



黑马程序员™
www.itheima.com

传智播客旗下
高端IT教育品牌

MongoDB数据库

目录 Contents

- ◆ 数据库概述及环境搭建
- ◆ MongoDB增删改查操作

1. 数据库概述及环境搭建

1.1 为什么要使用数据库

- 动态网站中的数据都是存储在数据库中的
- 数据库可以用来持久存储客户端通过表单收集的用户信息
- 数据库软件本身可以对数据进行高效的管理

<http://www.czxy.com/article?id=1>

<http://www.czxy.com/article?id=2>

你好，新传智人 | 一场开学典礼与你一起遇见未来!

来源 传智专修学院

你好，新传智人
经过一场开学典礼
感谢你们与传智相遇



传智专修学院2018年教师节庆祝暨表彰大会完美谢幕

来源 传智专修学院

在这充满丰收和喜悦的九月里，第34个教师节翩然而至。

9月10日下午，传智专修学院2018年教师节庆祝暨表彰大会在学院教学楼405室顺利举行。学院校长冯威、校党委书记陈力的主任宋海荣、软件工程系主任崔海凡、软件工程系副主任王向磊、学生处主任王利霞、校团委书记陈慧达出席会议，优秀教师代表、学院全体教职工及学生代表参加此次大会。

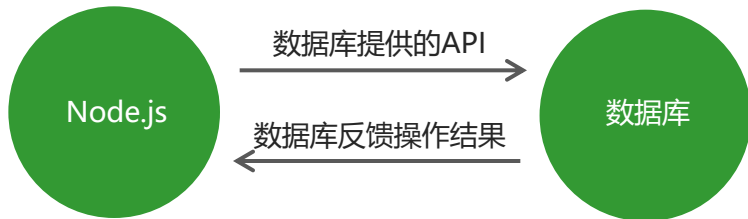


1.数据库概述及环境搭建

1.2 什么是数据库

数据库即存储数据的仓库，可以将数据进行有序的分门别类的存储。它是独立于语言之外的软件，可以通过API去操作它。

常见的数据库软件有：mysql、mongoDB、oracle。



 mongoDB

 MySQL™

1. 数据库概述及环境搭建

1.3 MongoDB数据库下载安装

下载地址: <https://www.mongodb.com/download-center/community>

The screenshot shows the MongoDB download center interface. At the top, there's a navigation bar with links like DOCS, LEARN, WHAT'S MONGODB?, BLOG, LOGIN, and a 'Get MongoDB' button. Below this, there's a section for 'SOLUTIONS' with tabs for Cloud, Server, and Tools. The 'Server' tab is selected, and it prompts the user to 'Select the server you would like to run:'. There are two main options: 'MongoDB Community Server' (highlighted with a green bar and the text 'FEATURE RICH. DEVELOPER READY.') and 'MongoDB Enterprise Server' (with the text 'ADVANCED FEATURES. PERFORMANCE GRADE.'). Under the Community Server section, there are dropdown menus for 'Version' (set to 4.0.4 (current release)) and 'OS' (set to Windows 64-bit x64). Below these is a 'Package' dropdown set to 'MSI' and a large green 'Download' button. A URL is provided: https://fastdl.mongodb.org/win32/mongodb-win32-x86_64-2008plus-ssl-4.0.4-signed.msi. To the right of the download options, there's a list of links: Release notes, Changelog, All version binaries, Installation instructions, Download source (tgz), and Download source (zip). At the bottom, a paragraph explains that MongoDB offers both an Enterprise and Community version, and that the Enterprise version is available as part of the MongoDB Enterprise Advanced subscription.

mongoDB. | FOR GIANT IDEAS

SOLUTIONS CLOUD CUSTOMERS RESOURCES ABOUT US

Cloud Server Tools

Select the server you would like to run:

MongoDB Community Server
FEATURE RICH. DEVELOPER READY.

MongoDB Enterprise Server
ADVANCED FEATURES. PERFORMANCE GRADE.

