

# Node.js 服务端框架及 ORM 框架介绍

---

## node.js 服务端框架 express

---

在 node.js 中，有非常多优秀的服务端框架，例如 express/koa/think/egg 等等，其中很多框架都是由国内的团队完成的，也取得了不错的反响。

而影响力最大的两个服务端框架就是 express 和 koa，它们在 node.js 服务端框架来说是鼻祖一般的存在，很多框架都是基于他们的思路来实现和封装的。

### express

在服务端，我们要区分清楚 request 和 response 的概念，request 是由客户端发起的，经过服务端处理之后，服务端的返回内容叫做 response。

之前我们也简单了解过，node.js 内部有非常多的内置模块，其中就有 http 模块，**express/koa** 实际上就是对这个 **http** 模块的再封装，增加了中间件策略和其他各种路由的通用处理，让我们写起来更加方便。

## 使用 **express** 来定义一个 **http** 服务器

我们需要在一台已经安装了 node.js 的系统中，使用 npm 或其他包管理工具，安装我们的 express 框架。

```
npm install express
```

之后我们就能直接通过 node.js 的 CommonJS 模块化规范，加载 express 模块。

## 使用常见的 **express** 中间件

对于大型应用来说，我们的应用一般需要使用到 body-parser 和 cookie-parser 这两个中间件，他们的作用是处理我们常见请求中 body 和 cookie 的部分。

## 编写一个自定义 **express** 中间件

这里我们可以自行编写一个业务中会使用到的中间件，来加深我们的理解。

## 中间件原理

中间件实际上我们可以这样理解，就是对于需要多次书写的业务逻辑，我们可以使用一种切面的形式，对相同逻辑进行通用处理。

## sequelize ORM 框架

---

### 数据库分类及 mysql 简介

数据库（db）就是一类数据存储的集合，他能帮助我们有效的存储数据，分类数据。

其实文件系统就可以理解为一种最简单的「数据库」，我们也可以简单的直接把内容存储在文件里面，最终文件放置在硬盘中。

但是文件系统我们无法针对某个内容做更详细的分析，排序等等操作，数据库内部会对我们存储的数据进行归类分析，使得我们通过数据库相关的 sql 语法，就能快速的找到匹配的数据，还能保证诸如事务一致性等等非常多的特性，对于我们业务中存储的数据来说，使用数据库是非常有优势的。

### mysql

对于大多数入门级别的同学来说，我们只需要学习一种基础且简单的 mysql 数据库即可，其他的数据库我们就可以触类旁通。

首先我们可以自行的在我们的操作系统中安装一个 mysql 服务，设置好账号和密码，那么我们就可以通过账号密码登陆我们的数据库系统。

在 mysql 中，我们需要定义表 (table) 结构，每个表代表着一类数据的集合。例如对于用户来说，我们就可以定义一张用户表，把所有用户的数据存储在这里。

在定义表之前，我们还需要定义清楚每个表中存放哪些字段，这个字段对于这样表来说我们称之为列 (column) 。

具体的每一份数据，在我们的表中我们称之为行或者记录 (record)，这张表中的每一行记录，都有着我们定义的列中的数据。

## 为什么我们需要 ORM 框架

有的时候，我们使用 node.js 做一个脚本工具，写一些 node.js 脚本进行执行，有时候我们可以使用 node.js 做一个中间层，对我们的请求和响应做一些转发，有的时候我们可以直接把我们的 node.js 做为最终的服务器，让它直连数据库做一些大型应用。

但是直连数据库会有一些工业级的问题：

- 如何保证我们拼写的 sql 语句是正确的？如何防止出现数据库注入的问题？
- 对于不同的数据库例如 mysql, oracle , 或者相同数据库的不同版本，我如何去兼容它们的语法？

这时候就需要我们的 ORM 框架了，它帮助我们抹平了底层数据库的细节，我们使用这类框架，就能按照它的语法进行书写，最终生成能够应用于各个平台的 sql 语句。

## node.js ORM 框架 sequelize 简介

[sequelize.js](#) 就是一个在 node.js 中优秀的 ORM 框架，使用了 ORM 框架之后，只需要一些简单的学习，即使我们完全没有数据库相关知识，我们也能写好一个数据库应用。

我们只需要在我们的服务端 node.js 项目中，按照要求安装好这个库，在我们的 node.js 中定义下相关的 model 文件。

我们在写入数据了之后，就可以使用相关 model 上的方法，对数据进行增删改查了。