Mysql-node 连接：

1. Npm init： 初始化npm数据保管包
2. Npm i express mysql --save：安装express， mysql 第三方包
3. 创建文件 app.js
4. 倒入包： express，mysql

const express = require("express");

const mysql = require("mysql");

1. 实例化 app 对象

const app = express();

1. 连接mysql

const connection = mysql.createConnection({

host: "localhost",

user: "root",

password: "Jj89757123!",

database: "my\_database"

});

connection.connect(err => {

*if* (err) *throw* err;

console.log("mysql is connected.....");

});

1. 监听接口

app.listen("59999", () => {

console.log("server on port: localhost:59999");

});

1. 基本操作
2. *//创建数据库请求*
3. app.get("/createdb", (req, res) => {
4. let sql = "create database nodemysql";
5. connection.query(sql, (err, result) => {
6. *if* (err) *throw* err;
7. console.log(result);
8. res.send("database created");
9. });
10. });
11. *//创建表*
12. app.get("/createtable", (req, res) => {
13. let sql =
14. "create table node(id int auto\_increment not null primary key, title varchar(255), body varchar(255))";
15. connection.query(sql, (err, result) => {
16. *if* (err) *throw* err;
17. console.log(result);
18. res.send("table is created.....");
19. });
20. });
21. *//插入语句*
22. app.get("/addpost", (req, res) => {
23. let post = { title: "post one", body: "this is post one" };
24. let sql = "insert into node set ?";
25. connection.query(sql, post, (err, result) => {
26. *if* (err) *throw* err;
27. console.log(result);
28. res.send("data is created.....");
29. });
30. });
31. *//查询语句*
32. app.get("/getpost", (req, res) => {
33. let sql = "select \* from node";
34. connection.query(sql, (err, result) => {
35. *if* (err) *throw* err;
36. console.log(result);
37. res.send(result);
38. });
39. });
40. *//查询单条语句*
41. app.get("/getpost/:id", (req, res) => {
42. let sql = `select \* from node where id = ${req.params.id}`;
43. connection.query(sql, (err, result) => {
44. *if* (err) *throw* err;
45. console.log(result);
46. res.send(result);
47. });
48. });
49. *//跟新·内容*
50. app.get("/updatepost/:id", (req, res) => {
51. let updatePost = "this is updated post";
52. let sql = `update node set body = '${updatePost}' where id = ${req.params.id}`;
53. connection.query(sql, (err, result) => {
54. *if* (err) *throw* err;
55. console.log(result);
56. res.json("更新成功");
57. });
58. });
59. *//删除*
60. app.get("/deletepost/:id", (req, res) => {
61. let sql = `delete from node where id = ${req.params.id}`;
62. connection.query(sql, (err, result) => {
63. *if* (err) *throw* err;
64. console.log(result);
65. res.json("删除成功");
66. });
67. });