Graph Databases with Python

Mark Henderson

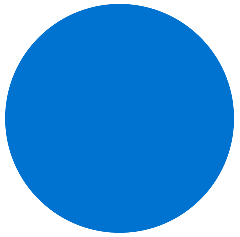
Who am I?

Who am I?

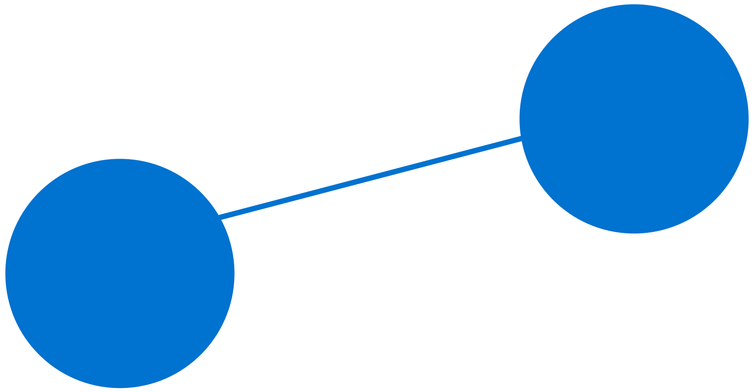
```
{
  name: Mark Henderson Sr.,
  occupation: Programmer/Manager @ DST,
  email: emehrkay@gmail.com,
  website: emehrkay.com,
  twitter: emehrkay,
  github: emehrkay,
  skype: emehrkay,
  aim: emehrkay2
}
```

