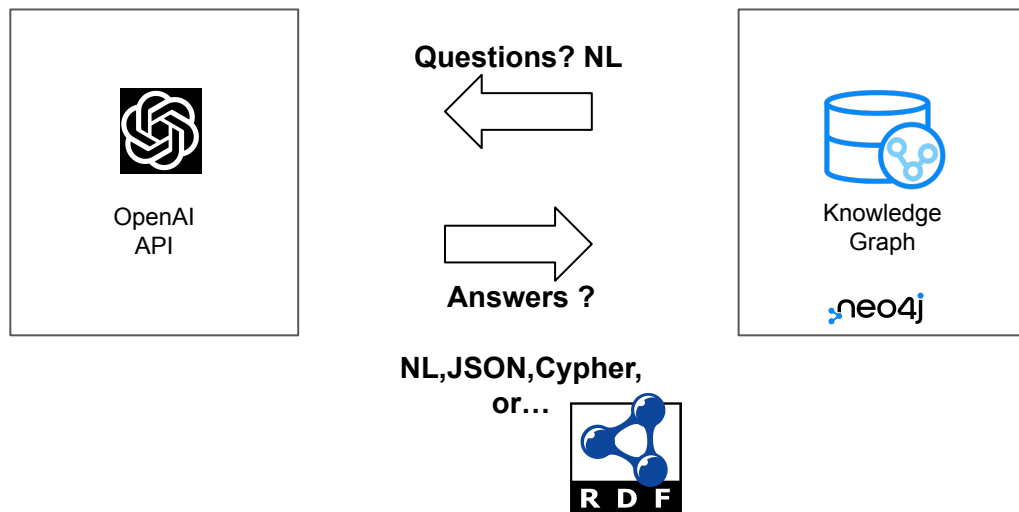


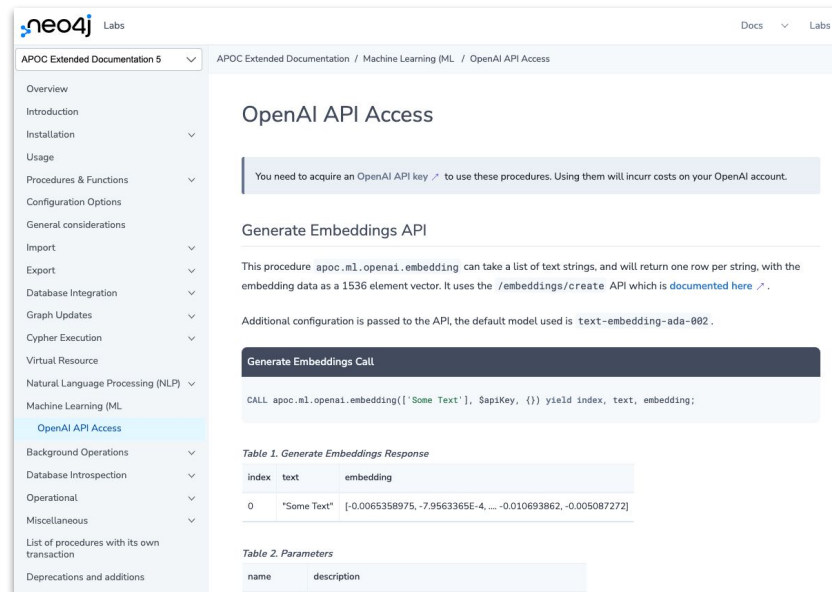
Going Meta #17: RDF-ing between OpenAI and Neo4j

A decorative network diagram in the bottom right corner of the slide. It consists of several blue circles of varying sizes connected by thin white lines, forming a complex web-like structure. The circles are distributed across the lower right quadrant, with some larger circles acting as hubs and many smaller ones as peripheral nodes.

LLM to speed up the construction of a KG



Integration in APOC (Extended)



The screenshot shows the Neo4j Labs documentation page for "OpenAI API Access". The left sidebar contains a navigation menu with categories like Overview, Introduction, Installation, Usage, Procedures & Functions, Configuration Options, General considerations, Import, Export, Database Integration, Graph Updates, Cypher Execution, Virtual Resource, Natural Language Processing (NLP), Machine Learning (ML), and OpenAI API Access (highlighted). The main content area is titled "OpenAI API Access" and includes a warning about acquiring an OpenAI API key. It also features a "Generate Embeddings API" section with a code block for a Cypher query and two tables: "Table 1. Generate Embeddings Response" and "Table 2. Parameters".

OpenAI API Access

You need to acquire an OpenAI API key to use these procedures. Using them will incur costs on your OpenAI account.

Generate Embeddings API

This procedure `apoc.ml.openai.embedding` can take a list of text strings, and will return one row per string, with the embedding data as a 1536 element vector. It uses the `/embeddings/create` API which is [documented here](#).

Additional configuration is passed to the API, the default model used is `text-embedding-ada-002`.

Generate Embeddings Call

```
CALL apoc.ml.openai.embedding(['Some Text'], $apiKey, {}) yield index, text, embedding;
```

Table 1. Generate Embeddings Response

index	text	embedding
0	"Some Text"	[-0.0065358975, -7.9563365E-4, ..., -0.010693862, -0.005087272]

Table 2. Parameters

name	description
------	-------------



```
call apoc.ml.openai.chat(...  
call n10s.rdf.import.inline(...
```

