

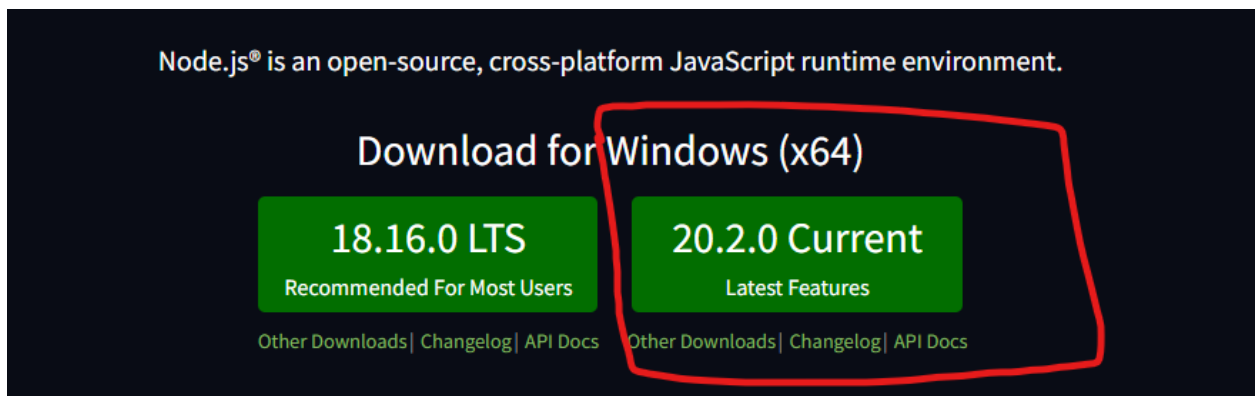
NODEJS SQL CONNECTIVITY

PREREQUISITES

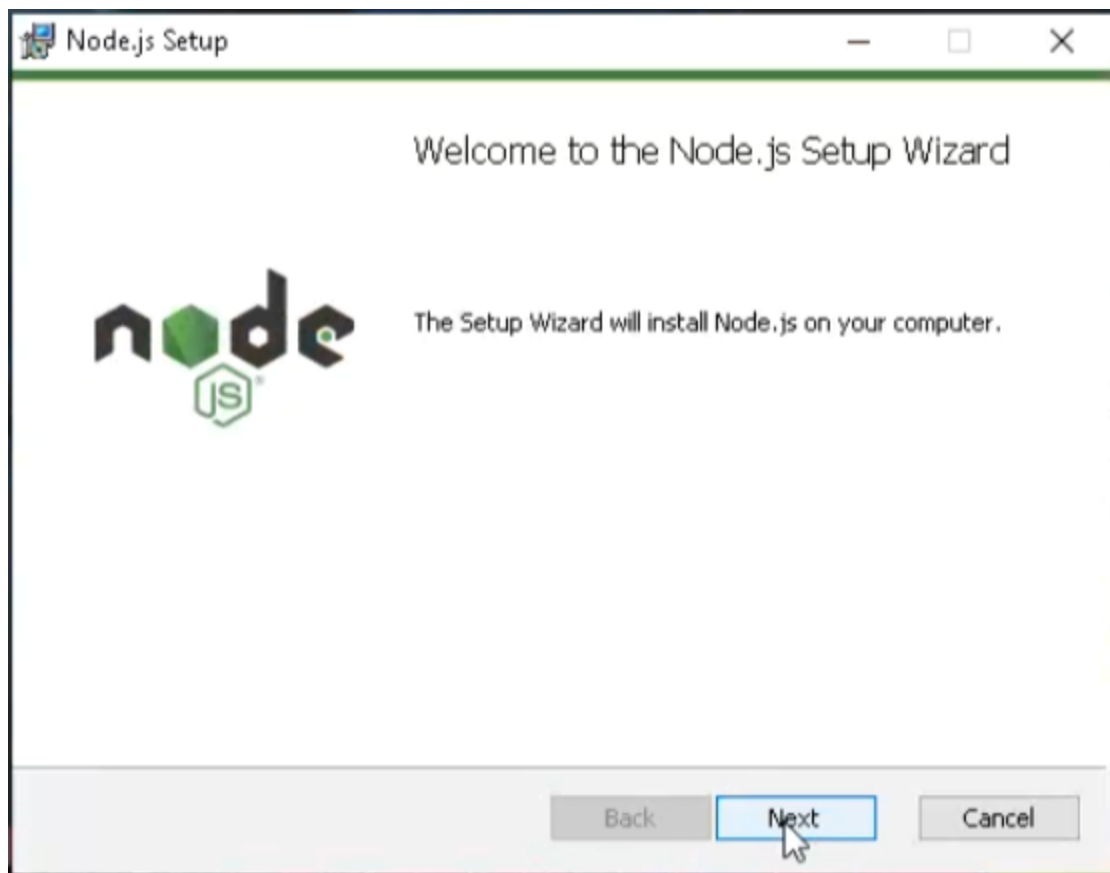
1. SQL Server
2. SQL Server Management Studio
3. Nodejs
4. An Editor

