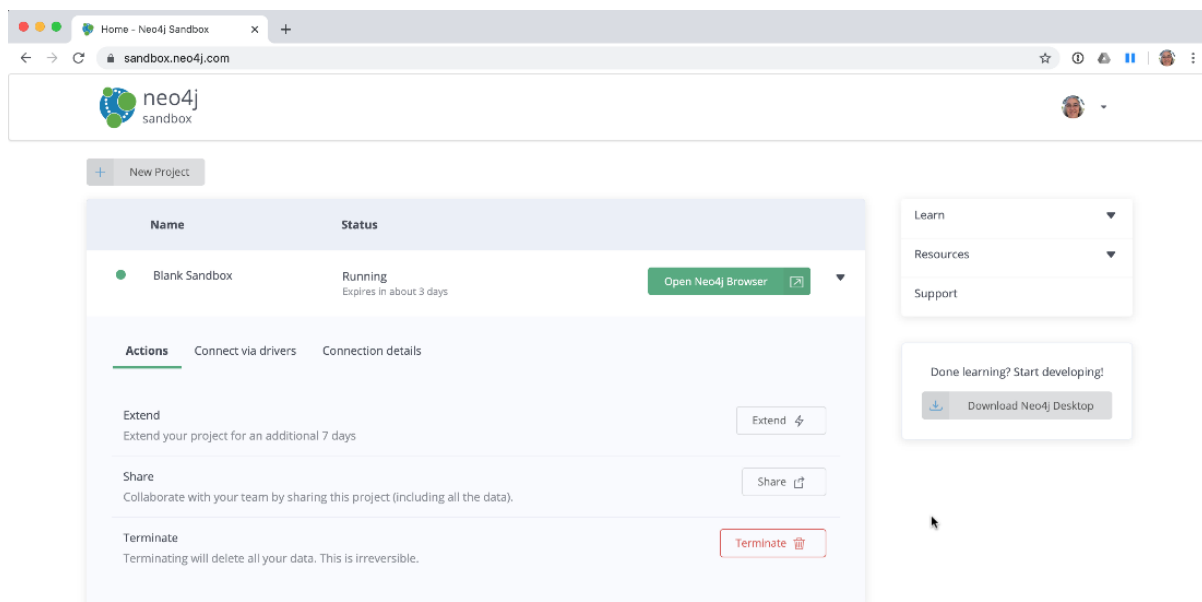
By default, the Neo4j Sandbox is available for three days, but you can extend it for up to 10 days. If you do not want to install Neo4j Desktop on your system, consider creating a Neo4j Sandbox. You must make sure that you extend your lease of the Sandbox, otherwise you will lose your graph and any saved Cypher scripts you have created in the Sandbox. However, you can use Neo4j Browser Sync to save Cypher scripts from your Sandbox. We recommend you use the Neo4j Desktop or Neo4j Aura for a real development project. The Sandbox is intended as a temporary environment or for learning about the features of Neo4j as well as specific graph use-cases.

You create a Sandbox by creating an account at the [Neo4j Sandbox site](#) ↗ .

Need help? [Ask in the Neo4j Community](#) ↗

Here is a video that shows how to create a Neo4j Sandbox account and a Neo4j Sandbox instance:

Creating a Blank Neo4j Sandbox

Neo4j Desktop

Neo4j Desktop is intended for developers who want to develop a Neo4j application and test it on their local machine. It is free to use. Neo4j Desktop is a UI that enables you to create projects, each with its own Neo4j DBMS instances where you can easily add or remove graph applications and libraries for use with your Neo4j DBMS. It includes an application called Neo4j Browser which is the UI you use to access the started database using Cypher queries.

Neo4jDesktop

The Neo4j Desktop runs on OS X, Linux, and Windows. You can download it from our **download page** ↗ . Follow the instructions on the download page for installing Neo4j Desktop.

> Before you install on Windows, make sure you have the latest version of PowerShell installed.

This video shows how to get started using Neo4j Desktop 1.4.1 after you have installed it.