Version: 4.0.4 (current release)

OS: Windows 64-bit x64

Package: MSI

Download

https://fastdl.mongodb.org/win32/mongodb-win32-x86_64-2008plus-ssl-4.0.4-signed.msi

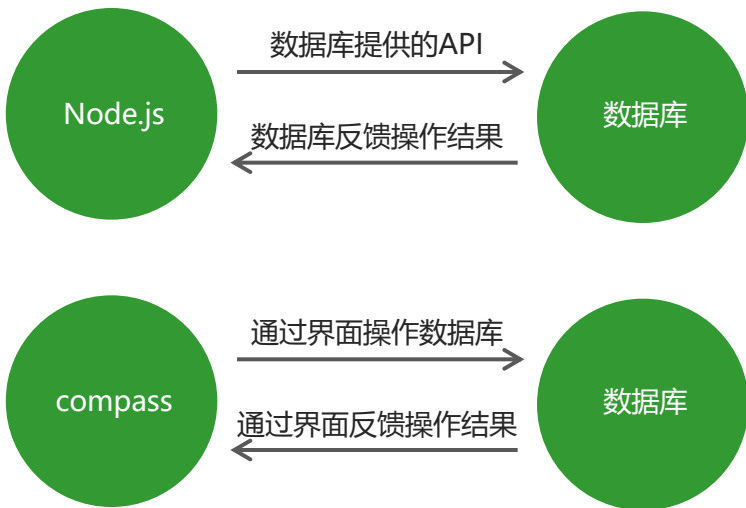
- Release notes
- Changelog
- All version binaries
- Installation instructions
- Download source (tgz)
- Download source (zip)

MongoDB offers both an Enterprise and Community version of its powerful non-relational database. MongoDB Enterprise is available as part of the MongoDB Enterprise Advanced subscription, which features the most comprehensive support for MongoDB and the best SLA, in addition to Ops Manager, Compass, and the Connector for BI.

1.数据库概述及环境搭建

1.4 MongoDB可视化软件

MongoDB可视化操作软件，是使用图形界面操作数据库的一种方式。



1.数据库概述及环境搭建

1.5 数据库相关概念

在一个数据库软件中可以包含多个数据仓库，在每个数据仓库中可以包含多个数据集合，每个数据集合中可以包含多条文档（具体的数据）。

术语	解释说明
database	数据库，mongoDB数据库软件中可以建立多个数据库
collection	集合，一组数据的集合，可以理解为JavaScript中的数组
document	文档，一条具体的数据，可以理解为JavaScript中的对象
field	字段，文档中的属性名称，可以理解为JavaScript中的对象属性



1.数据库概述及环境搭建

1.6 Mongoose第三方包

- 使用Node.js操作MongoDB数据库需要依赖Node.js第三方包mongoose
- 使用**npm install mongoose**命令下载

■ 1.数据库概述及环境搭建

1.7 启动MongoDB

在命令行工具中运行**net start mongoDB**即可启动MongoDB，否则MongoDB将无法连接。



1.数据库概述及环境搭建

1.8 数据库连接

使用mongoose提供的**connect**方法即可连接数据库。

```
mongoose.connect('mongodb://localhost/playground')  
  .then(() => console.log('数据库连接成功'))  
  .catch(err => console.log('数据库连接失败', err));
```

2. MongoDB数据库连接

1.9 创建数据库

在MongoDB中**不需要显式创建数据库**，如果正在使用的数据库不存在，**MongoDB会自动创建**。

目录 Contents

- ◆ 数据库概述及环境搭建
- ◆ MongoDB增删改查操作

3.MongoDB增删改查操作

3.1 创建集合

创建集合分为两步，一是对**对集合设定规则**，二是**创建集合**，创建mongoose.Schema构造函数的实例即可创建集合。

// 设定集合规则

```
const courseSchema = new mongoose.Schema({  
  name: String,  
  author: String,  
  isPublished: Boolean  
});
```

// 创建集合并应用规则

```
const Course = mongoose.model('Course', courseSchema); // courses
```

3.MongoDB增删改查操作

3.2 创建文档

创建文档实际上就是**向集合中插入数据**。

分为两步：

- ① 创建集合实例。
- ② 调用实例对象下的save方法将数据保存到数据库中。

// 创建集合实例

```
const course = new Course({  
  name: 'Node.js course',  
  author: '黑马讲师',  
  tags: ['node', 'backend'],  
  isPublished: true  
});
```

// 将数据保存到数据库中

```
course.save();
```

3.MongoDB增删改查操作

3.2 创建文档

```
Course.create({name: 'JavaScript基础', author: '黑马讲师', isPublish: true}, (err, doc) => {  
    // 错误对象  
    console.log(err)  
    // 当前插入的文档  
    console.log(doc)  
});
```

```
Course.create({name: 'JavaScript基础', author: '黑马讲师', isPublish: true})  
    .then(doc => console.log(doc))  
    .catch(err => console.log(err))
```

