

Spring Neo4j

Spring Neo4j Component

Available as of Camel Extra 2.11

The **camel-spring-neo4j** library is provided by the [Camel Extra](#) project which hosts *GPL related components for Camel.

The **neo4j**: component allows you to treat Neo4j as a camel producer endpoint. This means you can use this component in to() calls but not from() calls. This component is backed by the [Spring Data Neo4j Library](#).

- As a producer, can create or remove nodes, and create or remove relationships.
- Can support as many endpoints as required, eg for multiple databases across multiple instances.
- Headers set for node id (for created nodes), relationship id (for created relationships)

Maven users will need to add the following dependency to their `pom.xml` for this component:

```
<dependency>
  <groupId>org.apache.camel</groupId>
  <artifactId>camel-spring-neo4j</artifactId>
  <version>x.x.x</version>
  <!-- use the same version as your Camel core version -->
</dependency>
```

URI format

```
spring-neo4j:http://hostname[:port]/database?options
```

Where the URL is the location of the of running neo4j rest server.

Headers

The following headers are set on exchanges during message transport.

| Property | Value |
|---------------------|---|
| Neo4jOperation | One of the Neo4jOperation enum values `CREATE_NODE, REMOVE_NODE, CREATE_RELATIONSHIP, REMOVE_RELATIONSHIP` which determines which action to perform |
| Neo4jNodeId | the id of the created node |
| Neo4jRelationshipId | the id of the created relationship |

The producer will set the headers for downstream processors once the operation has taken place. Any ID headers set prior to the producer are ignored.

Operations

The neo4j component looks for the Neo4jOperation header to determine what kind of entity to create, which is one of the following enum types

`CREATE_NODE, REMOVE_NODE, CREATE_RELATIONSHIP, REMOVE_RELATIONSHIP`

The body of the message is used to determine the node or relationship to manipulate. The following body types are supported:

For CREATE_NODE:

- null body - create default node
- Map body - create node with the properties set from the map

For REMOVE_NODE:

- Long or Integer - remove node using the body as the id
- neo4j Node instance - remove the node specified by that instance

For CREATE_RELATIONSHIP:

- SpringDataRelationship - create relationship specified by any @NodeEntity annotated Spring entities.
- org.apache.camel.component.neo4j.BasicRelationship - create relationship specified by the neo4j node types

For REMOVE_RELATIONSHIP:

- Long or Integer - remove relationship using the body as the id
- SpringDataRelationship - remove relationship specified by the @NodeEntity annotated Spring entities.
- org.apache.camel.component.neo4j.BasicRelationship - remove relationship specified by the neo4j node types

Samples

If you wanted to insert a new empty node every 1 seconds

```
from("timer://foo?period=1000").to("spring-neo4j:http://localhost:7474/data")
```

If you wanted to delete a specific node specified by a filename, eg a file of 100 would delete node 100.

```
from("file:/var/data").process(new Processor() {  
    @Override  
    public void process(Exchange exchange) throws Exception {  
        exchange.getIn().setBody(exchange.getIn().getHeader("CamelFileName"));  
    }  
}).to("spring-neo4j:http://localhost:7474/data")
```