Refactoring Large Graphs

- The batch refactoring workflow
  What are we going to do?
- Automate batch refactorings with the apoc library

Why do we need to batch?

Transaction State

Cypher keeps **all transaction state in memory** while running a query, which is fine most of the time.
When refactoring the graph, however, this state can get **very large** and may result in an **OutOfMemory** exception.

Adapt your heap size to match, or operate in batches.

```
dbms.memory.heap.initial_size=2G
dbms.memory.heap.max_size=2G
```

The batch refactoring workflow

The batch refactoring workflow

- tag all the nodes we need to process with a temporary label
  e.g. `Process`
- iterate over a subset of nodes flagged with that label (using `LIMIT`) and execute the refactoring
- remove the tag from the node
- return a count of how many rows were processed
- once the count reaches 0 then we've finished.

The batch refactoring workflow

```
MATCH (itemToProcess:Process)
WITH itemToProcess
LIMIT 1000
// do the refactoring
REMOVE itemToProcess:Process
WITH itemToProcess
RETURN count(*)
```

Start playing the next guide….

…if you aren’t playing it already


```
:play http://guides.neo4j.com/modeling_airports/05_refactoring_large_graphs.html
```

or

```
:play http://guides.neo4j.com/modeling_sandbox/05_refactoring_large_graphs.html
```

End of Module
Refactoring Large Graphs

Questions ?