Basic Introduction to Using the LoopBackRequest node

5

GREGH

Published on September 1, 2016 / Updated on September 19, 2016

Introduction

IBM Integration Bus v10.0.0.6 provides the LoopBackRequest node which is a new node that allows flow developers to easily create, retrieve, update and delete data records in external systems such as MongoDB, Cloudant and PostgreSQL.

The LoopBackRequest node uses the LoopBack framework to facilitate interaction with external systems through LoopBack connectors. LoopBack is an open source Node.js framework that can be extended by downloading and installing connectors from a large open source catalog using the ‘npm’ tool. The LoopBack framework also allows you to develop and use your own connectors.

This article will provide a simple example to introduce how to use the LoopBackRequest node to retrieve documents from MongoDB, a no-SQL database, and the setup that’s necessary to achieve that. The example will illustrate how to install the LoopBack MongoDB connector to make it available in the Integration Node runtime environment and the configuration that’s required to enable connection to a MongoDB database using the connector through the LoopBackRequest node.

MongoDB Data

For the purpose of this article the LoopBackRequest node will interact with a MongoDB database called ‘loopback’ that contains the following collection of SHAREPRICE documents held as JSON data:

```
{ “_id” : ObjectId(“576952b71181b9fc27121165”), “companyId” : “100”, “company” : “Barclays PLC”, “phone” : “0800 400100”, “price”: 171 }
{ “_id” : ObjectId(“576952b71181b9fc27121166”), “companyId” : “200”, “company” : “HSBC Holdings PLC”, “phone” : “0345 740 4404”, “price” : 430 }

{ “_id” : ObjectId(“576952b71181b9fc27121167”), “companyId” : “300”, “company” : “Prudential PLC”, “phone” : “0800 000 000”, “price” : 1298 }

{ “_id” : ObjectId(“576952b71181b9fc27121168”), “companyId” : “400”, “company” : “Worldpay Group PLC”, “phone” : “0808 253 0870”, “price” : 261 }

{ “_id” : ObjectId(“576952b71181b9fc27121169”), “companyId” : “500”, “company” : “Aviva Insurance PLC”, “phone” : “0800 142142”, “price” : 440 }

{ “_id” : ObjectId(“576952b71181b9fc2712116a”), “companyId” : “600”, “company” : “Eastspring Investments”, “phone” : “0800 000 000”, “price” : 325 }
```

The mongod service that manages the MongoDB database is hosted on the same machine that Integration Bus is running on and listens for client connections on the default port of 27017. The service is configured to authenticate client connections and a user lbadmin with password lbpassword has been granted full permissions on the ‘loopback’ database.

Configuring the LoopBack Connection to MongoDB

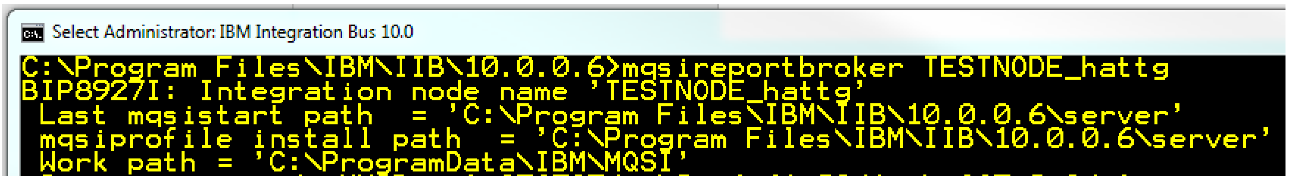
To connect to external systems like MongoDB, the LoopBackRequest node uses connection details that are provided in a file called datasources.json. This file must be located in the connectors/loopback subdirectory in the Integration Node’s workpath. The connectors/loopback directory will be created automatically when a new Integration Node is created using IIB v10.0.0.6. However, if you are using an Integration Node that was created at an earlier version, you will need to create these directories manually.