How we think about relationships

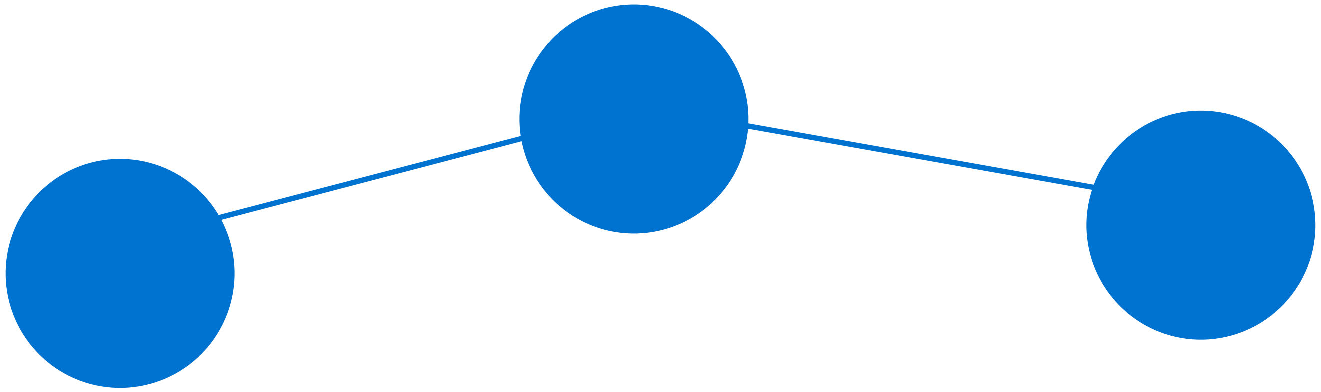
How we think about relationships



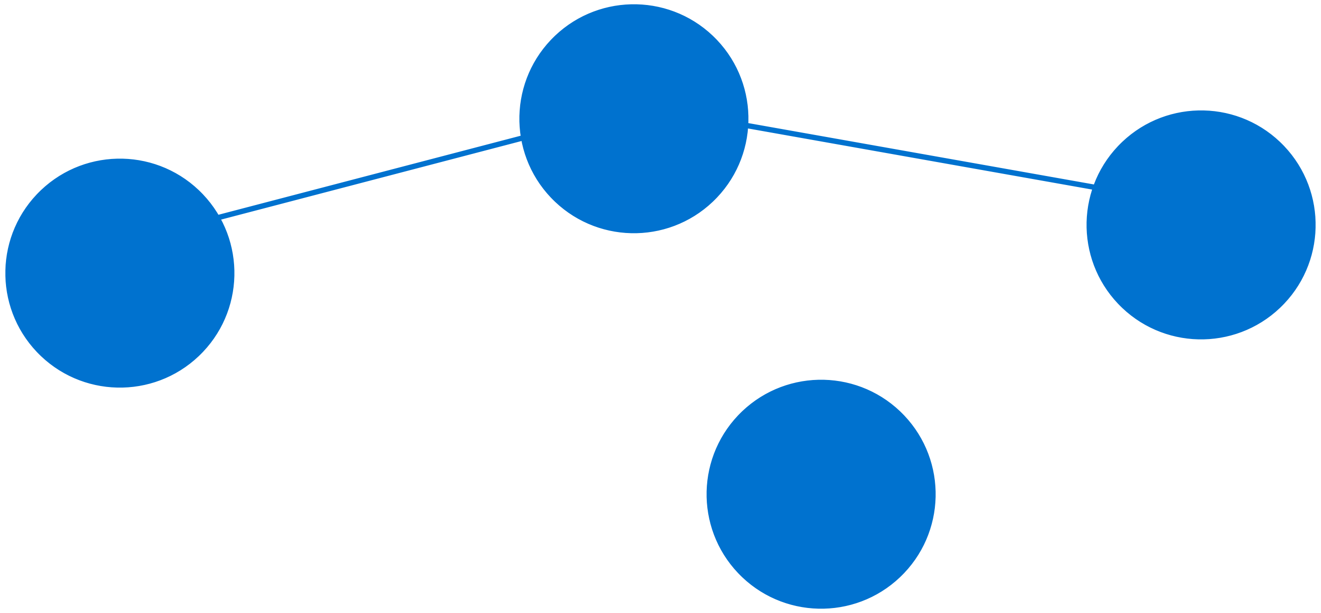
How we think about relationships



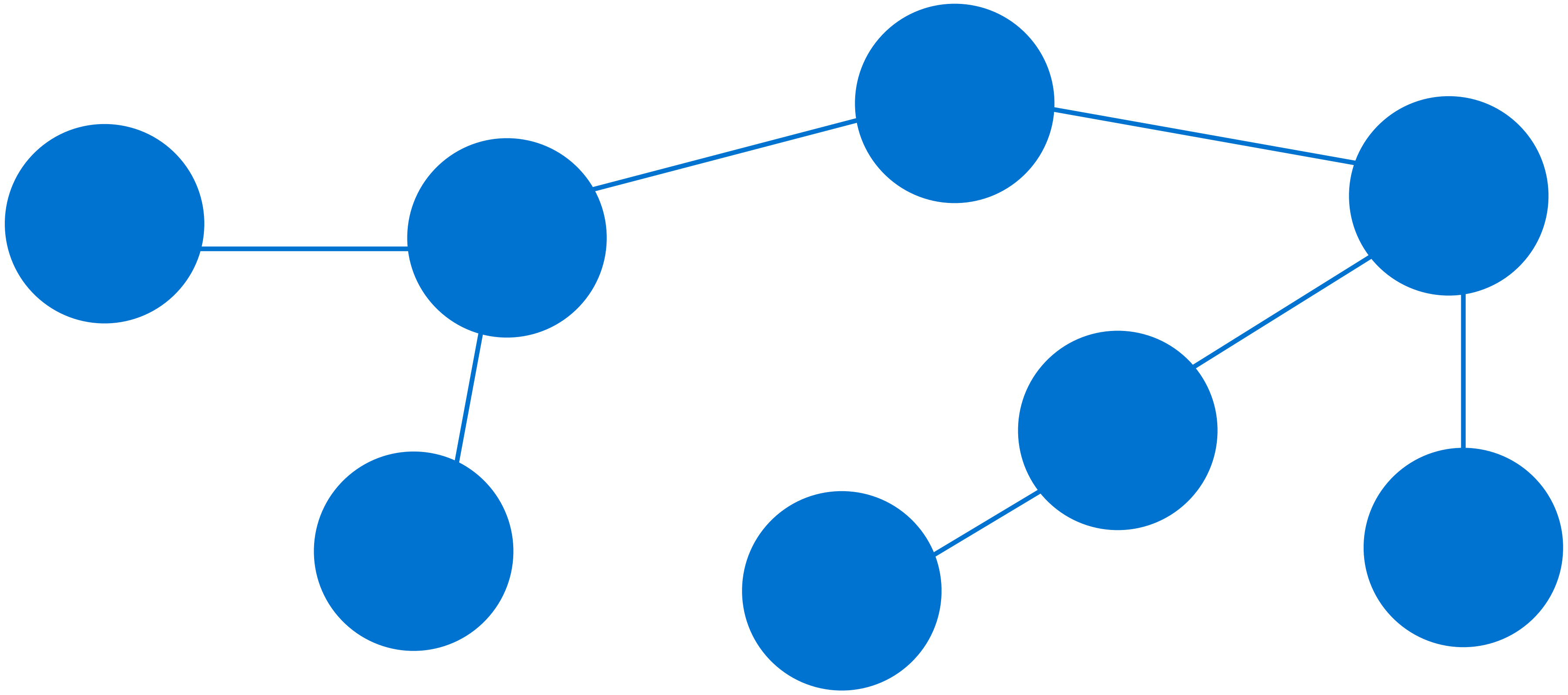