Step1: Download and install Nodejs from

<https://nodejs.org/en>



After downloading the file, run the setup



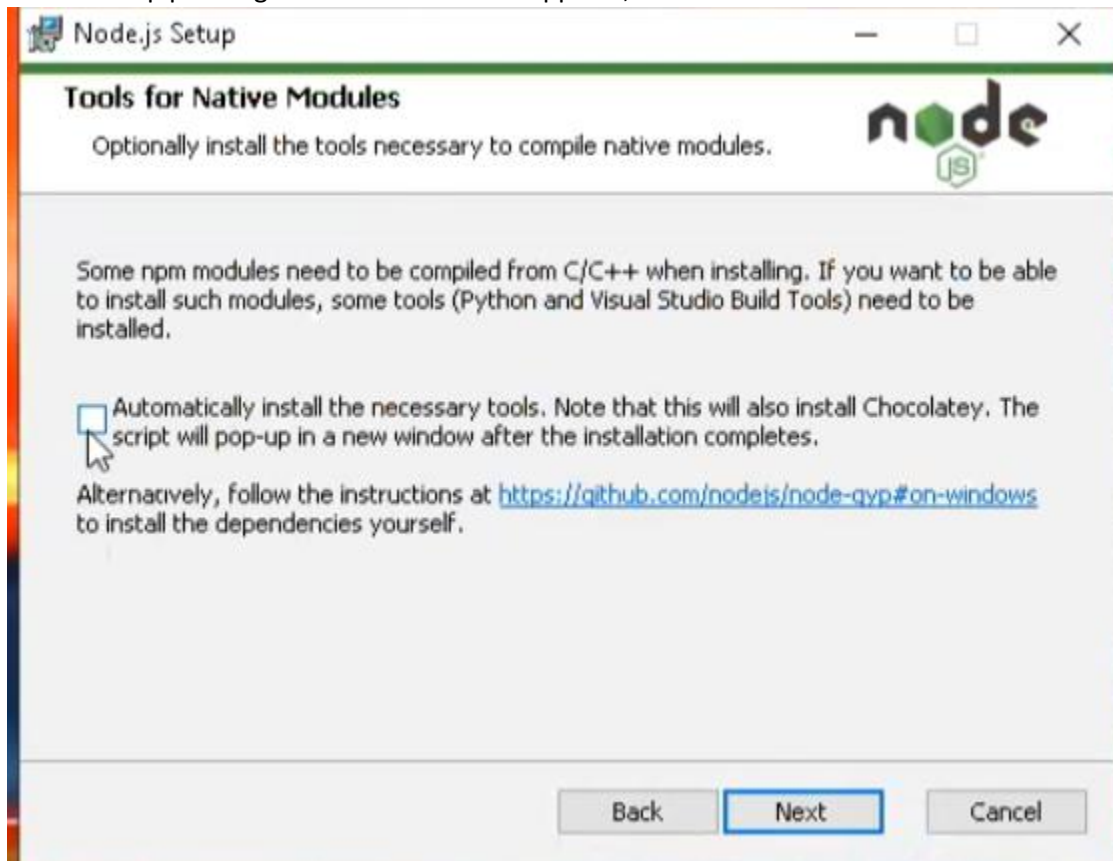
Click on next



Check the "I accept the terms " and click next.

