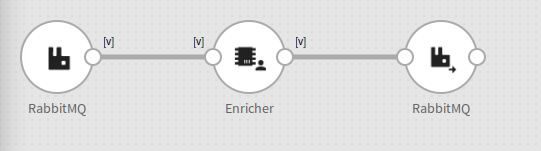
| **STREAMANALTIX 3.1.0** | Learn about **Couchbase Lookup component** |

# COUCHBASE LOOKUP

Couchbase Lookup takes input from the pipeline and creates a connection with Couchbase server through its custom implementation . It retrieves data on the basis of query and bucket information.

**Couchbase Configuration**



**Configure RabbitMQ Channel**

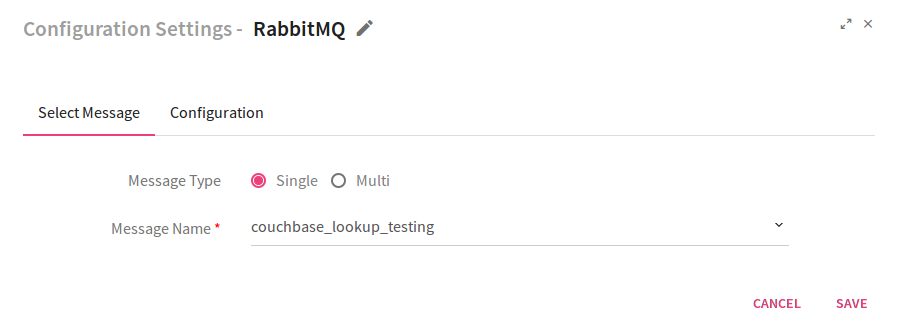
**Step 1:** Drag the RabbitMQ channel on the pipeline canvas and right click on it to configure.

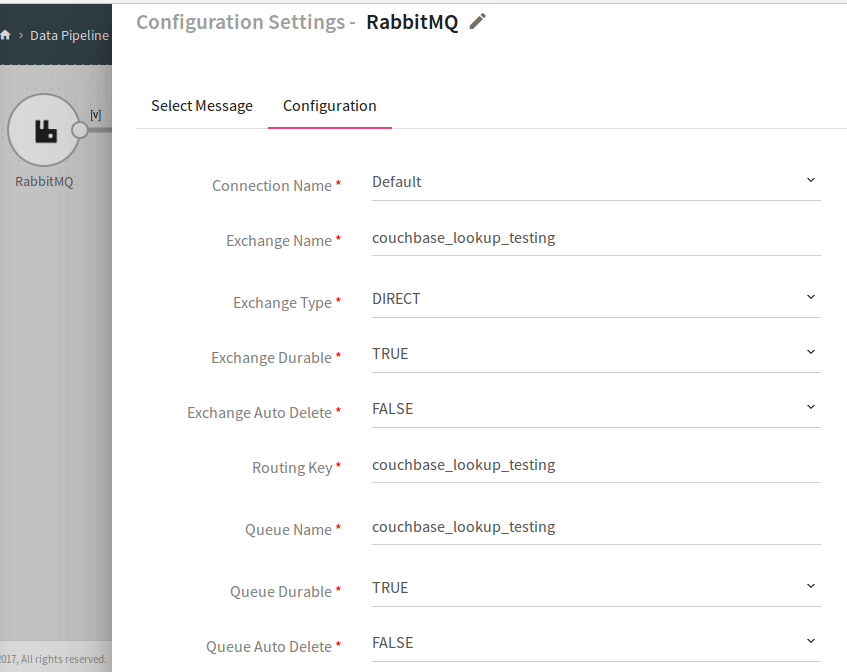
In Initially, message name is selected and then its queue and exchanges are defined. Using this queue only one can publish message which will trigger this pipeline

|  |  |
| --- | --- |
| **Field** | **Description** |
| **Message Type** | **Single:** If only one message needs to be parsed in the pipeline.  **Multi:** If multiple messages needs to be parsed in the pipeline. |
| [**Message Name**](#_References) | Click on the Message Name input field to get a list of all the messages and select message(s). |
| [**Schema Identifier**](#_References) | In case, you have written your own code for schema identification, check the box to provide fully qualified name of your class. |

