Node.js MSSQL tedius ConnectionError: Failed to connect to localhost:1433 - connect ECONNREFUSED

I am trying to connect to MSSQL 2012 using NodeJS with the mssql connection interface.

When attempting to connect I get the following error:

```
{ [ConnectionError: Failed to connect to localhost:1433 - connect ECONNREFUSED]
  name: 'ConnectionError',
  message: 'Failed to conncet to localhost:1433 - connect ECONNREFUSED',
  code: 'ESOCKET' }
```

Any ideas on how to fix this?

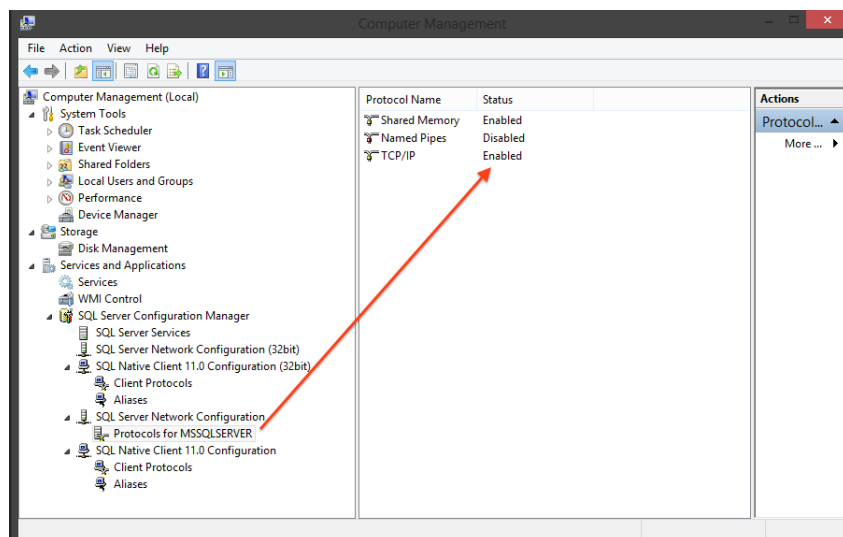sql-server   node.js   node-mssql   tedious

asked Aug 29 '14 at 22:50

Matt Carrier
1,550 ●2 ●14 ●25

4 Answers

The solution is to enable TCP connections which are disabled by default.
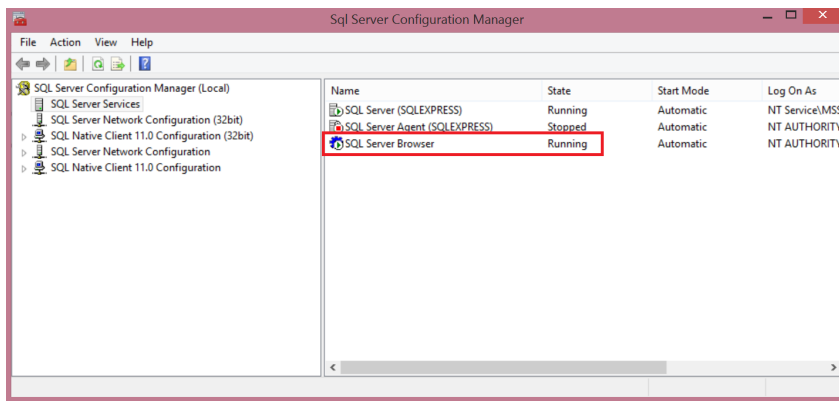

answered Aug 29 '14 at 22:50

Matt Carrier
1,550 ●2 ●14 ●25

1   Thank you! This solved my issue exactly. – Andy Poquette Dec 9 '14 at 14:08

1   I was looking for this for almost whole day! Thanks for saving time :) – Fka Jun 19 at 20:11

My case wasn't exactly the same as Matt's, but his screenshot was enough to remember me what was missing.

As it is said here, when you are using the SQL Server Instance Name to connect to it, you must have SQL Server Browser running.

> options.instanceName The instance name to connect to. The SQL Server Browser service must be running on the database server, and UDP port 1444 on the database server must be reachable. (no default) Mutually exclusive with options.port.

answered Jun 8 '15 at 13:08

MMalke
588 ● 1 ● 6 ● 22

Best practice is to first verify the connection to the SQL server using a query analyzer (SQL Management Studio (Windows) or SQLPro for MSSQL (Mac)) using the same protocol, port and credentials as you wish to use via your application.

In Management Studio, the format is Server,Port (e.g. 192.168.1.10,1433); and you'll probably be using SQL Server Authentication instead of Windows Authentication.

Steps to configure the SQL Server:

Install with Mixed Authentication, if you intend to use SQL Server Authentication.

Setup SQL Server to listen on TCP on a fixed port number:

- SQL Configuration Manager SQL Server Network Configuration
  - Protocols for {Instance}
    - TCP/IP - Enabled (double-click)
    - IP Address (on all desired interfaces)
      - TCP Dynamic Ports = BLANK! (not zero)
      - TCP Port - 1433 (or desired port)

answered Jun 22 at 1:33

Adam Gering
64 ● 5

I couldn't connect with 'localhost' although I use 'localhost' in SQL Management Studio and other applications. When I used Computer Name (network address), it worked!

answered Aug 5 at 14:25

DaNeSh
194 ● 11