Need help? **Ask in the Neo4j Community** ↗

GettingStartedDesktop I 4 I

Neo4j graph applications

Graph applications provide specific functionality to users that make their roles as developers, administrators, data scientists, or data analysts easier. Some of them are Web browser-based and some run in their own JVM. Graph applications are written by Neo4j engineers or Neo4j community members. Many of the graph applications supported by Neo4j are the work of **Neo4j Labs** ↗ . Some graph applications are supported by Neo4j and some are not, so you must be aware of the type of support you can receive for a particular graph application. You typically install graph applications from your Neo4j Desktop environment from **https://install.graphapp.io/** ↗ .

Here are some Neo4j graph applications:

GraphApps

Some Neo4j graph applications

Here are <u>some</u> of the graph applications you can use:

- **Neo4j Browser**
  - UI for testing Cypher queries and visualizing the graph.

Need help? **Ask in the Neo4j Community** ↗

- Neo4j Bloom

    - A tool for exploring graphs and generating Cypher code.

- Neo4j ETL Tool

    - UI for connecting to a data source to import into the graph.

- Halin

    - Monitor your Neo4j DBMS.

- Query Log Analyzer

    - Analyze queries that are executed on your system.

- Neo4j Cloud Tool

    - Tools for working with Neo4j Aura.

- Data Science Playground

    - Run graph algorithms and generate code for them.

Accessing Neo4j Browser

Within Neo4j Desktop, you can connect to a started database by simply opening it (Clicking the **Open** button for the started database). Another way is by opening the graph apps sidebar and selecting Neo4j Browser.
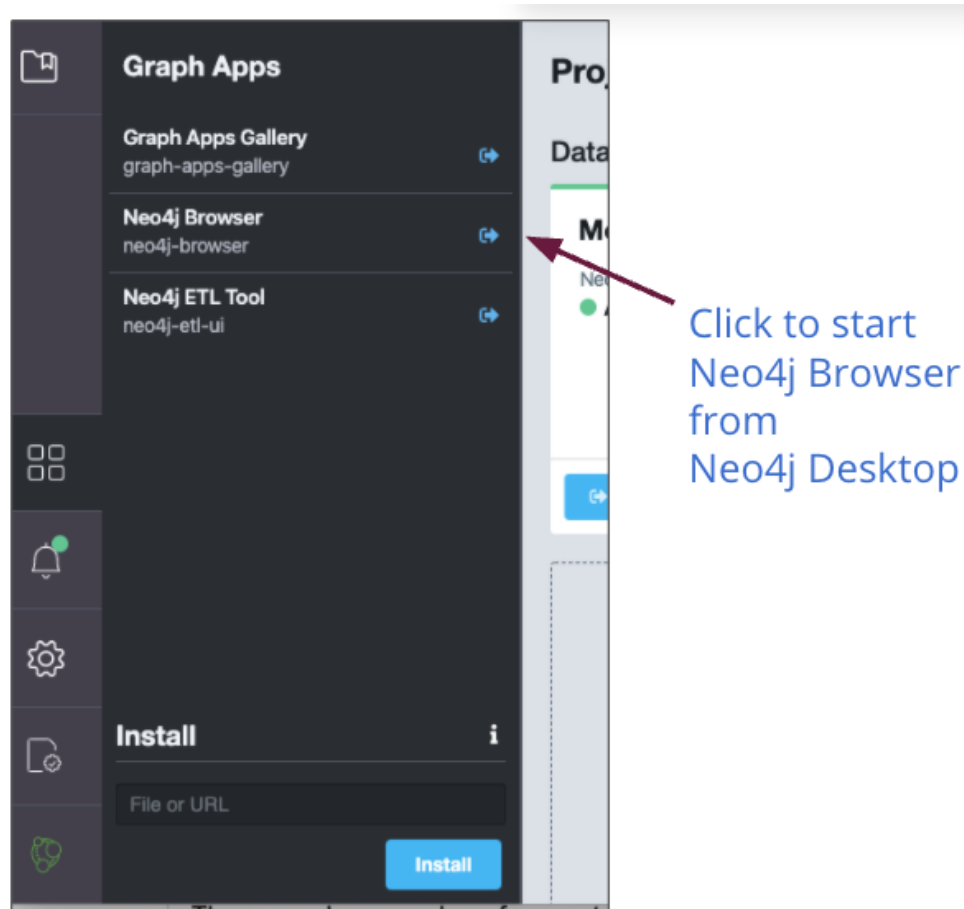
Runs as a graph app from Neo4j Desktop.

