




# Building a Graph Solution in Neo4j

# Agenda

- Logistics
- Introduction
- Use Case Explanation
- Modeling
- BREAK
- Building the solution
- Q & A

That's right!  
Time for Tea/Coffee

# Logistics

WIFI Access:	etcvenues / stayconnected
Restrooms:	In corridor to your right (my left)
Chargers/Sockets:	Under <i>some</i> of the desks
Material for the workshop:	<a href="https://github.com/cskardon/gsummit2023">https://github.com/cskardon/gsummit2023</a> 
More training:	luke.masters@neo4j.com



# Introduction

What is a property graph?



Good news!

you only need to know

4

things

# Graph components

## Node (Vertex)

- The main data element from which graphs are constructed

Keanu  
Reeves

The  
Matrix

# Graph components

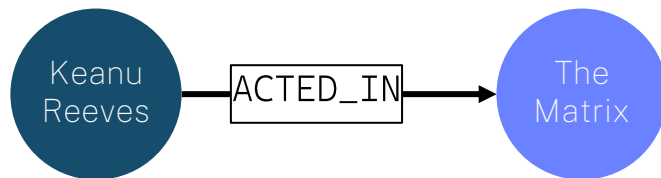
## Node (Vertex)

- The main data element from which graphs are constructed

## Relationship (Edge)

- A link between two nodes
  - Direction
  - Type

A node without relationships is permitted, a relationship without nodes is not





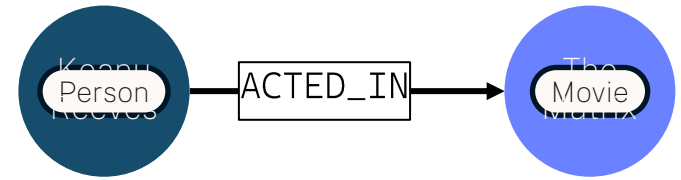
# Property graph database

Node (Vertex)

Relationship (Edge)

Label

- Define node role (optional)



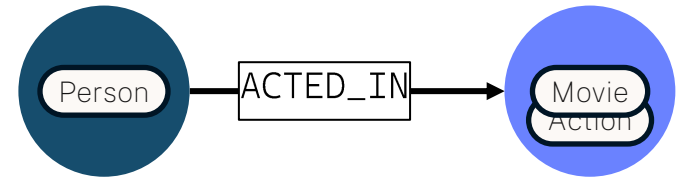
# Property graph database

Node (Vertex)

Relationship (Edge)

Label

- Define node role (optional)
- Can have more than one



# Property graph database

Node (Vertex)

Relationship (Edge)

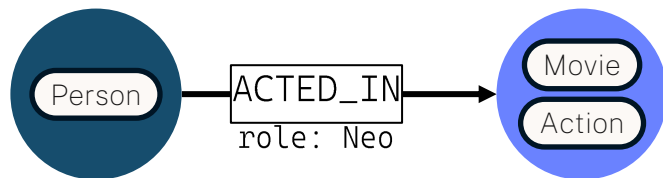
Label

- Define node role (optional)
- Can have more than one

Properties

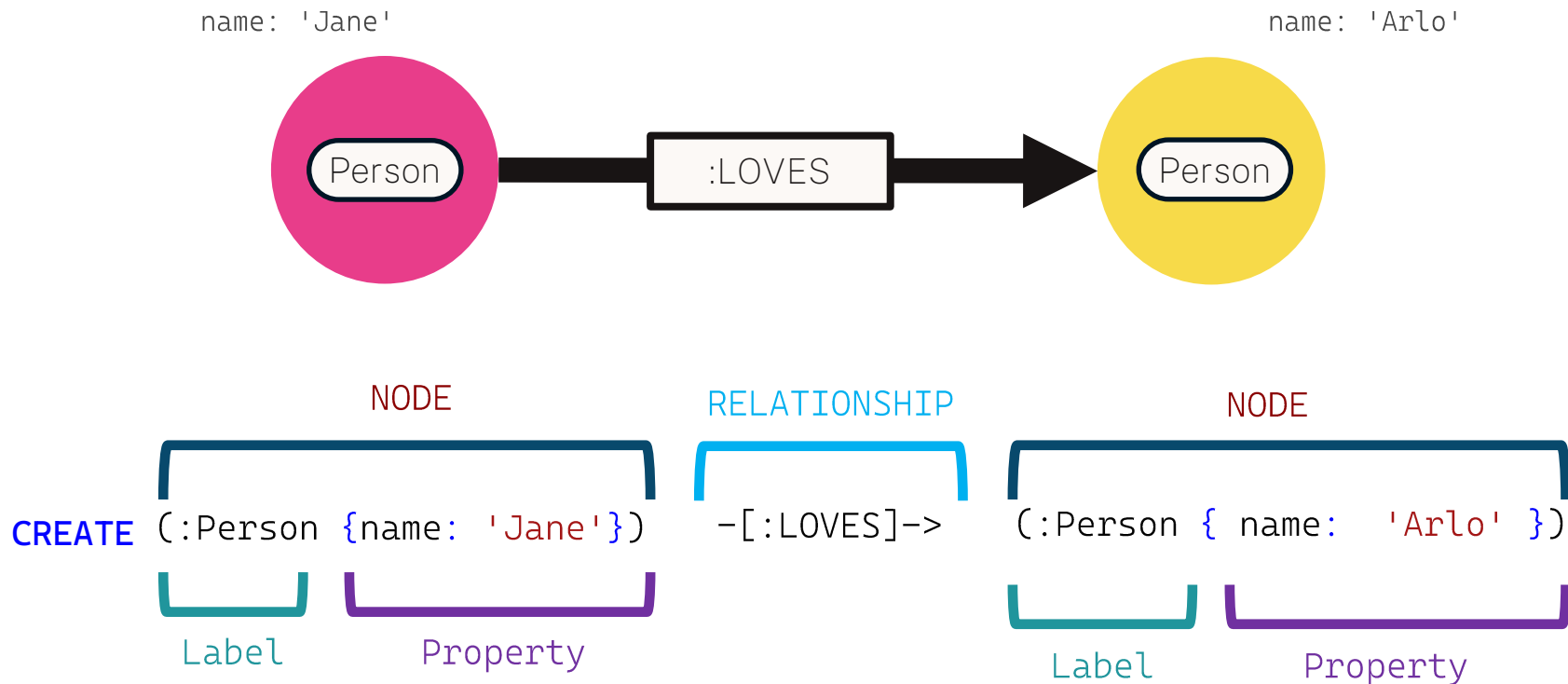
- Enrich
  - nodes
  - relationships

name: Keanu Reeves



title: The Matrix  
released: 1999  
tagline: Welcome...