Next again

Keep pressing next until this screen appears;



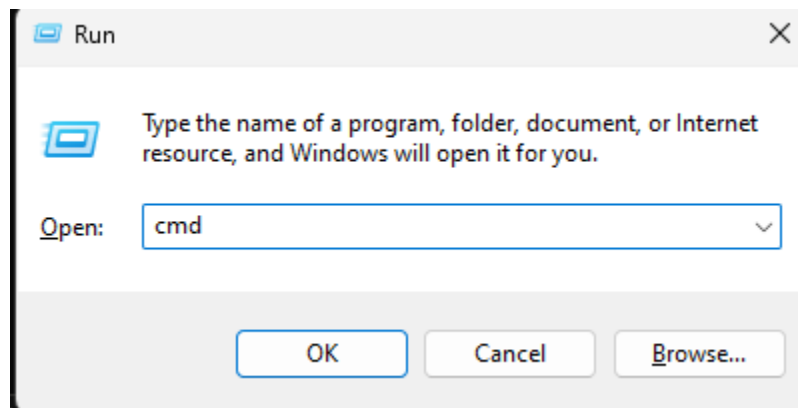
Don't check the automatically install..... option

Now click next again

Give it permission to run as admin and let it install

After it finishes installation, to verify if it was correctly installed
press windows + R key

Type cmd in it and press enter



```
Command Prompt
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Amaterasu>
```

A console will appear type

node -v

and press enter. If it returns a version

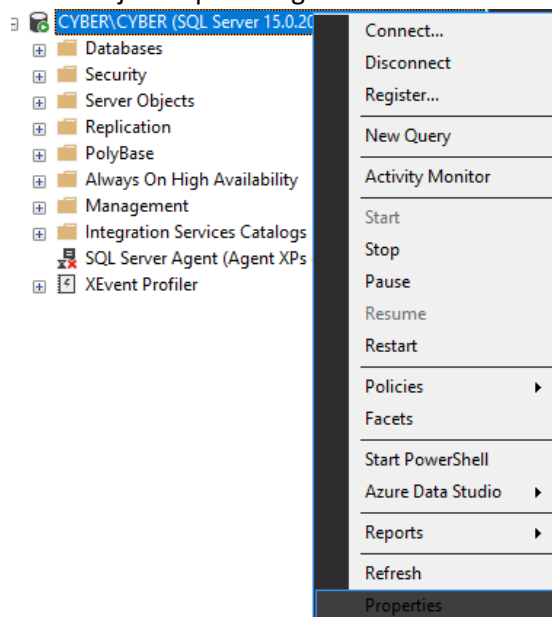
```
C:\Users\Amaterasu>node -v
v16.16.0
```

Then node js was installed correctly, if not try the steps again.

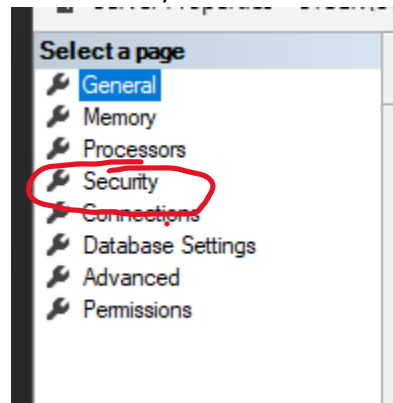
Step2: SETUP SQL USER

Open SSMS connect the server.