**Step 2:** Configure RabbitMQ channel properties on the **Configuration** tab.

.





|  |  |
| --- | --- |
| **Field** | **Description** |
| **Connection Name** | Select a connection for the message broker. |
| **Exchange Name** | Exchange Name where data is pushed. |
| **Exchange Type** | Type of exchange where data is pushed. |
| **Exchange Durable** | If RabbitMQ is selected as broker, and if this property is set to True then the exchange will not be deleted if you restart RabbitMQ . |
| **Routing Key** | Routing key helps in routing specific data to specific queue based on binding using this key. |
| **Queue Name** | Name of the queue where data is stored. |
| **Queue Durable** | If this property is set to True then the exchange will not be deleted if you restart RabbitMQ |

## How to use Couchbase Lookup Function

Couchbase Lookup is an Enricher functionality in which data will be processed in order to enrich it with desired properties.

**Function Name**: **lookUpCouchbase**

**Parameters:**  lookUpCouchbase function takes two parameters: **bucket\_name** and **query**

You must specify these two parameters for retrieving the information from Couchbase server.

Given below is an example of a query wherein you will search for the record **first Name** “**Mike”** in the bucket **travel sample.**

After data retrieval, record will be stored in **dataMap** Variable.

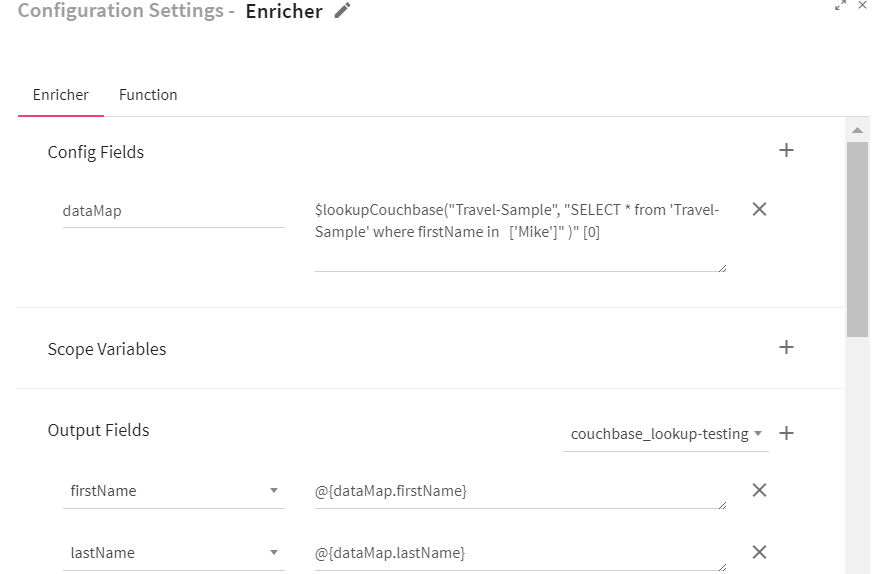
Now the Couchbase column values will be mapped with message fields.

**firstName**  field of message will be mapped with datamap.firstname column of Couchbase.

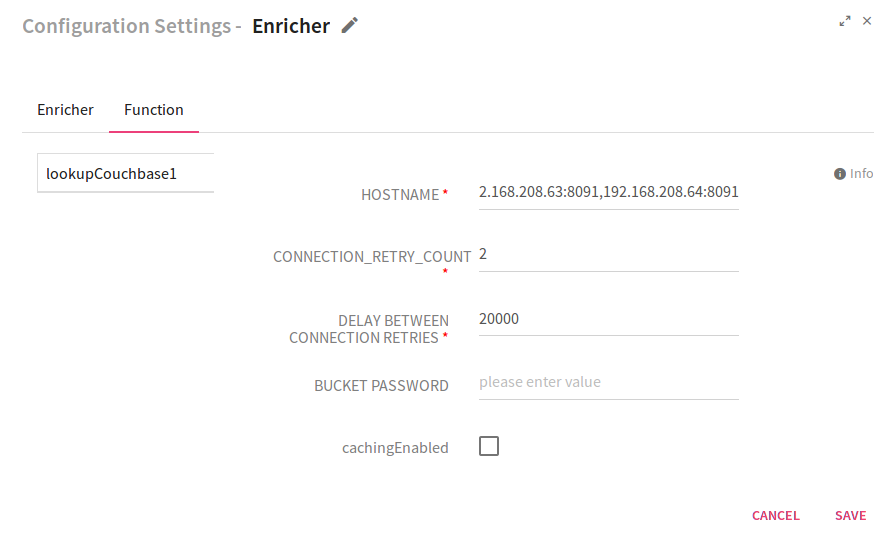
**LastName** field of message will be mapped with datamap.Lastname column of Couchbase.