# Cypher: powerful and expressive query language



# Cypher: Matching

name: 'Jane'



Diagram illustrating the components of the Cypher query:

```
MATCH (p:Person {name: 'Jane'}) -[:MARRIED_TO]-> (spouse:Person)
```

Labels and their corresponding parts:

- NODE** (red): Points to the first node pattern `(p:Person {name: 'Jane'})`.
- RELATIONSHIP** (blue): Points to the relationship pattern `-[:MARRIED_TO]->`.
- NODE** (red): Points to the second node pattern `(spouse:Person)`.
- Variable** (green): Points to `p`.
- Label** (green): Points to `Person`.
- Property** (purple): Points to `{name: 'Jane'}`.
- Variable** (green): Points to `spouse`.

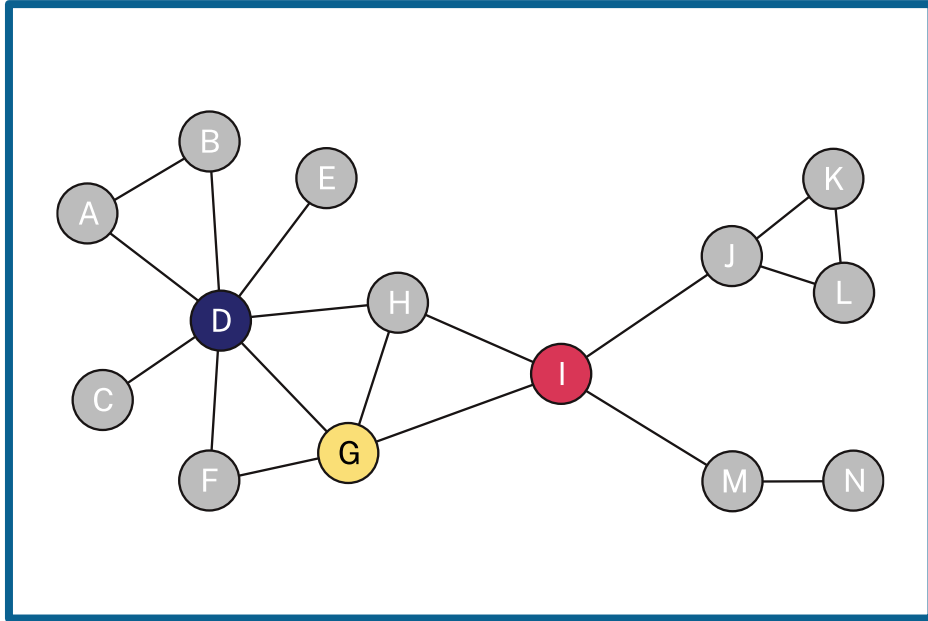
**RETURN** p, spouse



# Neo4j Graph Data Science

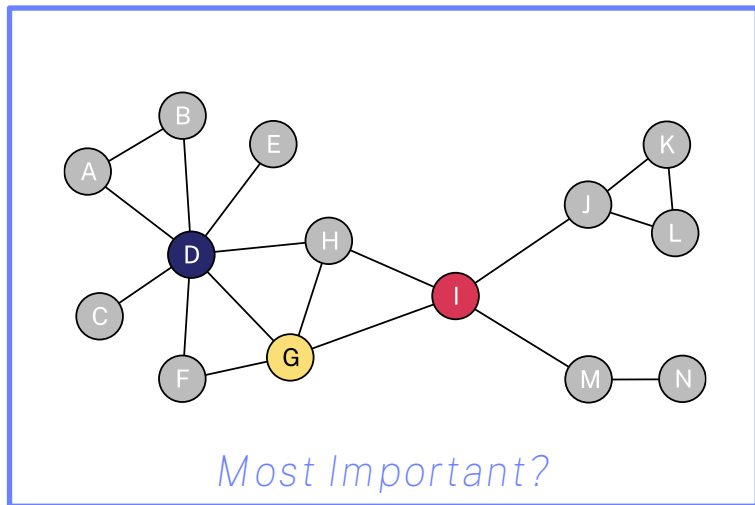


# Pop Quiz



*Which of the coloured nodes would be considered the most “important”?*

# Graphs Contain Implicit Knowledge



D has the highest degree centrality (7)  
This is the most connected individual in the network. If important is now well you are personally known, you pick D.



G has the highest closeness centrality (0.52)  
Information will disperse through the network more quickly through this individual. If you need to get a message out rapidly, choose G.

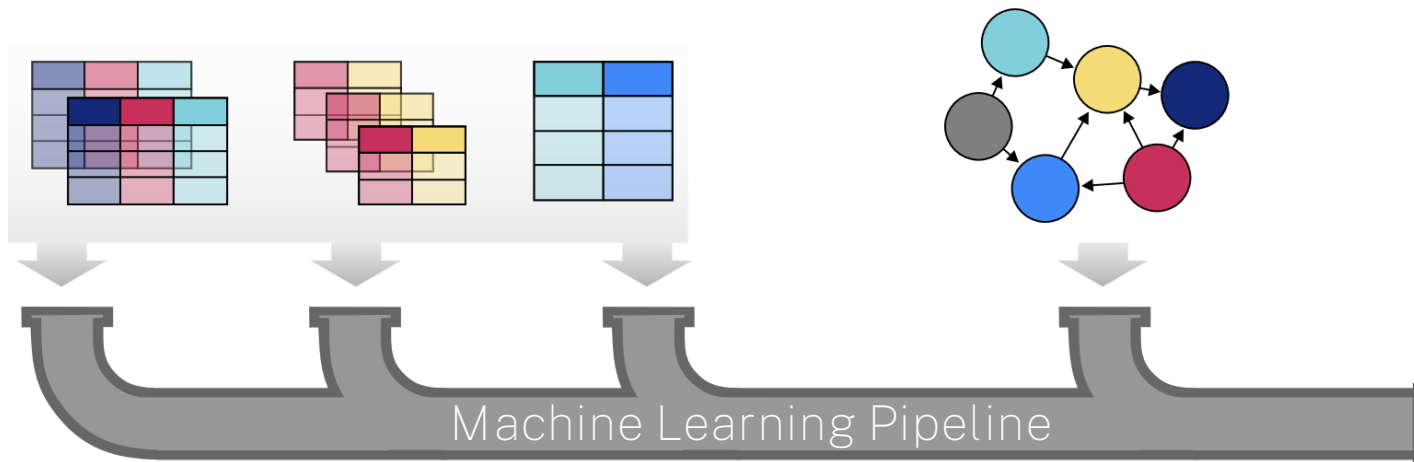


I has the highest betweenness centrality (0.59)  
This person is an efficient connector of other people. Risk of network disruption is higher if you lose this individual



