

Nestjs 中使用 TypeORM 操作 Mysql 数据库、实现数据库的增 删 改 查

目录

一、 关于 TypeORM.....	1
二、安装 TypeORM 操作 Mysql 模块.....	1
三、配置数据库连接地址.....	2
三、配置实体 entity.....	2
四、在控制器对应的 Module 中配置 Model.....	3
五、在服务里面使用@InjectRepository 获取数据库 Model 实现操作数据库.....	3
六、 TypeORM 操作 mysql 数据库实现增删改查.....	4

一、关于 TypeORM

TypeORM 是一个 ORM 框架，是一款比较成熟的对象关系映射器，它是由 typescript 写的。支持 MySQL, PostgreSQL, MariaDB, SQLite, MS SQL Server, Oracle, WebSQL 等数据库。

二、安装 TypeORM 操作 Mysql 模块

Nest 操作 Mysql 官方文档: <https://docs.nestjs.com/techniques/database>

```
npm install --save @nestjs/typeorm typeorm mysql
```

三、配置数据库连接地址

在 `app.module.ts` 中配置数据库连接

```
import { Module } from '@nestjs/common';
import { TypeOrmModule } from '@nestjs/typeorm';
@Module({
  imports:
    [ TypeOrmModule.forRoot({
      type: 'mysql',
      host: 'localhost',
      port: 3306,
      username: 'root',
      password: '123456',
      database: 'test',
      entities: [__dirname + '**/*.entity{.ts,.js}'],synchronize: true,
    })],
})
export class AppModule {}
```

三、配置实体 entity

```
import { PrimaryGeneratedColumn, Column, Entity } from "typeorm"; @Entity()
export class Nav
{ @PrimaryGeneratedColumn() id:
  number;

  @Column({length: 45}) name: string;

  @Column({length:255}) url: string;

  @Column('int') status: number;
}
```

四、在控制器对应的 Module 中配置Model

```
import { Module } from '@nestjs/common';
import { UserController } from '../controller/user/user.controller';
import { NewsController } from
'../controller/news/news.controller';

import { TypeOrmModule } from

'@nestjs/typeorm'; import { Nav } from

'../entity/nav.entity';

import { Navinfo } from '../entity/navinfo.entity';

import { AppService } from '../app.service';

@Module({ imports:[TypeOrmModule.forFeature([
  Nav,Navinfo])], controllers: [UserController,
  NewsController], providers:[AppService]
})
export class AdminModule {}
```

五、在服务里面使用@InjectRepository 获取数据库 Model 实现操作数据库

```
import { Injectable } from '@nestjs/common';
import { InjectRepository } from
'@nestjs/typeorm'; import { Repository }
from 'typeorm';
import { Nav } from '../entity/nav.entity';
import { Navinfo } from '../entity/navinfo.entity';
@Injectable()
export class
  AppService
  { constructor(
    @InjectRepository(Nav) private readonly navRepository: Repository<Nav>,
    @InjectRepository(Navinfo) private readonly navinfoRepository:
    Repository<Navinfo>,
  ) {}
  async findAll(){
    return await this.navRepository.find();
  }
}
```

六、TypeORM 操作 mysql 数据库实现增删改查

查找:

```
async findAll(){
    // return await this.navRepository.find();

    // return await this.navRepository.findOne(1);

    // return await this.navRepository.findOne({"name":"国际"});

    // return await this.navRepository.find({"id":1});

    // return await this.navRepository.query("select * from nav");

    // return await this.navRepository.count();

    let [allPhotos, photosCount] = await this.navRepository.findAndCount();
    console.log("All photos: ", allPhotos);
    console.log("Photos count: ", photosCount);
    return await this.navRepository.find();
}
```

增加数据

```
async add(){
    var nav=new Nav()
    nav.name="友情链接";
    nav.url='itying.com'

    return await this.navRepository.save(nav);
}
```

修改数据

```
async update(){

    let navUpdate = await this.navRepository.findOne(3);
    navUpdate.name = "搞笑";
    return await this.navRepository.save(navUpdate);

    // return await this.navRepository.update()
}
```

删除数据

```
async delete(){

    let navRemove = await this.navRepository.findOne(5);

    return await this.navRepository.remove(navRemove);
}
```

其他更多操作: <https://typeorm.io/#/>