高级node第十单元(6.11)

2020年6月9日 15:25

集成mysql的服务器应用
数据库表之间的关系以及设计原则
实现文件上传和下载(前后端)

koa中的路由可以匹配多个处理函数

```
router.get('/test', async (ctx, next) => {
  ctx.body = 'test'
  await next();
}, async (ctx) => {
  ctx.body = 'test1'
})
```

封装mysql的链接

1. 插入数据的封装

```
/**
 * @param {string} tableName 表名
 * @param {plain object} data 要插入的数据
 *
 * @return {Promise<code: 1 / 0, err/info>}
 */
export async function insert(tableName, data) {
  let keys = Object.keys(data);
  let values = keys.map(item => {
    if (typeof data[item] === 'string') {
      return `'${data[item]}'`
    }
    return data[item];
  });
  // 创建对应的sql语句
  let sql = `insert into ${tableName} (${keys.join(',')}) values (${values.join(',')})`;
  // 响应与数据库交互的结果
  let result;
  try {
    let succ = await new Promise((reslove, reject) => {
      connection.query(sql, (err, info) => {
        err ? reject(err) : reslove(info);
      });
    });
    result = { code: 1, info: succ }
  } catch (err) {
    result = { err, code: 0 };
  }
  return result;
}
```

2. 完整的封装，看如下目录

mysql的链接与断开

每次查询结束之后，应该要断开链接，每次交互前应该要重新建立链接对象

```
async function dealResult(sql) {
  // 与数据库建立一个链接对象
  const connection = mysql.createConnection(config);
  // 响应与数据库交互的结果
  let result;
  try {
    let succ = await new Promise((reslove, reject) => {
      connection.query(sql, (err, info) => {

    result = { err, code: 0 };
  }
  connection.end(); //断开链接
  return result;
}
```

可以通过扩展ctx的方式去添加我们封装的sql方法

```
import Koa from 'koa'
import { select, update, inster, query } from './database'
const app = new Koa();
// 给ctx扩展与数据库交互的方法
app.context.db = {
  select, update, inster, query
}


export default app;
```

在各个请求的接口中，可以直接调用

```
export default async (ctx) => {
  let result = await ctx.db.select('users')
  ctx.body = result;
}
```

数据库设计原则

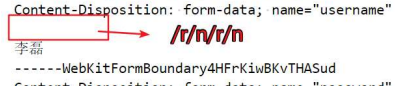
1. 应该创建多少张表
   在描述的一段数据关系中有多少类的对象存在，应该为每一类对象创建一张表
2. 每张表中的字段
   a. 表的字段代表的是列，也描述了这类对象的特性
   b. 主键 每张表必须有的，主键必须保持唯一性，最好无实际意义
   c. 外键 表名表中的每条数据和其他表的关系，不一定存在，一般存在外键的话，该外键一般都是别的 表的主键
3. 表和表之间的关系
   a. 一对一关系 （用一种表就可以表示）
   b. 一对多关系 （用两张表就可以表示）
   c. 多对多关系 （用三张表才可以表示），需要额外的建一张表明两张表中数据关系的表

在服务端接收上传的文件（如何处理formData格式的数据）

1. formData数据格式是什么样子的呢

2. 边界如何获取
   边界字符串是自动生成的，存储于content-Type里边

3. formData中的每一段数据的key和value是通过 /r/n/r/n划分开的



4. 具体的实现在以下目录

独立的用户系统的功能

1. 注册
2. 登录
3. 登录的验证（token）
4. 权限的判断和验证
5. 提供一个可以进行用户管理的界面