How we think about relationships



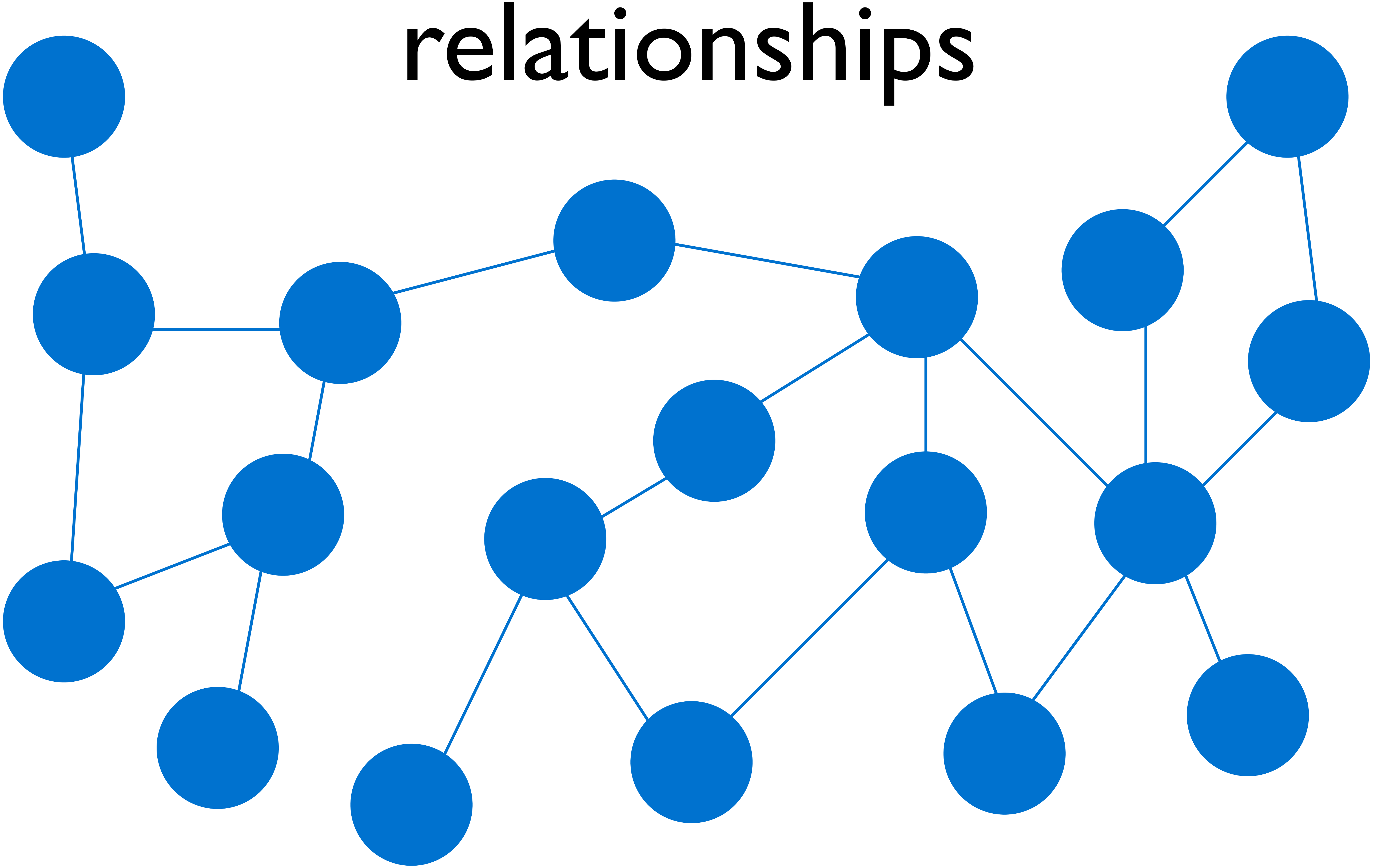
How we think about relationships



How we think about relationships



How we think about relationships



What is a Graph?