The datasources.json file can contain connection details for multiple backend systems which are separated into named stanzas. The LoopBackRequest node has a property called ‘Data source Name’ which must match the name of a stanza for the backend system to connect to. To connect to the ‘loopback’ MongoDB database I’ve created a datasources.json file with a stanza that I’ve called MONGO, as follows:

```
{
  “MONGO”: {
     “host”: “localhost”,
     “port”: 27017,
     “database”: “loopback”,
     “name”: “MONGO”,
     “connector”: “mongodb”
  }
}
```

This configuration uses the loopback-connector-mongodb connector provided by StrongLoop which is identified by the connector property (note, the LoopBack framework automatically adds the prefix ‘loopback-connector-’ to this name). Please refer to the documentation for further details on the configuration parameters that can be provided to the connector.

To determine the Integration Node’s workpath we will use the `mqsireportbroker` command.



Select Administrator: IBM Integration Bus 10.0

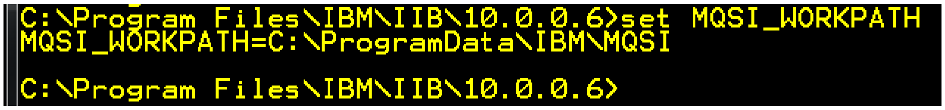
```
C:\Program Files\IBM\IIB\10.0.0.6>mqsireportbroker TESTNODE_hattg
BIP8927I: Integration node name 'TESTNODE_hattg'
 Last mqsistart path  = 'C:\Program Files\IBM\IIB\10.0.0.6\server'
 mqsiprofile install path  = 'C:\Program Files\IBM\IIB\10.0.0.6\server'
 Work path = 'C:\ProgramData\IBM\MQSI'
```

Using mqsireportbroker to determine the Integration Node’s workpath.

Having determined the Integration Node’s workpath we need to copy our datasources.json file into the connectors\loopback subdirectory within that workpath, in this case, C:\ProgramData\IBM\MQSI\connectors\loopback.
Note: It is possible to provide a datasources.json file that is specific to an Integration Node or Integration Server but, to keep things simple for the purposes of this article, the Integration Node will access these connection details from the default system wide workpath.

Installing the LoopBack MongoDB Connector

The LoopBack connectors referenced in the datasources.json file must be installed into the node_modules subdirectory within the directory referenced by the MQSI_WORKPATH environment variable. This directory isn’t necessarily the same as the workpath reported by the `mqsireportbroker` command as a specific workpath can be set for an Integration Node. However, in this case, we haven’t set a specific Integration Node workpath so the directories are the same.