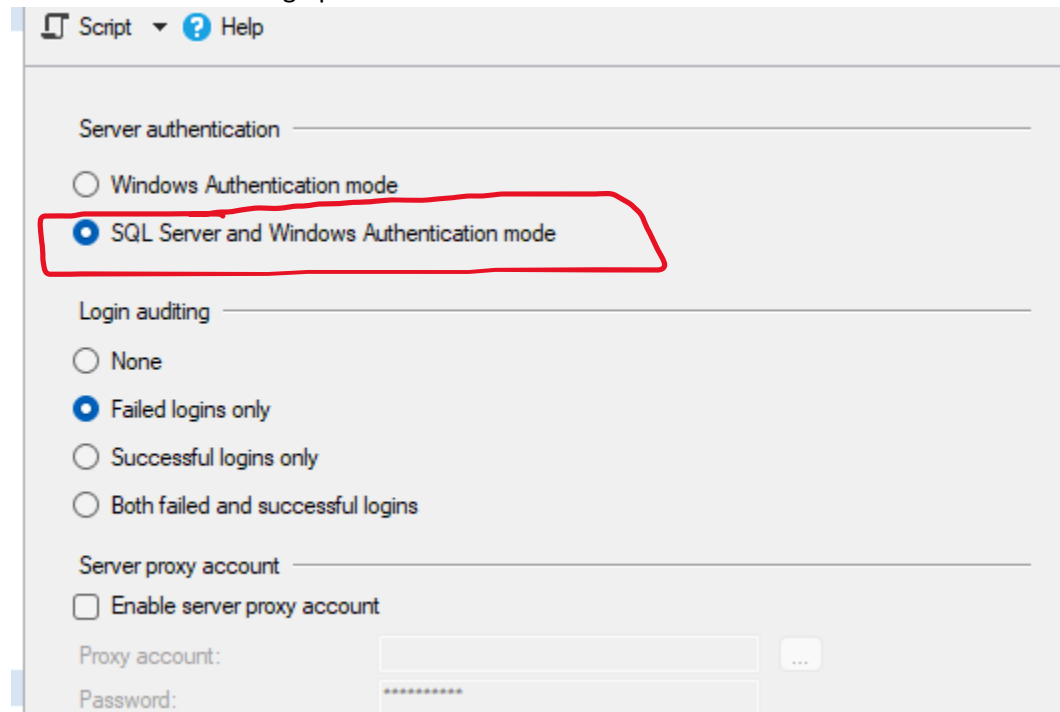
1. In the object explorer right click on the server and open properties



Goto Security:

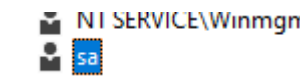


And check the following option



Click on ok.

2. In the object explorer, expand the security tab, then expand the logins tab as well, at the end of list you will see a user named 'sa'



Right click on it and open properties

On this page

Login Properties - sa

Select a page

- General
- Server Roles
- User Mapping
- Status

Script Help

Login name: sa Search...

☐ Windows authentication

☒ SQL Server authentication

Password:

Confirm password:

☐ Specify old password

Old password:

☐ Enforce password policy

☐ Enforce password expiration

☐ User must change password at next login

☐ Mapped to certificate

☐ Mapped to asymmetric key

☐ Map to Credential

Mapped Credentials

| Credential | Provider |
|------------|----------|
|------------|----------|

Add

Remove

Default database: master

Default language: English - us_english

OK Cancel

Change the password, I am going to change it to 123

Then go to status from left bar

Select a page

- General
- Server Roles
- User Mapping
- Status

Check the following settings

Settings

Permission to connect to database engine:

☒ Grant

☐ Deny

Login:

☒ Enabled

☐ Disabled

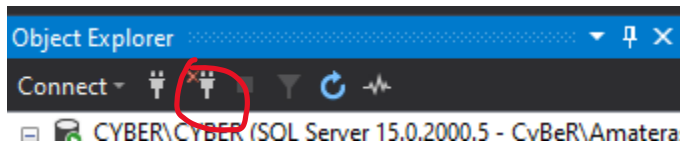
Status

SQL Server authentication:

☒ Login is locked out

And press Ok

Now disconnect your server from object explorer



Then Click on connect again



Server name: CYBER\CYBER

Authentication: SQL Server Authentication

Login: Windows Authentication

Password: SQL Server Authentication

Azure Active Directory - Universal with MFA

Select sql server authentication from Authentication:

Then enter the credentials

Server name: CYBER\CYBER

Authentication: SQL Server Authentication

Login: sa

Password: ...

