

Node.js Driver 1.6.1

SQream Technologies

2019-03-04 | Version 1.6.1

Table of Contents

1. Prerequisites	1
1.1. Install Node.js 10.x	1
2. Test the connection to Scream DB	2
2.1. Create a node.js file to configure your connection and database query	2
2.2. Run and test the Node.js file	3
3. Events	3
3.1. getConnectionId	3
3.2. getStatementId	3
3.3. getTypes	3
3.4. Event Examples	4
4. Set Statement Flag	4
5. Recommended Configuration	4
Copyright	5

This guide describes:

- The required prerequisites, and installation of the SQream Node.js driver.
- How to create and test a connection to SQream DB for a Node.js application.

1. Prerequisites

1.1. Install Node.js 10.x

Node.js is an open source server framework. It allows you to run JavaScript on the server. To check whether you already have **Node.js** installed, run:

```
node --version
```

If you already have Node.js 10.x installed, you can skip this step.

1.1.1. Centos 7.x

If your PC has internet access you can install it directly from there (Option 1). If your PC doesn't have internet access you have to first download it and then install it (Option 2).

Option 1 - with Internet access

If your PC has internet access perform the following:

```
curl -sL https://rpm.nodesource.com/setup_10.x | sudo bash -  
sudo yum install -y nodejs
```

Option 2 – no Internet access

Download the latest version of node.js and then copy it onto the machine you want to install the Dashboard.

For example: <http://nodejs.org/dist/v10.14.1/node-v10.14.1.tar.gz>

Install:

```
tar xzvf node-v10.14.1.tar.gz && cd node-v10.14.1  
./configure  
make  
sudo make install
```

Verify Node 10.x is installed

```
node --version
```

In case you still have a lower version of node installed make sure to completely uninstall and install node 10.x.

1.1.2. Ubuntu 16.04

Install Node 10.x

```
cd ~  
curl -sL https://deb.nodesource.com/setup_10.x -o nodesource_setup.sh  
sudo bash nodesource_setup.sh  
sudo apt-get install -y nodejs
```

Verify Node 10.x is installed

```
node --version
```

In case you still have a lower version of node installed make sure to completely uninstall and install node 10.x.

2019-03-04 | Version 3.1 == Install the SQream Node.js Driver

- Download the SQream Node.js driver file: SQream-nodejs-connector-[version].tgz.
- Install or upgrade the driver with npm install from file.

For example:

```
sudo npm install SQream-nodejs-connector-1.6.1.tgz
```

2. Test the connection to Sqream DB

2.1. Create a node.js file to configure your connection and database query

For example: **node_connector.js**. Make sure to edit the required details, such as <your server address>, <your server port>, <your username> and <your password>, etc.

Example:

```
const Connection = require('SQream-nodejs-connector');
const config = {
  host: '<your server address>',
  port: <your server port>,
  username: '<your username>',
  password: '<your password>',
  connectDatabase: '<your database>',
  cluster: '<true/false>',
  service: '<Workload manager service, optional, defaults to 'scream'>'
};
const query1 = "SELECT 1 as test, 2 as other_test";
const myConnection = new Connection(config);
myConnection.runQuery(query1, function (err, data){
  console.log(err, data);
});
```

2.2. Run and test the Node.js file

For example:

```
node node_connector.js
```

On a successful connection, you should see:

```
null [ { test: 1, other_test: 2 } ]
```

3. Events

The scream-connector has a few events return with event emitter.

3.1. getConnectionId

getConnectionId event return the executing connection Id.

3.2. getStatementId

getStatementId event return the executing statement Id

3.3. getTypes

getTypes event return the results columns types

3.4. Event Examples

```
const myConnection = new Connection(config);

myConnection.runQuery(query1, function (err, data){
  myConnection.events.on('getConnectionId', function(data){
    console.log('getConnectionId', data);
  });

  myConnection.events.on('getStatementId', function(data){
    console.log('getStatementId', data);
  });

  myConnection.events.on('getTypes', function(data){
    console.log('getTypes', data);
  });
});
```

4. Set Statement Flag

You can add a 'SET' flag to statements inside 'runQuery'.

```
const setFlag = 'SET showfullexceptioninfo = true;';

const myConnection = new Connection(config);
myConnection.runQuery(query1, function (err, data){
  console.log(err, data);
}, setFlag);
```

5. Recommended Configuration

The node application which utilizes the sstream connector should consider the heap size node configuration.

When processing large datasets, it is recommended to increase the application heap size with the "--max_old_space_size" node run flag:

```
node --max_old_space_size={heapSize} my-application.js
```



The following error message indicates that the heap size configured is not sufficient:

FATAL ERROR: CALL_AND_RETRY_LAST Allocation failed - JavaScript heap out of memory

Copyright

Copyright © 2010-2019. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchant- ability or fitness for a particular purpose.

We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document.

This document may not be reproduced in any form, for any purpose, without our prior written permission.