# Better Predictions With Data You Already Have

- Traditional ML ignores network structure because it's difficult to extract
- Graphs use relationships to unlock otherwise unattainable predictions
- Add graph data to existing ML pipelines to increase accuracy



# Graph Algorithm Categories



## Pathfinding & Search

Finds optimal paths or evaluates route availability and quality



## Centrality / Importance

Determines the importance of distinct nodes in the network



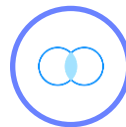
## Community Detection

Detects group clustering or partition



## Link Prediction

Estimates the likelihood of nodes forming a future relationship



## Similarity

Evaluates how alike nodes are by neighbours and relationships



## Embeddings & ML

Compute low-dimensional vector representations of nodes in a graph, and allow you to train supervised machine learning models



# Visualisation



# Data Visualization with Neo4j Bloom

Neo4j's user-friendly graph database visualization, exploration and collaboration tool.

- Visually explore graphs
- Prototype faster
- Visualize and discover
- Easy for non-technical users

The screenshot displays the Neo4j Bloom interface. On the left, a search bar contains the text "High risk bank account holders". Below it, a table lists properties for a selected node, including "balance", "bank account", and "phone number". The main area shows a graph visualization with nodes and edges. Nodes are color-coded and labeled with names like "John Doe", "Jane Smith", and "Bob Johnson". Edges represent relationships between these nodes. On the right, a sidebar shows a list of nodes and relationships, with a search bar and filters. The interface is designed for easy navigation and exploration of graph data.

**Search with type-ahead suggestions**

**Flexible Color, Size and Icon schemes**

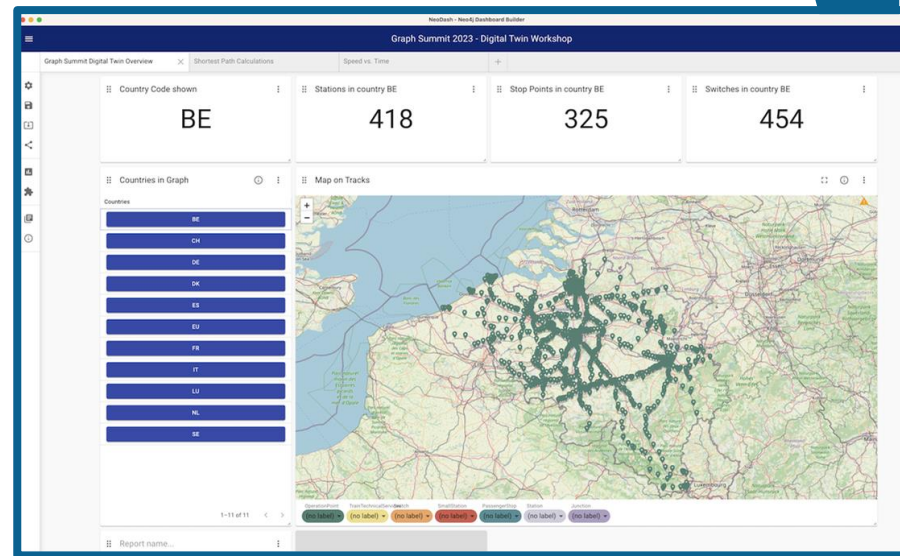
**Visualize, Explore and Discover**

**Pan, Zoom and Select**

**Property Browser and editor**

# NeoDash

- Fully open source
  - <https://github.com/neo4j-labs/neodash>
  - Extensible
- Can be integrated in existing portal
- Supported through Neo4j's Professional Services team





# graphsummit

## Use Case Explanation

### Digital Twin - An Overview



# What is a Digital Twin?



A Digital Twin is a digital representation of a [...] real-world physical product, system, or process [...] that serves as the effectively indistinguishable digital counterpart of it for practical purposes, such as [simulation](#), [integration](#), [testing](#), [monitoring](#) and [maintenance](#).





# Modeling a solution





# What is graph data modeling?

A *collaborative* effort where the application domain is analysed by stakeholders and developers to come up with the optimal model for use with Neo4j.

Stakeholders include:

- Business analysts
- Architects
- Managers
- Project leaders
- Data Scientists

# The Modeling Workflow



1. Derive the question



2. Obtain the data



3. Develop a model



4. Ingest the data



5. Query/Prove the model

no one  
EVER  
gets it right  
FIRST  
time

don't worry



# Modeling – Step 1

Domain knowledge – High level requirements



# We've already got this!

- (Normally)
- In this case we've talked about the Domain
- You will have the knowledge of your own Domain



# Modeling – Step 2

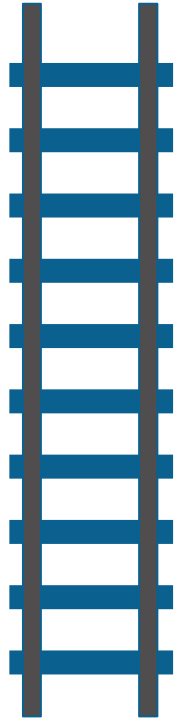
Sample Data



# Get Sample Data

## Static Rail Network\*

- Sections of lines
  - Length,
  - Speed
- Operational Points (Stations etc)
  - Geolocation information,
- Points of Interest (POI) along lines



\*Source: Register of Infrastructure (RINF) - <https://data-interop.era.europa.eu/>







# Modeling – Step 3

Domain Questions



# Data Modeling – Example Domain Questions

1. What is the route from Operational Point X to Operational Point Y?
  - What's the quickest way to get a repair crew from Technical Services to a given Switch?
2. What is an alternative route if an Operational Point on a route is closed?
  - A Switch is broken and we need to reroute Trains
3. How many routes are affected if I need to upgrade an Operational Point?
  - A Switch needs to be upgraded to support the network
4. What POIs are along a route?
  - Can we make revenue from referral commissions? Find busier routes during tourism season?