What is a Graph?

Wikipedia:

...a **graph** is a representation of a set of objects where some pairs of the objects are connected by links.

What is a Graph?

Wikipedia:

...a **graph** is a representation of a set of objects where some pairs of the objects are connected by links.

Node:

an object/point of interest in the graph

What is a Graph?

Wikipedia:

...a **graph** is a representation of a set of objects where some pairs of the objects are connected by links.

Node:

an object/point of interest in the graph

Edge:

connections between nodes

Types of graphs

Types of graphs

Directed:

relationships have different meanings based on start → end points

Types of graphs

Directed:

relationships have different meanings based on start → end points

Undirected:

relationships between nodes have no orientation

Types of graphs

Directed:

relationships have different meanings based on start → end points

Undirected:

relationships between nodes have no orientation

Mixed:

relationships can be directed or undirected

Types of graphs

Directed:

relationships have different meanings based on start → end points

Undirected:

relationships between nodes have no orientation

Mixed:

relationships can be directed or undirected

Multigraph, Quiver, Weighted, Simple, Half/loose edge

Graph Databases

Graph Databases

Neo4j (neo4j.org):

Full property graph. Java-based.

Graph Databases

Neo4j (neo4j.org):

Full property graph. Java-based.

Orientdb (orientdb.org):

Full document graph. Java-based.

Graph Databases

Neo4j (neo4j.org):

Full property graph. Java-based.

Orientdb (orientdb.org):

Full document graph. Java-based.

Twitter's FlockDB (github.com/twitter/flockdb):

Shallow relationships. JVM-based.

Standardization

Standardization

TinkerPop (tinkerpop.com):

Tools for giving some sanity to working with graphs.

Standardization

TinkerPop (tinkerpop.com):

Tools for giving some sanity to working with graphs.

Blueprints:

Models graph data.

Standardization

TinkerPop (tinkerpop.com):

Tools for giving some sanity to working with graphs.

Blueprints:

Models graph data.

Gremlin:

D.S.L. for querying graph data.

Standardization

TinkerPop (tinkerpop.com):

Tools for giving some sanity to working with graphs.

Blueprints:

Models graph data.

Gremlin:

D.S.L. for querying graph data.

Rexster

REST front-end server.

Standardization

TinkerPop (tinkerpop.com):

Tools for giving some sanity to working with graphs.

Blueprints:

Models graph data.

Gremlin:

D.S.L. for querying graph data.

Rexster

REST front-end server.

Furnace, Pipes, Frames

Python O.R.M. Solutions

Python O.R.M. Solutions

neo4j.py

Native Python-Java bridge code

Python O.R.M. Solutions

neo4j.py

Native Python-Java bridge code

neo4jrestclient (github.com/versae/neo4j-rest-client)

neo4j.py drop-in replacement vis REST consumption

Python O.R.M. Solutions

neo4j.py

Native Python-Java bridge code

neo4jrestclient (github.com/versae/neo4j-rest-client)

neo4j.py drop-in replacement vis REST consumption

Compass (github.com/emehrkay/Compass)

OrientDB REST client

Python O.R.M. Solutions

neo4j.py

Native Python-Java bridge code

neo4jrestclient (github.com/versae/neo4j-rest-client)

neo4j.py drop-in replacement vis REST consumption

Compass (github.com/emehrkay/Compass)

OrientDB REST client

Bulbs (http://bulbflow.com)

Rexter or Neo4j REST client

Examples

Examples

Connecting to neo4j

```
gdb = GraphDatabase("http://localhost:7474/db/data/")
```

Examples

Creating a node

```
n = gdb.nodes.create(color=‘Red’, width=16, height=32)
```

Examples

Creating a connection

```
n.relationships.create(‘KNOWS’, node2, **kwargs)
```

Examples

Finding nodes

```
nodes = Q(‘color’, exact=’RED’)
```

Examples

Traversing the graph

```
nodes = n.traverse(types=[client.All.KNOWS])[:]
```

Examples

Live Example

With class participation

Questions/Comments

Everything can be found at Github

Github.com/emehrkay/meetup-graph