☐ Remember password

Use the password u set for 'sa' user.

And click on connect.

Click on new query

Execute the following queries ONE BY ONE!

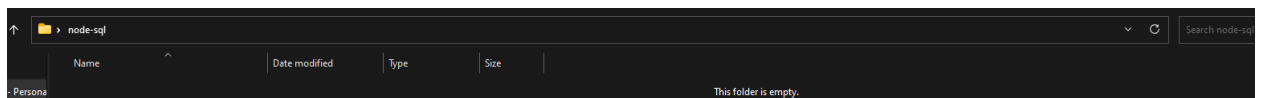
1. `CREATE DATABASE nodedb`
2. `USE nodedb`
3. `CREATE TABLE games(
 id int,
 name nvarchar(30)
);`
4. `INSERT INTO games VALUES(1, 'VALORANT'), (2, 'Call of duty Warzone')`
5. `SELECT * FROM games`

Execute the 5th query to verify if a table has been created and data has been inserted

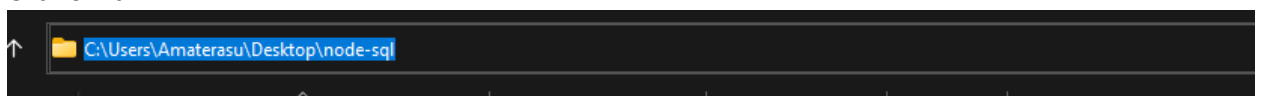
| Results | | | Messages |
|---------|----|----------------------|----------|
| | id | name | |
| 1 | 1 | VALORANT | |
| 2 | 2 | Call of duty Warzone | |

Step3: Setup node project

1. Navigate to desktop
2. Create a new folder
3. Open the folder
4. Click on the address bar



5. Click on it



6. Type cmd in it and press enter,
7. Command prompt will be opened.

```
Microsoft Windows [Version 10.0.22621.1702]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Amaterasu\Desktop\node-sql>
```

Type

`npm init`

and press enter

```
npm WARN config global '--global', '--local' are deprecated. Use '--location=global' instead.
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See 'npm help init' for definitive documentation on these fields
and exactly what they do.

Use 'npm install <pkg>' afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (sql) |
```

Keep pressing enter until you reach the following screen

```
Press ^C at any time to quit.
package name: (sql)
version: (1.0.0)
description:
entry point: (index.js)
test command:
git repository:
keywords:
author:
license: (ISC)
About to write to C:\Users\Amaterasu\Desktop\node-sql\package.json:

{
  "name": "sql",
  "version": "1.0.0",
  "description": "",
  "main": "index.js",
  "scripts": {
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "author": "",
  "license": "ISC"
}

Is this OK? (yes)

C:\Users\Amaterasu\Desktop\node-sql>
```

Now type & press enter

```
npm install tedious
```

and let the library install, After the library is installed you will see the following screen

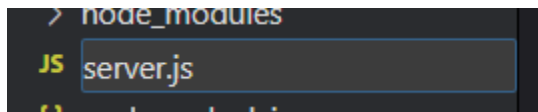
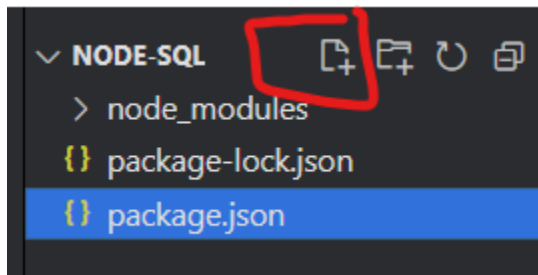
```
found 0 vulnerabilities
```

Step4: Open the project.

Open the project you created in any code editor that you are using but I recommend VS Code as I will be using it as well.

Open the folder in VS Code.