# Modeling – Step 4

Identifying entities and connections



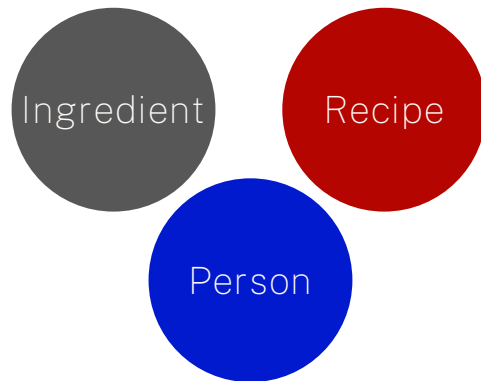
# Identify Entities from Questions

Entities are the nouns in the domain questions:

1. What ingredients are used in a **recipe**?

2. Who is married to this **person**?

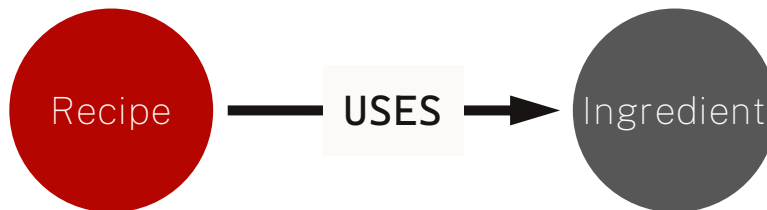
- The generic *nouns* often become labels in the model
- Use *domain knowledge* deciding how to further group or differentiate entities



# Identify Connections between Entities

Connections are the verbs in the domain questions:

- What ingredients are **used** in a recipe?



- Who is **married** to this person?

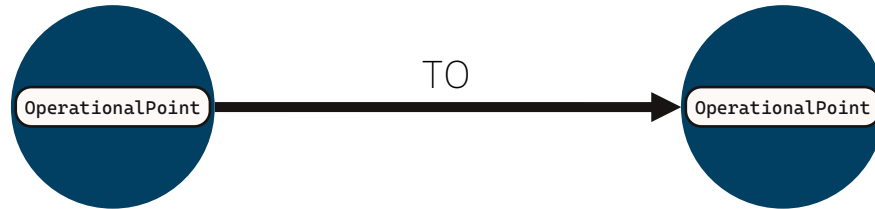


# Using our Questions – Question 1

1. What is the route **from** Operational Point X **to** Operational Point Y?
  - What's the quickest way to get a repair crew from Technical Services to a given Switch?
2. What is an alternative route if an Operational Point on a Section is closed?
  - A Switch is broken and we need to reroute Trains
3. How many routes are affected if I need to upgrade an Operational Point?
  - A Switch needs to be upgraded to support the network
4. What POIs are near Station Operational Points on a Section?
  - Can we make revenue from referral commissions? Find busier routes during tourism season?

# Using our Questions – Question 1 – Model

1. What is the route **from** Operational Point X **to** Operational Point Y?
  - What's the quickest way to get a repair crew from Technical Services to a given Switch?



# Using our Questions – Question 2

1. What is the route from Operational Point X to Operational Point Y?
  - What's the quickest way to get a repair crew from Technical Services to a given Switch?
2. What is an alternative route if an Operational Point on a Section is closed?
  - A Switch is broken and we need to reroute Trains
3. How many routes are affected if I need to upgrade an Operational Point?
  - A Switch needs to be upgraded to support the network
4. What POIs are near Station Operational Points on a Section?
  - Can we make revenue from referral commissions? Find busier routes during tourism season?



# Using our Questions – Question 2 – Model

2. What is an alternative route if an **Operational Point** on a **Section** is closed?
  - A Switch is broken and we need to reroute Trains



# Using our Questions – Question 3

1. What is the route **from** Operational Point X **to** Operational Point Y?
  - What's the quickest way to get a repair crew from Technical Services to a given Switch?
2. What is an alternative route if an Operational Point on a **Section** is closed?
  - A Switch is broken and we need to reroute Trains
3. How many routes are affected if I need to upgrade an Operational Point?
  - A Switch needs to be upgraded to support the network
4. What POIs are near Station Operational Points on a Section?
  - Can we make revenue from referral commissions? Find busier routes during tourism season?

# Using our Questions – Question 3 – Model

3. How many routes are affected if I need to upgrade an [Operational Point](#)?
- A Switch needs to be upgraded to support the network

