**Codes**

* Query for returning the shortest path between “Multigold Production Limited” and “Bartfeld Trading Limited”

:param limit => 50;

:param config => {concurrency: 8, nodeQuery: null, relationshipQuery: null, weightProperty: "weight", defaultValue: 1, writeProperty: "louvain"};

:param startNodeId => null;

:param endNodeId => null;

:param startNode => "MULTIGOLD PRODUCTION LIMITED";

:param endNode => "BARTFELD TRADING LIMITED";

CALL db.propertyKeys() YIELD propertyKey MATCH (start) WHERE start[propertyKey] contains $startNode

WITH start

LIMIT 1

CALL db.propertyKeys() YIELD propertyKey MATCH (end) WHERE end[propertyKey] contains $endNode

WITH start, end

LIMIT 1

CALL algo.shortestPath.stream(start, end, $config.weightProperty, $config)

YIELD nodeId, cost

RETURN algo.getNodeById(nodeId) AS node, cost

* Query for returning the shortest path between “Carol Jean Bartfeld” and “Ostryn Business Limited”

:param limit => 50;

:param config => {concurrency: 8, nodeQuery: null, relationshipQuery: null, weightProperty: "weight", defaultValue: 1, writeProperty: "louvain"};

:param startNodeId => null;

:param endNodeId => null;

:param startNode => "CAROL JEAN BARTFELD";

:param endNode => "OSTRYN BUSINESS LIMITED";

CALL db.propertyKeys() YIELD propertyKey MATCH (start) WHERE start[propertyKey] contains $startNode

WITH start

LIMIT 1

CALL db.propertyKeys() YIELD propertyKey MATCH (end) WHERE end[propertyKey] contains $endNode

WITH start, end

LIMIT 1

CALL algo.shortestPath.stream(start, end, $config.weightProperty, $config)

YIELD nodeId, cost

RETURN algo.getNodeById(nodeId) AS node, cost