Web browser interface can also be used for connecting to:

- Neo4j Desktop instance that is started

- Neo4j Sandbox instance

- Neo4j Aura instance

Neo4j Browser

Neo4jBrowserFollows

Neo4j Browser is a Neo4j-supported tool that enables you to access a Neo4j Database by executing Cypher statements to create or update data in the graph and to query the graph to return data. The data returned is typically visualized as nodes and relationships in a graph, but can also be displayed as tables. In addition to executing Cypher statements, you can execute a number of system calls that are

Need help? Ask in the Neo4j Community ↗

related to the database being accessed by the Browser. For example, you can retrieve the list of queries that are currently running in the server.

1. Open Neo4j Browser from Neo4j Desktop

Neo4j Browser is a graph application that comes with Neo4j Desktop. You typically use it to access a database that is running locally, but you can use it to access a remote database.

If you save your frequently-used Cypher code in **Favorites**, you can download them so you can use them elsewhere (like in your application code).

Neo4jBrowserFavorites

2. Use Neo4j Browser Web browser interface

You can use the Web interface to access a local Neo4j DBMS, or a database in Neo4j Aura or Neo4j Sandbox.

WebNeo4jBrowser

Just as in the Neo4j Browser application, you can save frequently-used Cypher code in **Favorites**, then you can download them so you can use them elsewhere (like in your application code). In addition, when using the Web browser interface to Neo4j Browser, you can use Browser Sync to keep your favorites in the Cloud.

Guided Exercise: Getting Started with Neo4j Browser

Follow along with this video to become familiar with common tasks in Neo4j Browser.

Get Started with Neo4j Browser for Neo4j 4.0 (Query, Update, Visualize)

Need help? Ask in the Neo4j Community ↗

> Before you perform the tasks shown in this video, you must have either created and started the database in the Neo4j Desktop, created a Database in Neo4j Aura, or created a Neo4j Sandbox.

Neo4j Bloom

Neo4j Bloom is a Neo4j-supported graph application where you can experience:

- Visual presentation of your graph data tangibly reveals non-obvious connections.

- Easy-to-understand visualizations explain data connectedness to every colleague.

- Codeless search tools let you quickly explore your data without technical expertise.

- Browsing tools make it easy for you to discover new insights from your data.

    ![Bloom]

Visit the **Bloom page** ↗ to learn more about Neo4j Bloom.

Another way that you can try Neo4j Bloom is to create a Bloom Visual Discovery Sandbox that you can use for up to 10 days.

Libraries

Just as there are graph applications written by Neo4j engineers and Neo4j community members, there are libraries you can incorporate into your application. A library is also called a plug-in as it is used to extend what you can do in Cypher. Some libraries are available in Neo4j Desktop, while you must download and install other libraries.

Functions and procedures from the Neo4j 3.5 Graph Algorithms Library are officially supported by Neo4j as the Graph Data Science Library (GDS) in Neo4j 4.x.

One of the most popular libraries that is used by most developers is Awesome Procedures of Cypher (APOC). This library has close to 500 procedures and functions that extend Cypher in ways that make your programming in Cypher much easier for complex tasks. Since APOC is so widely-used by developers, it comes already-installed in a Neo4j Sandbox and Neo4j Aura.

Another library that also comes with Neo4j Desktop is GraphQL. GraphQL is an open-source query language for querying parts of a graph. It is not as flexible or powerful as Cypher, but it is used by some applications.

A very popular library for graph visualization is neoviz.js, another project of Neo4j Labs.

Here are the plugins that come with Neo4j Desktop:

Plugins

When you install a plugin to be used with a Neo4j DBMS, it must be a version of the plugin that is compatible with the version of the Neo4j DBMS.