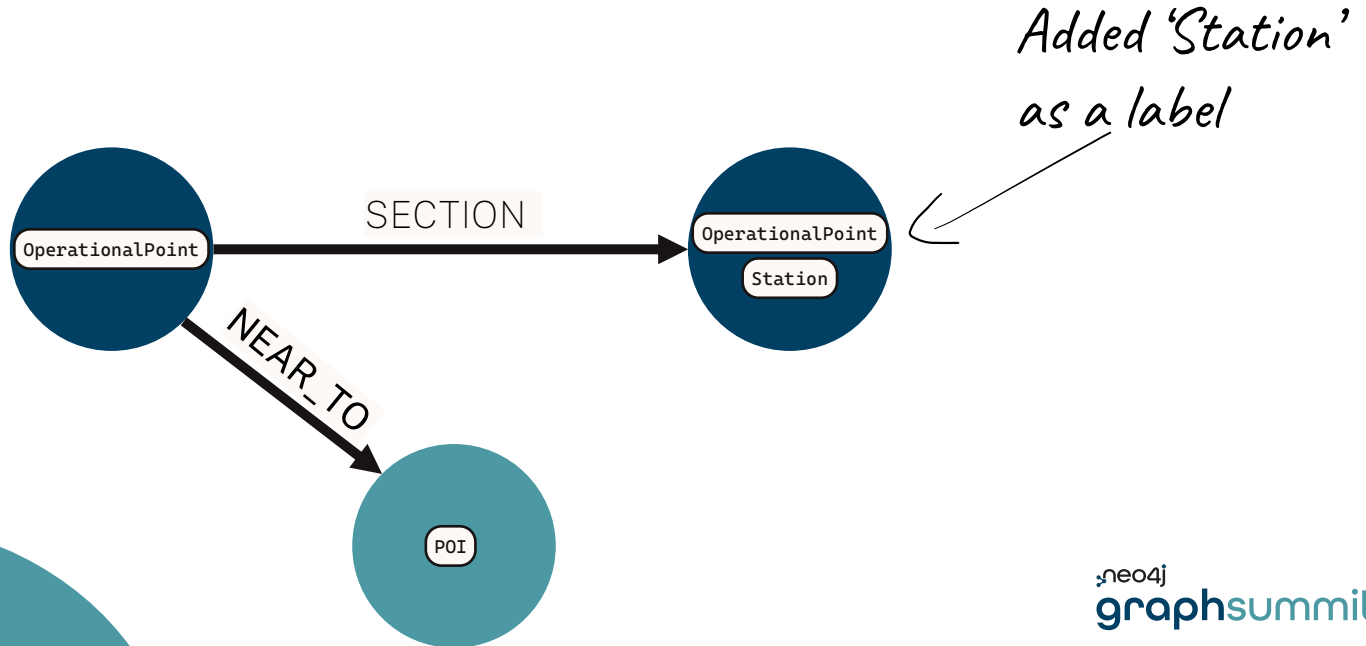


# Using our Questions – Question 4

1. What is the route **from** Operational Point X **to** Operational Point Y?
  - What's the quickest way to get a repair crew from Technical Services to a given Switch?
2. What is an alternative route if an Operational Point on a **Section** is closed?
  - A Switch is broken and we need to reroute Trains
3. How many routes are affected if I need to upgrade an Operational Point?
  - A Switch needs to be upgraded to support the network
4. What POIs are **near** Station Operational Points on a **Section**?
  - Can we make revenue from referral commissions? Find busier routes during tourism season?

# Using our Questions – Question 4 – Model

4. What POIs are near Station Operational Points on a Section?
- Can we make revenue from referral commissions? Find busier routes during tourism season?



# Hang on...

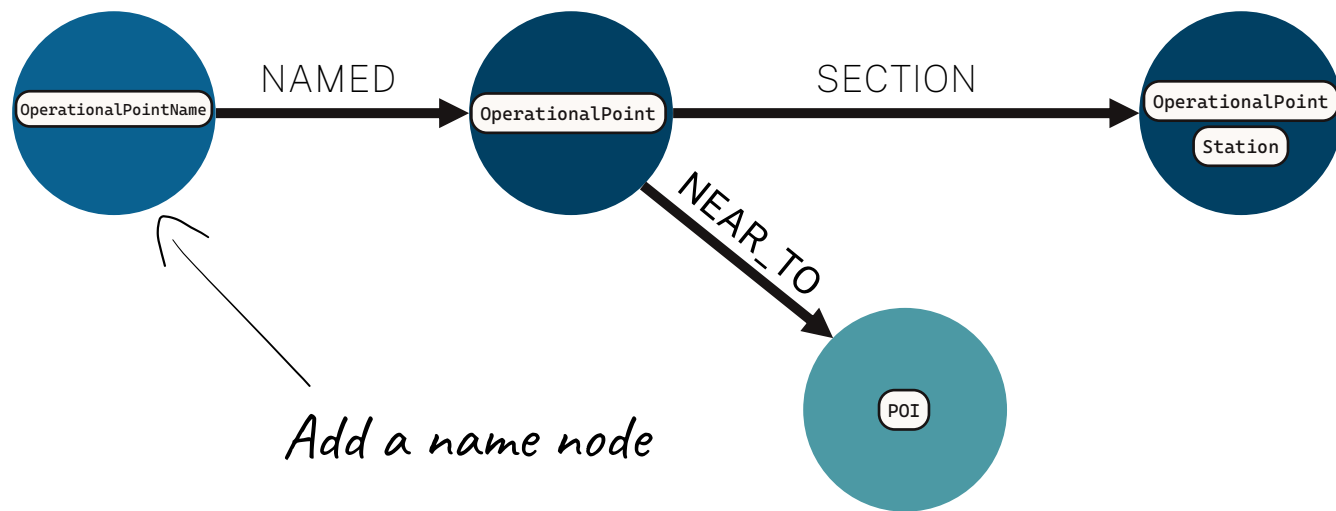
```
"EU00001", "BorderPoint", "Nieuweschans Grens", "53.189", "7.21112", "Netherlands"  
"EU00001", "BorderPoint", "Weener Grenze", "53.189", "7.21112", "Germany"
```

Wait, what?

We have duplicate entries

How do we  
deal with  
this?

# Name nodes



# How does that look in reality?

```
"EU00001", "BorderPoint", "Nieuweschans Grens", "53.189", "7.2112", "Netherlands"  
"EU00001", "BorderPoint", "Weener Grenze", "53.189", "7.2112", "Germany"
```

name: "Nieuweschans Grens"



country: "Netherlands"  
NAMED



country: "Germany"  
NAMED



name: "Weener Grenze"

id: "EU00001",  
geolocation: point(52.189, 7.2112)



# Whilst we're here, let's talk countries

- Multiple ways to model
  - Properties
  - Labels
- What should we choose?

Cheat mode  
enabled!

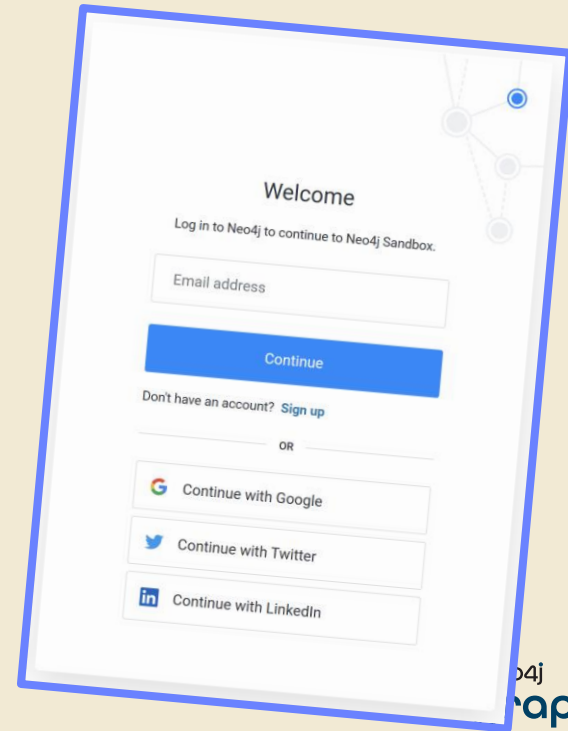
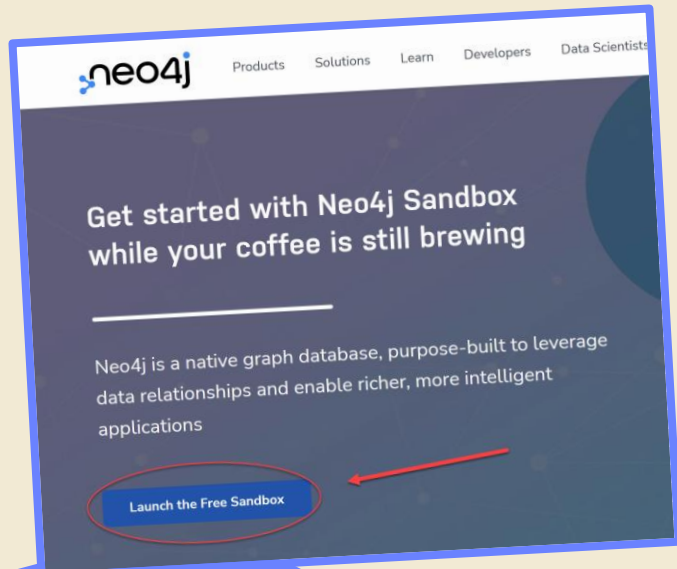
# Before we go further