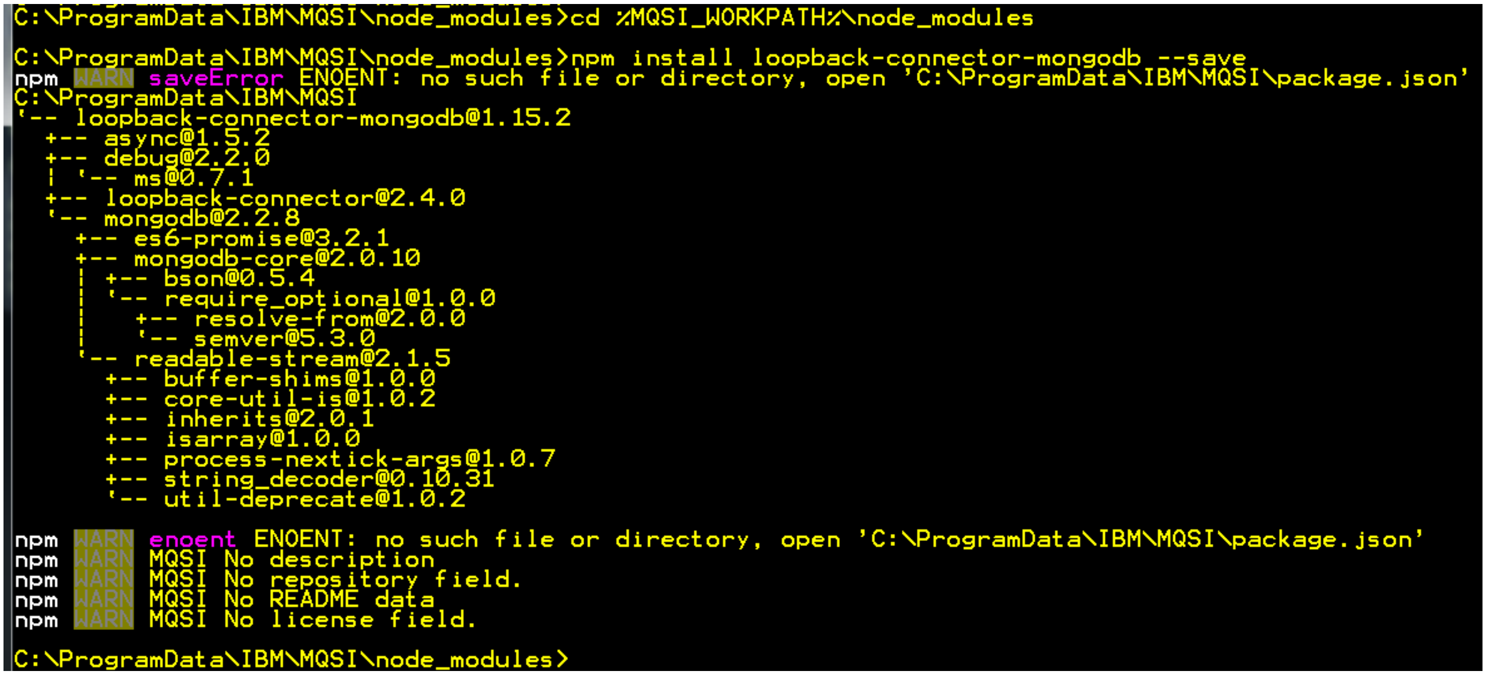


```
C:\Program Files\IBM\IIB\10.0.0.6>set MQSI_WORKPATH
MQSI_WORKPATH=C:\ProgramData\IBM\MQSI

C:\Program Files\IBM\IIB\10.0.0.6>
```

Checking the value of the MQSI_WORKPATH environment variable.

LoopBack connectors are installed using the Node JavaScript Package Manager `npm` command. It is important to use the version of the `npm` command that is packaged with Integration Bus to ensure compatibility with the NodeJS runtime environment that runs inside Integration Bus. To achieve this make sure `npm` is run from an Integration Bus command environment.
To install the MongoDB LoopBack connector change to the MQSI_WORKPATH\node_modules directory and then run:

```
npm install loopback-connector-mongodb –save
```

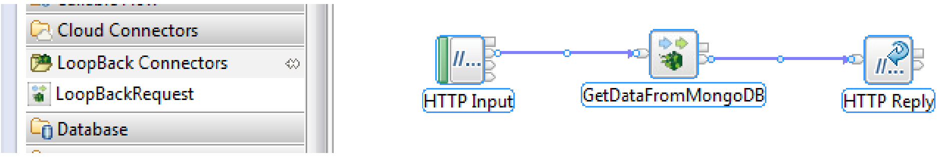


```
C:\ProgramData\IBM\MQSI\node_modules>cd %MQSI_WORKPATH%\node_modules

C:\ProgramData\IBM\MQSI\node_modules>npm install loopback-connector-mongodb --save
npm WARN saveError ENOENT: no such file or directory, open 'C:\ProgramData\IBM\MQSI\package.json'
C:\ProgramData\IBM\MQSI
`-- loopback-connector-mongodb@1.15.2
  +-- async@1.5.2
  +-- debug@2.2.0
  | `-- ms@0.7.1
  +-- loopback-connector@2.4.0
  `-- mongodb@2.2.8
    +-- es6-promise@3.2.1
    +-- mongodb-core@2.0.10
    | +-- bson@0.5.4
    | `-- require_optional@1.0.0
    |   +-- resolve-from@2.0.0
    |   `-- semver@5.3.0
    `-- readable-stream@2.1.5
      +-- buffer-shims@1.0.0
      +-- core-util-is@1.0.2
      +-- inherits@2.0.1
      +-- isarray@1.0.0
      +-- process-nextick-args@1.0.7
      +-- string_decoder@0.10.31
      `-- util-deprecate@1.0.2

npm WARN enoent ENOENT: no such file or directory, open 'C:\ProgramData\IBM\MQSI\package.json'
npm WARN MQSI No description
npm WARN MQSI No repository field.
npm WARN MQSI No README data
npm WARN MQSI No license field.

C:\ProgramData\IBM\MQSI\node_modules>
```

