Mongoose

讲师：李立超

简介

- 之前我们都是通过shell来完成对数据库的各种操作的，在开发中大部分时候我们都需要通过程序来完成对数据库的操作。
- 而Mongoose就是一个让我们可以通过Node来操作MongoDB的模块。
- Mongoose是一个对象文档模型（ODM）库，它对Node原生的MongoDB模块进行了进一步的优化封装，并提供了更多的功能。
- 在大多数情况下，它被用来把结构化的模式应用到一个MongoDB集合，并提供了验证和类型转换等好处

mongoose的好处

- 可以为文档创建一个模式结构（Schema）
- 可以对模型中的对象/文档进行验证
- 数据可以通过类型转换转换为对象模型
- 可以使用中间件来应用业务逻辑挂钩
- 比Node原生的MongoDB驱动更容易

新的对象

- mongoose中为我们提供了几个新的对象
  - Schema(模式对象)
    - Schema对象定义约束了数据库中的文档结构
  - Model
    - Model对象作为集合中的所有文档的表示，相当于MongoDB数据库中的集合collection
  - Document
    - Document表示集合中的具体文档，相当于集合中的一个具体的文档

通过mongoose连接MongoDB

- 使用Mongoose必须先安装mongoose包
  - npm install mongoose
- 加载Mongoose
  - const mongoose = require("mongoose")
- 连接数据库
  - mongoose.connect("mongodb://地址")
  - 地址例子：mongodb://127.0.0.1/mg_test
- 断开连接
  - mongoose.disconnect()

connection

- 一旦连接了MongoDB数据库，底层的Connection对象就可以通过mongoose模块的conection属性来访问。
- connection对象是对数据库连接的抽象，它提供了对象连接、底层的Db对象和表示结合的Model对象的访问。
- 并且可以对connection对象上的一些事件进行监听，来获悉数据库连接的开始与端开。
- 比如，可以通过open和close事件来监控连接的打开和关闭。

Schema模式对象

- 使用Mongoose你必须经常定义模式。
- 模式为集合中的文档定义字段和字段类型。
- 如果你的数据是被结构化成支持模式的，这是非常有用的。
- 简单来说，模式就是对文档的约束，有了模式，文档中的字段必须符合模式的规定。否则将不能正常操作。

定义模式

- 模式为集合中的文档定义字段和字段类型。
- 对于在模式中的每个字段，你都需要定一个特定的值类型。受支持的类型如下：
  - String
  - Number
  - Boolean
  - Array
  - Buffer
  - Date
  - ObjectId或Oid
  - Mixed
- 需要为你计划使用的每个不同的文档类型都定义一个模式。

创建模式定义

- 模式需要通过mongoose的Schema属性来创建，这个属性是一个构造函数。
  - new Schema(definition,option)
    - definition（描述模式）
    - options 配置对象，定义与数据库中集合的交互

```
//创建schema
let userSchema = new mongoose.Schema({
    username:{type:String,index:1,unique:true,require:true},
    password:String,
    hobby:[String],
    age:Number
});
```

options常用选项

- autoIndex
  - 布尔值，开启自动索引，默认true
- bufferCommands
  - 布尔值，缓存由于连接问题无法执行的语句，默认true
- capped
  - 集合中最大文档数量
- collection
  - 指定应用Schema的集合名称
- id
  - 布尔值，是否有应用于_id的id处理器，默认true
- _id
  - 布尔值，是否自动分配id字段，默认true
- strict
  - 布尔值，不符合Schema的对象不会被插入进数据库，默认true

Model模型对象

- 一旦定义好了Schema对象，就需要通过该Schema对象来创建Model对象。
- 一旦创建好了Model对象，就会自动和数据库中对应的集合建立连接，以确保在应用更改时，集合已经创建并具有适当的索引，且设置了必须性和唯一性。
- Model对象就相当于数据库中的集合，通过Model可以完成对集合的CRUD操作。

Model模型对象

- 创建模型对象需要使用mongoose的model()方法，语法如下：
  - model(name, [schema], [collection] , [skipInit])
    - name参数相当于模型的名字，以后可以同过name找到模型。
    - schema是创建好的模式对象。
    - collection是要连接的集合名。
    - skipInit是否跳过初始化，默认是false。
- 一旦把一个Schema对象编译成一个Model对象，你就完全准备好开始在模型中访问、添加、删除、更新和删除文档了。也就是说有了模型以后我们就可以操作数据库了。

Model对象的方法

- remove(conditions, callback)
- deleteOne(conditions, callback)
- deleteMany(conditions, callback)
- find(conditions, projection, options, callback)
- findById(id, projection, options, callback)
- findOne(conditions, projection, options, callback)
- count(conditions, callback)
- create(doc, callback)
- update(conditions, doc, options, callback)
- 等等

Document文档对象

- 通过Model对数据库进行查询时，会返回Document对象或Document对象数组。
- Document继承自Model，代表一个集合中的文档。
- Document对象也可以和数据库进行交互操作。

Document对象的方法

- equals(doc)
- id
- get(path,[type])
- set(path,value,[type])
- update(update,[options],[callback])
- save([callback])
- remove([callback])
- isNew
- isInit(path)
- toJSON()
- toObject()


尚硅谷
www.atguigu.com