## Environment for later

- Neo4j Desktop
- Neo4j Sandbox

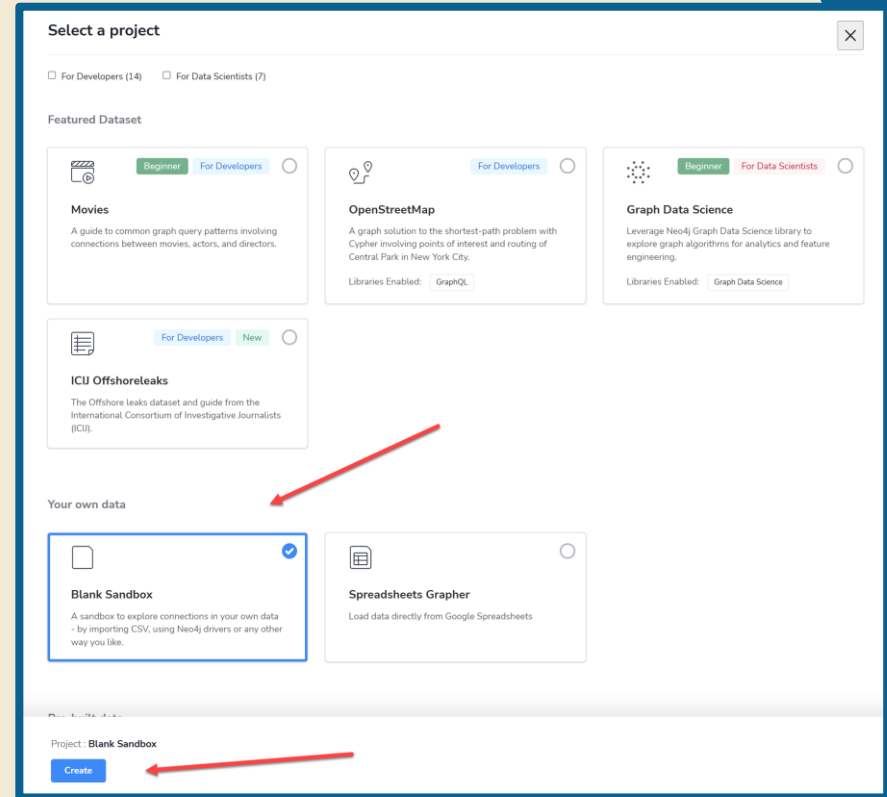
# Before we go further – Sandbox Login

- Create a sandbox for later
- <https://sandbox.neo4j.com/>



# Before we go further – Blank Sandbox

- <https://sandbox.neo4j.com/>
- Create a 'Blank Sandbox'
- If you're using *Desktop*
  - Install APOC
  - Install GDS



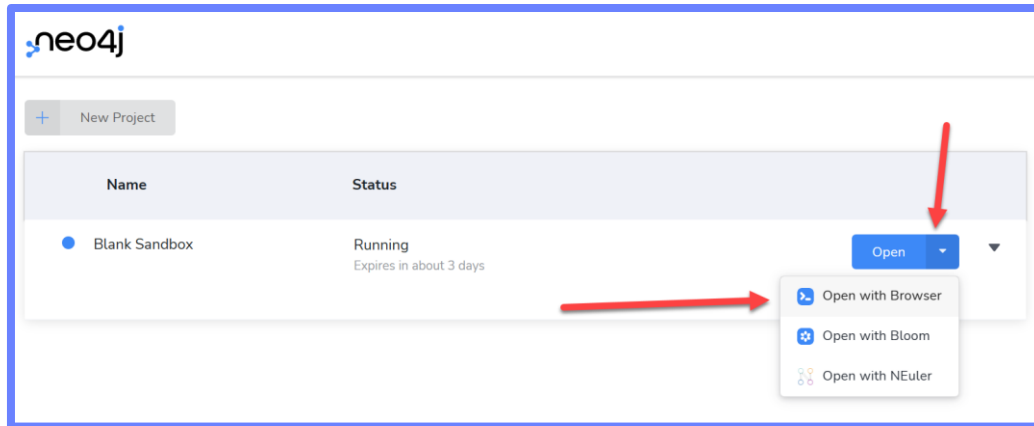


# Workshop



# Examples

- Go to the Neo4j Sandbox you created earlier (or Desktop!)
  - <https://sandbox.neo4j.com/>
- 'Open with Browser'