Installing a LoopBack connector with the 'npm' command

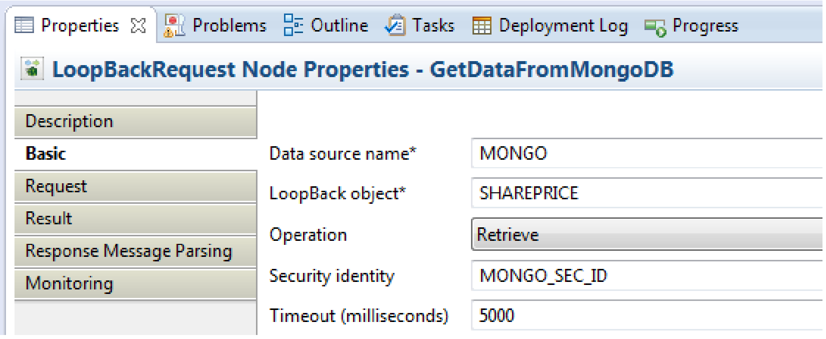
Using the LoopBackRequest Node in a Message Flow

The LoopBackRequest node can be found in the 'LoopBack Connectors' drawer of the node palette of the Integration Bus ToolKit.



Use of the LoopBackRequest node in a message flow

To test our earlier setup and configuration we will use a simple message flow, shown above, that uses the LoopBackRequest node to retrieve the SHAREPRICE documents that were described earlier from MongoDB. The 'GetDataFromMongoDB' LoopBackRequest node is configured with the following node properties:



LoopBackRequest Node Properties - GetDataFromMongoDB

Data source name* MONGO

LoopBack object* SHAREPRICE

Operation Retrieve

Security identity MONGO_SEC_ID

Timeout (milliseconds) 5000

Basic properties of the LoopBackRequest node

The 'Data source name' property is set to 'MONGO' which references the stanza of the same name in the datasources.json file that we configured earlier.

The 'LoopBack object' property is set to the name of the 'collection' that we want to retrieve documents from in our MongoDB database, in this case SHAREPRICE.

The 'Operation' is set to 'Retrieve' as we want to get the documents from MongoDB.

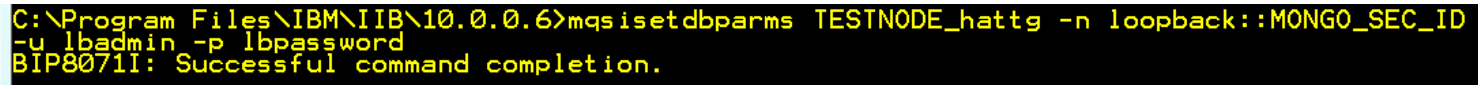
'Security identify' is set to name of a security alias, in this case MONGO_SEC_ID, that we will create using the `mqsisetdbparms` command that will allow us to pass a username and password to connect to MongoDB with. This approach is used in preference to including a username and password in the configuration provided in the datasources.json file as it avoids having security credentials displayed in plain-text.

‘Timeout’ controls how long the LoopBackRequest node will wait for a response to the request. For this example, we set a value of 5 seconds.