■ 3.MongoDB增删改查操作

3.3 mongoDB数据库导入数据

mongoimport -d 数据库名称 -c 集合名称 -file 要导入的数据文件

找到mongodb数据库的安装目录，将安装目录下的bin目录放置在环境变量中。

3.MongoDB增删改查操作

3.4 查询文档

// 根据条件查找文档 (条件为空则查找所有文档)

```
Course.find().then(result => console.log(result))
```

// 返回文档集合

```
[{  
  _id: 5c0917ed37ec9b03c07cf95f,  
  name: 'node.js基础',  
  author: '黑马讲师',  
},{  
  _id: 5c09dea28acfb814980ff827,  
  name: 'Javascript',  
  author: '黑马讲师',  
}]
```

3.MongoDB增删改查操作

3.4 查询文档

// 根据条件查找文档

```
Course.findOne({name: 'node.js基础'}).then(result => console.log(result))
```

// 返回文档

```
{  
  _id: 5c0917ed37ec9b03c07cf95f,  
  name: 'node.js基础',  
  author: '黑马讲师',  
}
```

3.MongoDB增删改查操作

3.4 查询文档

// 匹配大于 小于

```
User.find({age: {$gt: 20, $lt: 50}}).then(result => console.log(result))
```

// 匹配包含

```
User.find({hobbies: {$in: ['敲代码']}}).then(result => console.log(result))
```

// 选择要查询的字段

```
User.find().select('name email').then(result => console.log(result))
```

// 将数据按照年龄进行排序

```
User.find().sort('age').then(result => console.log(result))
```

// skip 跳过多少条数据 limit 限制查询数量

```
User.find().skip(2).limit(2).then(result => console.log(result))
```

3.MongoDB增删改查操作

3.5 删除文档

// 删除单个

```
Course.findOneAndDelete({}).then(result => console.log(result))
```

// 删除多个

```
User.deleteMany({}).then(result => console.log(result))
```

3.MongoDB增删改查操作

3.6 更新文档

// 更新单个

```
User.updateOne({查询条件}, {要修改的值}).then(result => console.log(result))
```

// 更新多个

```
User.updateMany({查询条件}, {要更改的值}).then(result => console.log(result))
```

3.MongoDB增删改查操作

3.6 mongoose验证

在创建集合规则时，可以设置当前字段的验证规则，验证失败就则输入插入失败。

- required: true 必传字段
- minlength: 3 字符串最小长度
- maxlength: 20 字符串最大长度
- min: 2 数值最小为2
- max: 100 数值最大为100
- enum: ['html', 'css', 'javascript', 'node.js']
- trim: true 去除字符串两边的空格
- validate: 自定义验证器
- default: 默认值

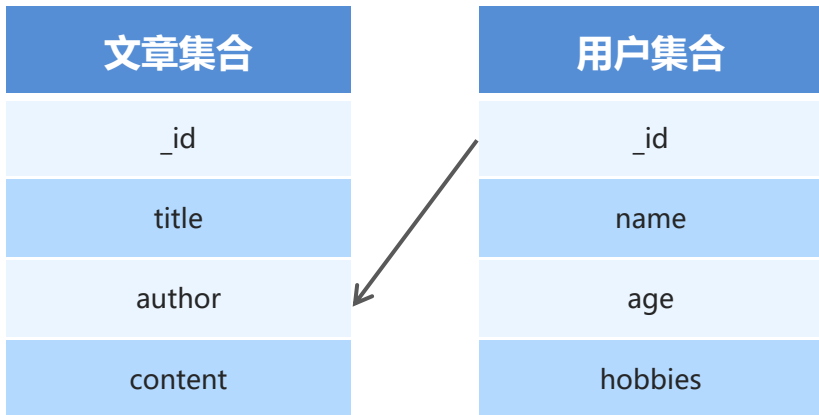
获取错误信息：error.errors['字段名称'].message

3.MongoDB增删改查操作

3.7 集合关联

通常**不同集合的数据之间是有关系的**，例如文章信息和用户信息存储在不同集合中，但文章是某个用户发表的，要查询文章的所有信息包括发表用户，就需要用到集合关联。

- 使用id对集合进行关联
- 使用populate方法进行关联集合查询



3.MongoDB增删改查操作

3.7 集合关联实现

// 用户集合

```
const User = mongoose.model('User', new mongoose.Schema({ name: { type: String } }));
```

// 文章集合

```
const Post = mongoose.model('Post', new mongoose.Schema({  
  title: { type: String },  
  // 使用ID将文章集合和作者集合进行关联  
  author: { type: mongoose.Schema.Types.ObjectId, ref: 'User' }  
}));
```

//联合查询

```
Post.find()  
  .populate('author')  
  .then((err, result) => console.log(result));
```