# Get the code

- Open the Github page:
- <https://github.com/cskardon/gsummit2023>





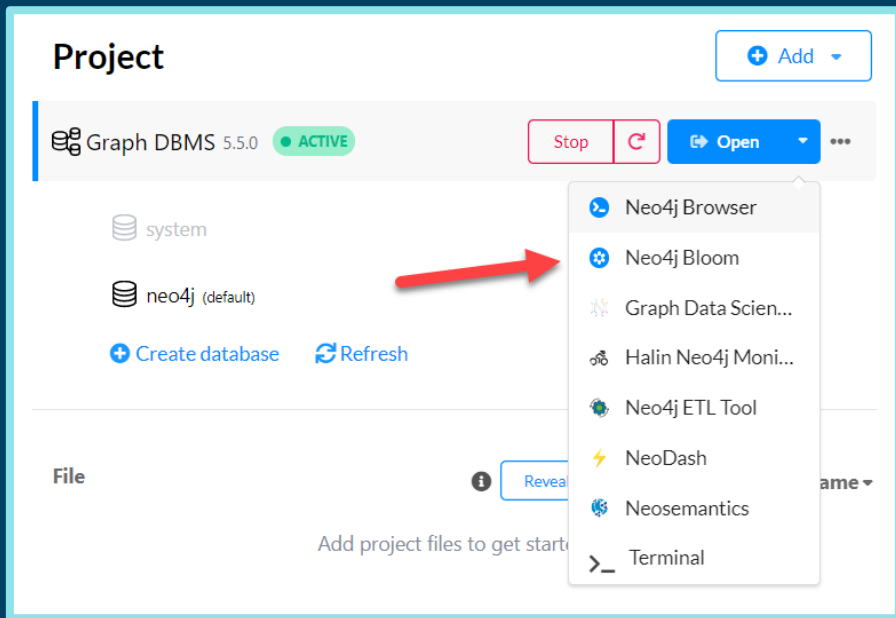
# Visualisation - Bloom





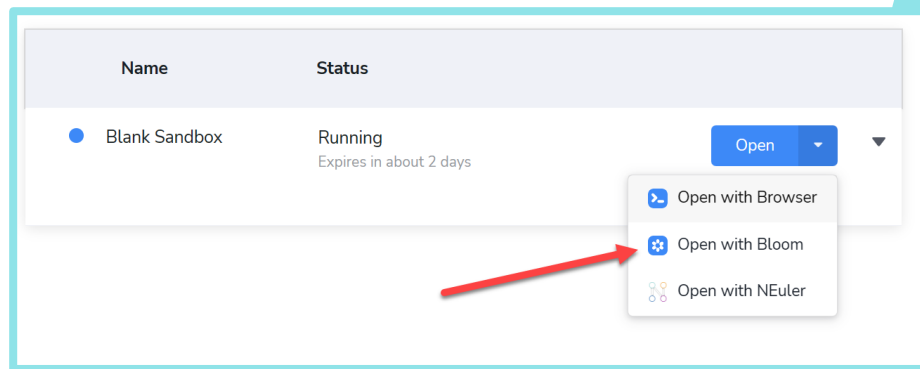
# Bloom - Desktop

- Open Bloom

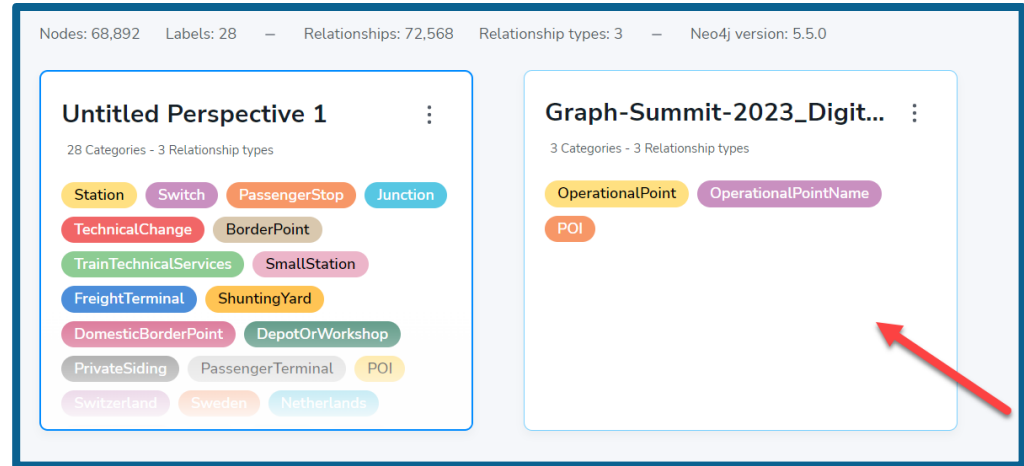
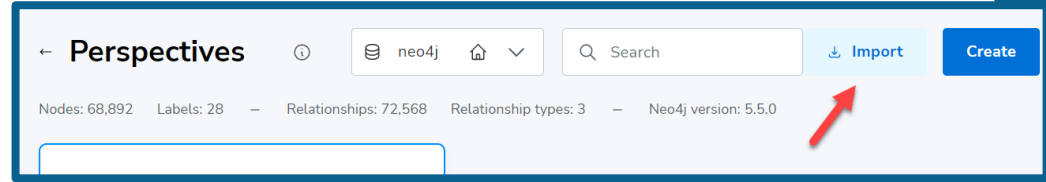
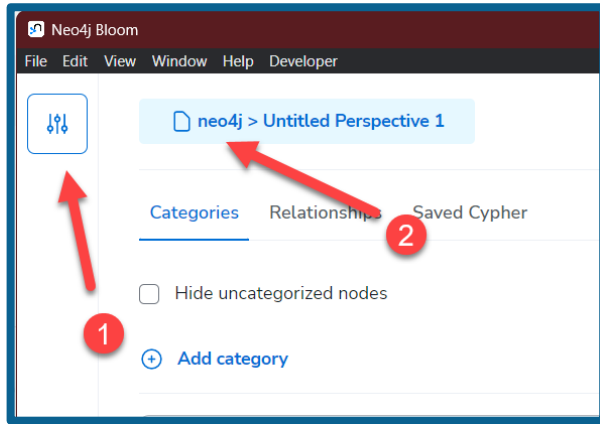


# Bloom - Sandbox

- Open Bloom



# Bloom – Import Perspective





# Visualisation - NeoDash



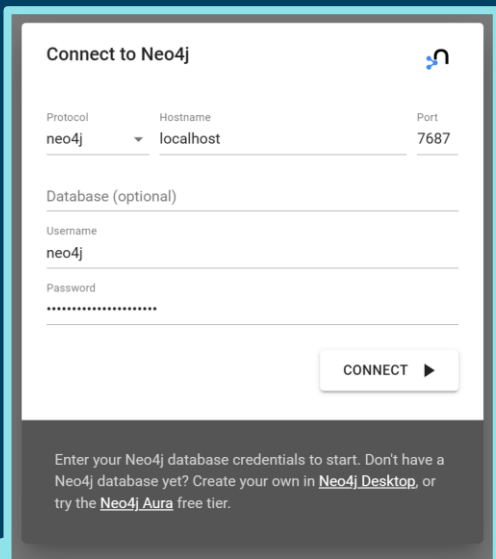
# NeoDash

- Open NeoDash:
  - <http://neodash.graphapp.io/>



# NeoDash - Desktop

- Host: localhost
- User: neo4j
- Password: YOURS!



The image shows the 'Connect to Neo4j' form in the NeoDash Desktop application. It includes fields for Protocol (neo4j), Hostname (localhost), Port (7687), Database (optional), Username (neo4j), and Password (masked with dots). A 'CONNECT' button is at the bottom right. A footer note mentions creating a Neo4j database or trying the Neo4j Aura free tier.