To keep things simple, we’ll leave the properties on the other tabs set with their default values.

Setting the Security Credentials for the MongoDB Connections

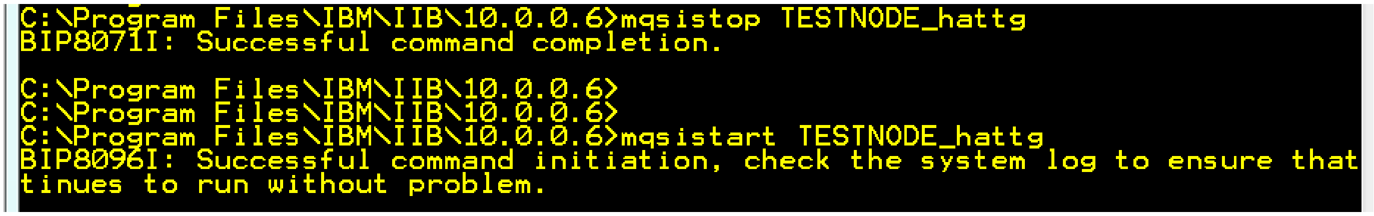
Before we can deploy the LoopBackRequest message flow we need to provide one last piece of configuration using the `mqsisetdbparms` command to specify the username and password with which to connect to MongoDB. The ‘Security identity’ property on the LoopBackRequest node was set to ‘MONGO_SEC_ID’ so, using the `mqsisetdbparms` command, we can associate a username and password of ‘lbadmin’ and ‘lbpassword’ respectively with that identity, as follows:



```
C:\Program Files\IBM\IIB\10.0.0.6>mqsisetdbparms TESTNODE_hattg -n loopback::MONGO_SEC_ID
-u lbadmin -p lbpassword
BIP8071I: Successful command completion.
```

Using mqsisetdbparms to set security credentials for a LoopBack datasource

The Integration Node must be restarted after running this command for this configuration to take effect.

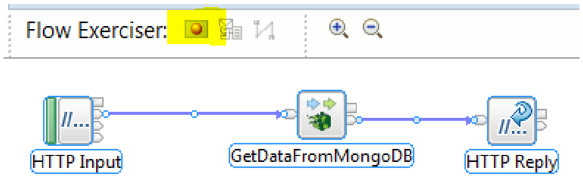


```
C:\Program Files\IBM\IIB\10.0.0.6>mqsistop TESTNODE_hattg
BIP8071I: Successful command completion.

C:\Program Files\IBM\IIB\10.0.0.6>
C:\Program Files\IBM\IIB\10.0.0.6>
C:\Program Files\IBM\IIB\10.0.0.6>mqsistart TESTNODE_hattg
BIP8096I: Successful command initiation, check the system log to ensure that
tinues to run without problem.
```

Restarting the Integration Node

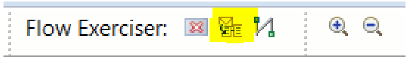
Deploying and Testing the LoopBackRequest Message Flow

We will deploy and test our LoopBackRequest message flow to an Integration Node using the ‘Flow Exerciser’. To do this click on the ‘Record Button’ and then select the Integration Node and Server in which you want to run the message flow.