Drivers and language support

Here are some drivers that Neo4j supports:

Drivers

Because Neo4j is open source, you can delve into the details of how the Neo4j Database is accessed, but most developers simply use Neo4j without needing a deeper understanding of the underlying code. Neo4j provides a full stack that implements all levels of access to the database and clustering layer where you can use our published APIs. The language used for querying the Neo4j database is Cypher, an open source language.

In addition, Neo4j supports Java, JavaScript, Python, C#, and Go drivers out-of-the box that use Neo4j's bolt protocol for binary access to the database layer. Bolt is an efficient binary protocol that compresses data sent over the wire as well as encrypting the data

Need help? **Ask in the Neo4j Community** ↗

that uses the Bolt driver to access the Neo4j database, and the application may use other packages that allow data integration between Neo4j and other data stores or uses a common framework such as spring. You download drivers from the Neo4j driver download page ↗ .

It is also possible for you to develop your own server-side extensions in Java that access the data in the database directly without using Cypher. The Neo4j community has developed drivers for a number of languages including Ruby, PHP, and R.

You can also extend the functionality of Neo4j by creating user defined functions and procedures that are callable from Cypher.

Neo4j integration

Neo4j has integrations with many systems in the IOT ecosystem. Neo4j can be part of a system that uses GRANDstack, Kettle, Docker, and many others. How you integrate Neo4j into a larger system will depend on how you intend to use Neo4j. Neo4j engineers and Community members have worked through some of the challenges of integration and their discussions and work can be found on the Neo4j User slack channel ↗ , the Neo4j online forum ↗ , stack overflow ↗ , and on Github ↗ .

One Neo4j-supported integration that you can download enables data to be streamed to/from Kafka ↗ .

Administration tools

Developers and administrators use command-line tools for managing the Neo4j DBMS. The three main tools used that are part of the Neo4j installation (located in the **bin** directory) include:

Tool	Description
cypher-shell	Create, start, stop, and drop a particular database as well as query the "user" database.
neo4j	Start, stop and retrieve the status of the Neo4j DBMS instance.
neo4j-admin	Create, copy, remove, backup, restore and perform other administrative tasks.

Need help? Ask in the Neo4j Community ↗

> Neo4j Desktop also supports creating, starting, and stopping databases.

cypher-shell

cypher-shell

`cypher-shell` is part of the Neo4j installation and is located in the **bin** directory. It is a command-line tool that you can use to connect to a Neo4j DBMS instance and run Cypher statements against the database.

It is useful if you want to create scripts that automatically run against the database(s). It is commonly used for advanced query tuning.

Even if you have not installed Neo4j, you can download and install cypher-shell as a stand-alone application if you want to connect to a running database and execute Cypher queries.

Check your understanding

Question 1

What are some of the benefits provided by the Neo4j DBMS?

Select the correct answers.

- ☐ Clustering
- ☐ ACID
- ☐ Index-free adjacency
- ☐ Optimized graph engine

Question 2

What libraries are available in Neo4j Desktop for developing your application?

Select the correct answers.

Need help? Ask in the Neo4j Community ↗

- [ ] APOC

- [ ] JGraph

- [ ] Graph Algorithms (Neo4j 3.5) or Graph Data Science (Neo4j 4.x)

- [ ] GraphQL

Question 3

What are some of the language drivers that come with Neo4j out of the box?

Select the correct answers.

- [ ] Java

- [ ] Ruby

- [ ] Python

- [ ] JavaScript

Summary

You can now:

- Describe the components and benefits of the Neo4j Graph Platform.

✔ Check Answers

Contents

Need help? Ask in the Neo4j Community ↗

Contact Us →

US: 1-855-636-4532

Sweden +46 171 480 113

UK: +44 20 3868 3223

France: +33 (0) 8 05 08 03 44

Learn

Sandbox

Neo4j Community Site

Neo4j Developer Blog

Neo4j Videos

Social

Twitter

Meetups

Github

Need help? Ask in the Neo4j Community ↗

GraphAcademy

Neo4j Labs