**Connect to Neo4j**

Protocol: neo4j Hostname: localhost Port: 7687

Database (optional)

Username: neo4j

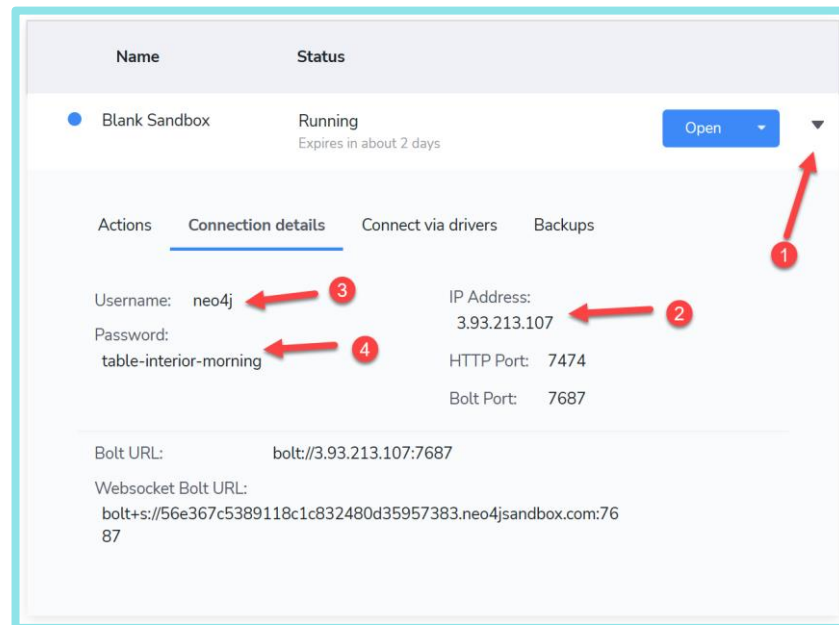
Password: .....

**CONNECT**

Enter your Neo4j database credentials to start. Don't have a Neo4j database yet? Create your own in [Neo4j Desktop](#), or try the [Neo4j Aura](#) free tier.

# NeoDash - Sandbox

- Open Sandbox
- Get connection details



The image shows the NeoDash Sandbox interface. It displays a table with columns 'Name' and 'Status'. The first row is 'Blank Sandbox' with status 'Running' and a note 'Expires in about 2 days'. An 'Open' button is next to it. Below the table, there are tabs for 'Actions', 'Connection details', 'Connect via drivers', and 'Backups'. The 'Connection details' tab is active, showing fields for Username (neo4j), Password (table-interior-morning), IP Address (3.93.213.107), HTTP Port (7474), Bolt Port (7687), Bolt URL (bolt://3.93.213.107:7687), and Websocket Bolt URL (bolt+ws://56e367c5389118c1c832480d35957383.neo4jsandbox.com:7687). Red arrows and numbers 1 through 4 highlight specific elements: 1 points to the 'Open' button, 2 points to the IP Address, 3 points to the Username, and 4 points to the Password.

Name	Status
Blank Sandbox	Running Expires in about 2 days

**Open**

Actions | **Connection details** | Connect via drivers | Backups

Username: neo4j IP Address: 3.93.213.107

Password: table-interior-morning HTTP Port: 7474

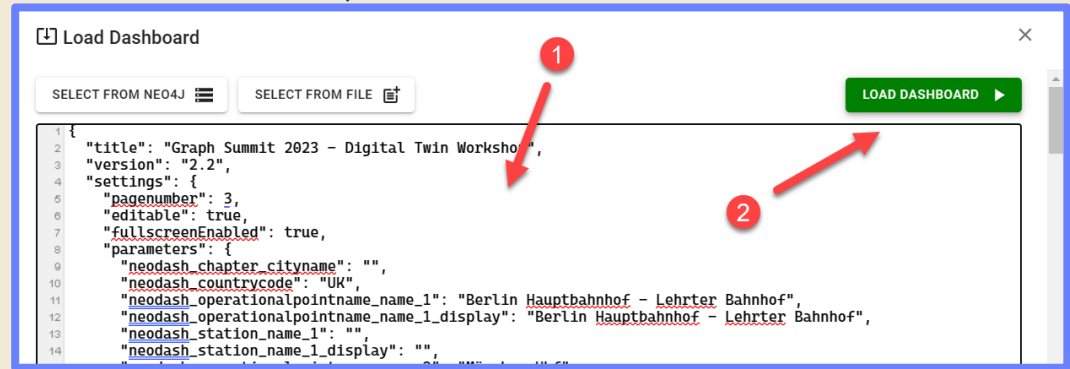
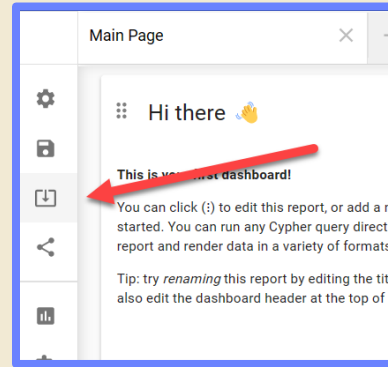
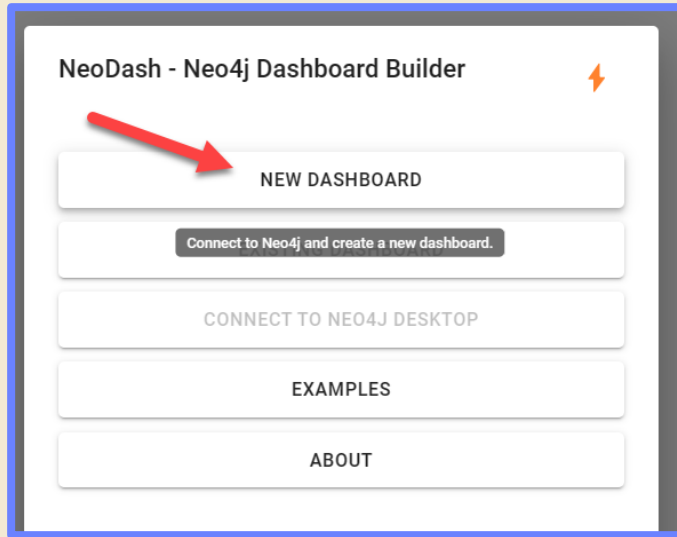
Bolt Port: 7687

Bolt URL: bolt://3.93.213.107:7687

Websocket Bolt URL: bolt+ws://56e367c5389118c1c832480d35957383.neo4jsandbox.com:7687

# NeoDash – Import Dashboard

- Create New Dashboard
- Import from Source files





Q & A



# Thank you

Contact us at  
[sales@neo4j.com](mailto:sales@neo4j.com)