

In searching for the best way to use Javascript in reading from and writing to a MySQL database, a number of options were explored, including calling an HTTP API for database I/O, using a browser plugin for database API access, and using internet browsers via localStorage or IndexedDB to store one's own data locally. We decided that using node.js is probably our best option, however, as it seems to be the most widely used method with the most guides available for reference, and a number of our team members even have experience using it.

Connecting to a MySQL database from node.js involves first creating a folder for storing the node.js app, initializing it using “npm init”, then installing node.js for MySQL using “npm install mysql”. One must then create the connect.js file, which will contain code for connecting to the MySQL database server. The contents of a sample connect.js file are detailed in screenshots below (from <https://www.mysqltutorial.org/mysql-nodejs/connect/>).

Connecting to MySQL database server from node.js

First, import the `mysql` module by using the following statement:

```
let mysql = require('mysql');
```

Second, create a connection to the MySQL database by calling the `createConnection()` method and providing the detailed information on MySQL server such as host, user, password and database as follows:

```
let connection = mysql.createConnection({
  host: 'localhost',
  user: 'root',
  password: '',
  database: 'todoapp'
});
```

In this example, we created a connection to `todoapp` database in the local database server.

Third, call the `connect()` method on the `connection` object to connect to the MySQL database server:

```
connection.connect(function(err) {  
  if (err) {  
    return console.error('error: ' + err.message);  
  }  
  
  console.log('Connected to the MySQL server.');
```

The `connect()` method accepts a callback function that has the `err` argument which provides the detailed error if any error occurred.

Let's test `connect.js` program.

```
> node connect.js  
Connected to the MySQL server
```

If you see the message "connected to the MySQL server", then congratulation, you have been successfully connected to MySQL database server from the node.js application.