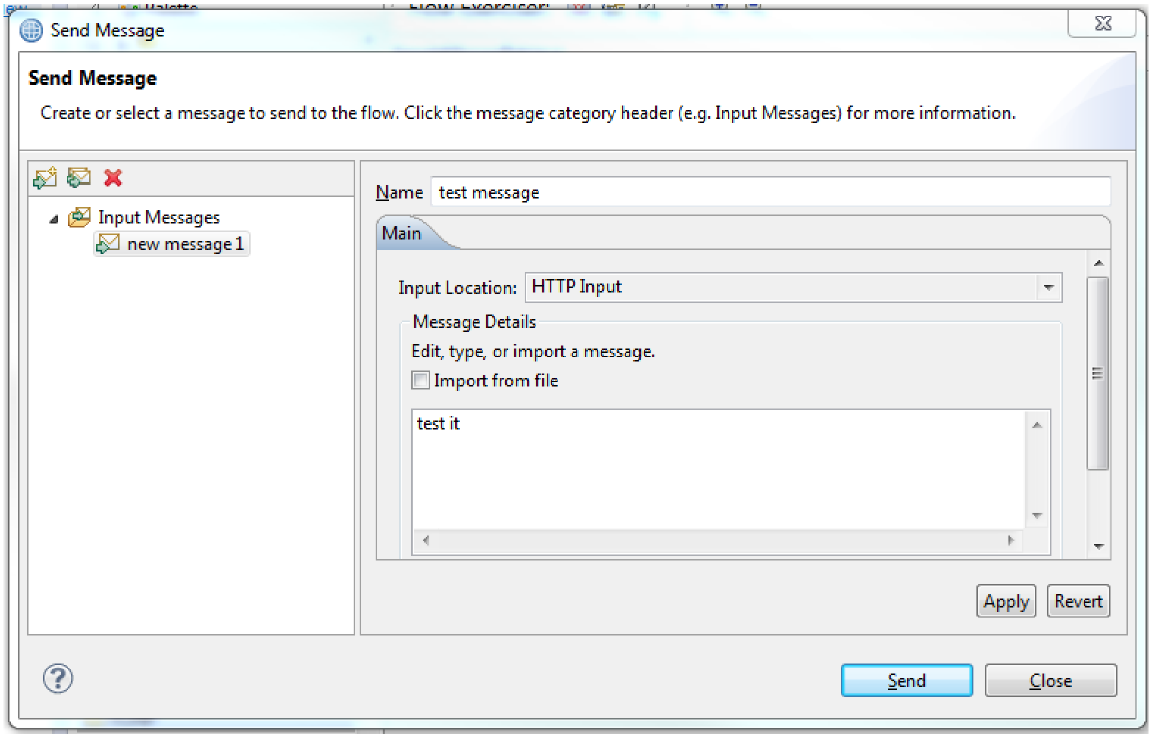
Testing the LoopBackRequest node with the Flow Exerciser

After the flow has deployed click on the ‘Send Message’ icon in the ‘Flow Exerciser’ to bring up the ‘Send Message’ dialog where a test message to send to the message flow can be entered.



Accessing the Send Message Dialog from the Flow Exerciser

Enter the message and then click on ‘Send’.



Send Message

Create or select a message to send to the flow. Click the message category header (e.g. Input Messages) for more information.

Name: test message

Input Location: HTTP Input

Message Details

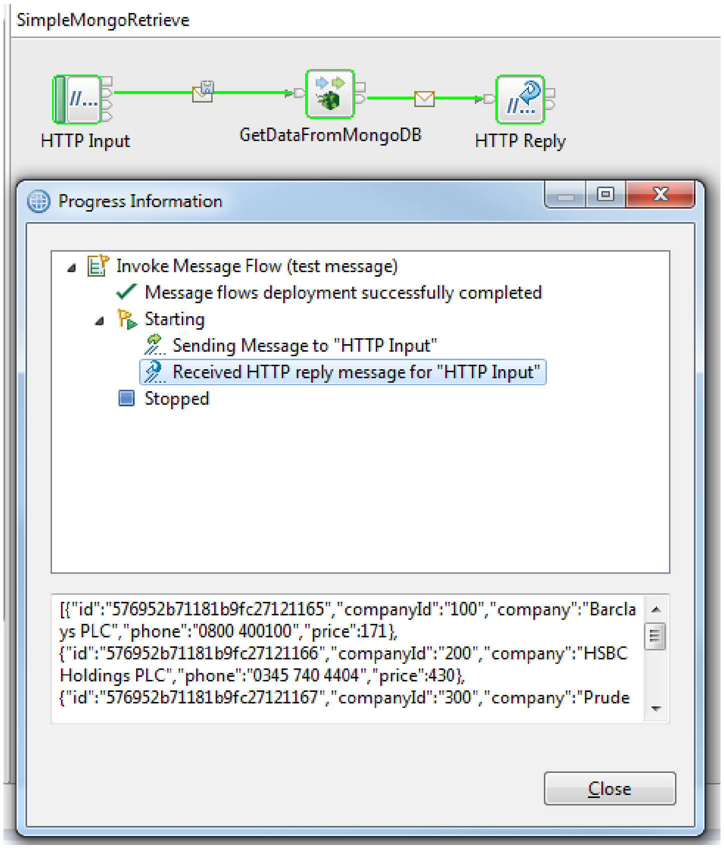
Edit, type, or import a message.