And create a file name it server.js(you can use any name u want, but make sure the extension is .js)



Press Enter

Now Paste this code in server.js file

```

var Connection = require('tedious').Connection;
let config = {
  server: 'CYBER',
  authentication: {
    type: 'default',
    options: {
      userName: 'sa',
      password: '123'
    }
  },
  options: {
    database: 'nodedb',
    validateBulkLoadParameters: false,
    trustServerCertificate: true
  }
}
var connection = new Connection(config);
connection.on('connect', function (err) {
  console.log('connected')
  executeStatement()
});

connection.connect();

var Request = require('tedious').Request;

function executeStatement() {
  request = new Request("SELECT * FROM games", function (err, rowCount) {
    if (err) {
      console.log(err);
    } else {
      console.log(rowCount + ' rows');
      // and we close the connection
      connection.close()
    }
  });

  request.on('row', function (columns) {
    columns.forEach(function (column) {
      console.log(column.value);
    });
  });

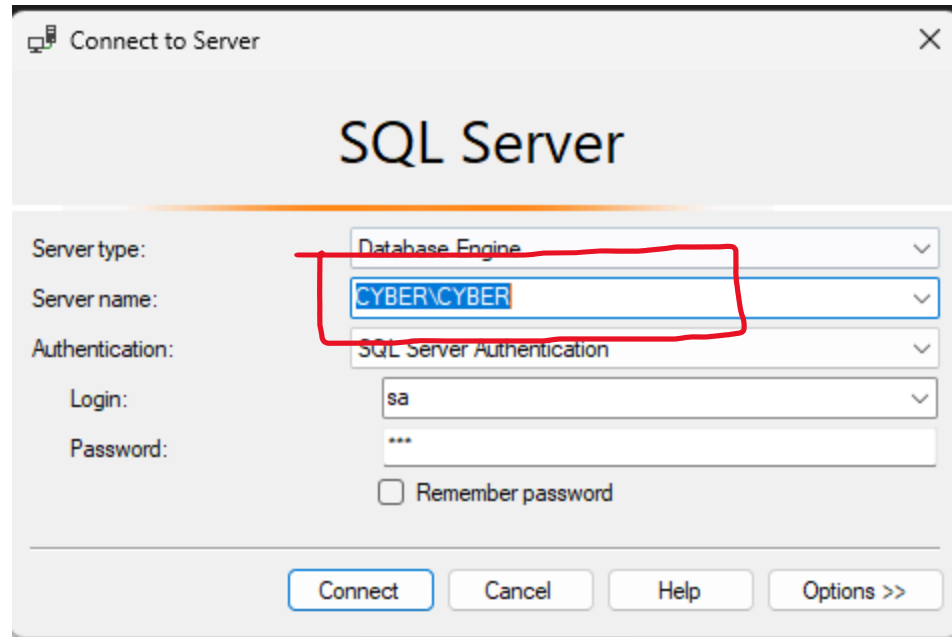
  connection.execSql(request); }

```

Update the following things in code

1. `server: 'CYBER',`

Change this to the server name of your sql server, your server name is written in the connection box of ssms software



The Name after '\ ' is the server name

2. Change the password in

```
options: {  
    userName: 'sa',  
    password: '123' // Change  
}
```

If you used any other.

3. Change database name if you created other than

```
database: 'nodedb', // change between the single quotes
```

4. Also change the tableName in the query if you created a table with other name

5. `request = new Request("SELECT * FROM games", function (err, rowCount)`

Step5: Run the script.

Now open the console of editor by pression CTRL + backtick key(if you are using vscode) [The backtick key is key beneath ESC Key]

A terminal will be opened at the bottom of the editor.

Type the following command in it

```
node server.js
```

and press enter,

If it return the following output then you have successfully connected nodejs with sql server

P.S. If you created .js file having name other then server

Then type the command with your file name

```
node yourfilenamehere.js
```

```
PS C:\Users\Amaterasu\Desktop\node-sql> node server.js
connected
1
VALORANT
2
Call of duty Warzone
2 rows
```

If it doesn't, try to follow the steps again carefully.

Code is like humor. When you have to explain it, it's bad. – Cory House