**Parameters Description**:

|  |  |
| --- | --- |
| **BUCKET\_NAME** | Bucket name, from where data needs to be fetched from Couchbase cluster |
| **QUERY** | A query is a request for information from a database. |



Specify the connection details of Couchbase server on **Function** tab.

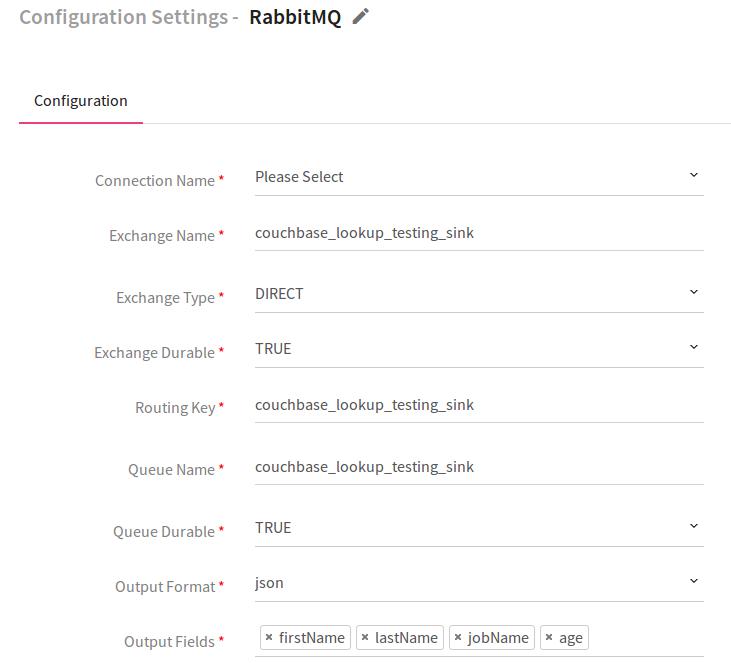


|  |  |
| --- | --- |
| **Field** | **Description** |
| **HOSTNAME** | Comma separated list of Couchbase cluster nodes(<IP>:<PORT>). |
| **CONNECTION\_RETRY\_COUNT** | In case Couchbase service is not available, it will try to connect for number of times entered in RETRY\_COUNT field. Default value of **CONNECTION\_RETRY\_COUNT** is 3. |
| **DELAY BETWEEN CONNECTION RETRIES** | Timeout used while making a connection with Couchbase. |
| **BUCKET PASSWORD** | It provides bucket level data protection. |
| **cachingEnabled** | Select this checkbox to enable caching. |

**Note:** Fields marked with asterisk sign () are mandatory.

**Sink RabbitMQ**

Configure RabbitMQ as emitter to analyze the output of custom Couchbase lookup. Below is the configuration.



|  |  |
| --- | --- |
| **Field** | **Description** |
| **Connection Name** | Select a connection for the message broker. |
| **Exchange Name** | Exchange Name where data is pushed. |
| **Exchange Type** | Type of exchange where data is pushed. |
| **Exchange Durable** | If RabbitMQ is selected as broker, and if this property is set to True then the exchange will not be deleted if you restart RabbitMQ . |
| **Routing Key** | Routing key helps in routing specific data to specific queue based on binding using this key. |
| **Queue Name** | Name of the queue where data is stored. |
| **Queue Durable** | If this property is set to True then the exchange will not be deleted if you restart RabbitMQ |
| **Output Format** | **Select JSON as output format.** |
| **Output Fields** | Select all fields which are being supplied as input and Hash Code. |

**Output:**



To give us your feedback on your experience with the application and report bugs or problems, mail us at [support@streamanalytix.com](mailto:support@streamanalytix.com) support@streamanalytix.com