□ Import from file

test it

Sending a message to the LoopBackRequest message flow

The Flow Exerciser will show the path the message took through the Message Flow and the SHAREPRICE documents returned in response to the 'Retrieve' request sent by the LoopBackRequest node will be sent in the HTTP Reply.



```
Invoke Message Flow (test message)
    Message flows deployment successfully completed
    Starting
        Sending Message to "HTTP Input"
        Received HTTP reply message for "HTTP Input"
    Stopped
```

```
[{"id":"576952b71181b9fc27121165","companyId":"100","company":"Barclays PLC","phone":"0800 400100","price":171},
{"id":"576952b71181b9fc27121166","companyId":"200","company":"HSBC Holdings PLC","phone":"0345 740 4404","price":430},
{"id":"576952b71181b9fc27121167","companyId":"300","company":"Prude
```

Documents retrieved from MongoDB by the LoopBackRequest node

Summary

This article has provided an example of how to use the LoopBackRequest node to access documents from a MongoDB database. It has described the use of the `npm` tool to download and install a LoopBack MongoDB connector into the Integration Bus runtime environment and provided details of the configuration that's required in the datasources.json file to interact with a MongoDB database. It has also described the configuration of the basic properties that are available on the LoopBackRequest node to retrieve documents from the MongoDB database and detailed using the `mqsisetdbparms` command to provide a username and password for the connection.

TAGS INTEGRATION, INTEGRATION BUS, IIBV10.0.0.6, LOOPBACK

GREGH

5 comments on"Basic Introduction to Using the LoopBackRequest node"

Cobianu March 04, 2019

did u resolve that ?

Reply (Edit)

Ian_Larner March 05, 2019

@Cobianu Hi
The problem was resolved by using fixpack 10.0.0.6 or later. See the comments in the LoopBackRequest MongoDB_RESTAPI project interchange page.

Regards,
Ian

Reply (Edit)

Cobianu March 04, 2019

BIP3879E: The LoopBackRequest node received an error from LoopBack when attempting to connect to the data source name 'MONGO'. Detail: ' WARNING: LoopBack connector "mongodb" is not installed as any of the following modules: ./connectors/mongodb loopback-connector-mongodb To fix, run: npm install loopback-connector-mongodb '.

I did the steps required in this tutorial but after deploy it says the error above, the connector is installed at :
C:\ProgramData\IBM\MQSI\node_modules\loopback-connector-mongodb

Any idea why it doesnt work ?

Reply (Edit)

martin_b March 06, 2019

Hi,

It's not apparnent from this why it's not working, you could cross check teh step with the related article "Interacting with MongoDB using IBM Integration Bus LoopBackRequest node" see https://developer.ibm.com/integration/blog/2016/09/02/interacting-with-mongodb-using-ibm-integration-bus-loopbackrequest-node/

Also looking in dwAnswers, this post could be relevant: https://developer.ibm.com/answers/questions/479111/iib-v100011-loopback-connector-deploy-error-for-mo/

HTH

Reply (Edit)

debabrata April 11, 2017

getting an error on loopbackrequest node on message flow.
Message node "com.ibm.connector.loopback.ComIbmRequest" cannot be located

how to resolve this error?

Reply (Edit)