**Nestjs 中使用Mongoose 实现数据库的增删改查**

**目录**

[一、mongoose 官方文档 1](#_bookmark0)

[二、实现Nav 集合的增删改查 1](#_bookmark1)

**一、mongoose 官方文档**

<https://mongoosejs.com/>

# 二、实现 Nav 集合的增删改查

## 1、创建接口

export interface NavInterface {

\_id?:String title?: String; url?: String; status?:Number;

}

**2、定义 nav 的服务**

import { Injectable } from '@nestjs/common';

import { InjectModel } from '@nestjs/mongoose'; import {NavInterface} from '../../interface/nav.interface'

@Injectable()

export class NavService {

constructor(@InjectModel('Nav') private navModel) { } async findAll(json:NavInterface = {}) {

var result = await this.navModel.find(json).exec(); return result;

}

async add(json:NavInterface) {

var nav = new this.navModel(json); var result=await nav.save();

return result;

}

async update(json1:NavInterface,json2:NavInterface) {

var result=await this.navModel.updateOne(json1,json2); return result;

}

async delete(json1:NavInterface) {

var result=await this.navModel.deleteOne(json1); return result;

}

}

**3、定义 nav 的控制器，调用服务实现增**

import { Controller, Get } from '@nestjs/common';

import { NavService } from '../../../../service/nav/nav.service'; @Controller('admin/nav')

export class NavController {

constructor(private navService:NavService){} @Get()

async index(){

var result=await this.navService.findAll(); return result;

}

@Get("add") async add(){

await this.navService.add({ title: '测试',

url: 'http://www.itying.com', status:1

});

return '增加成功';

}

@Get("update") async update(){

await this.navService.update({"\_id":"5d8330c9aa16ed2f681cafb4"},{"title":"这是测试需改的数据"}); return '修改成功';

}

@Get("delete") async delete(){

await this.navService.delete({"\_id":"5d8330ea630bd9088cd7d716"}); return '删除成功';

}

}