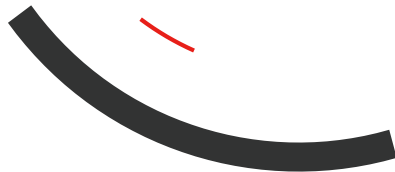




黑马程序员™
www.itheima.com

传智播客旗下
高端IT教育品牌



博客

目录 Contents

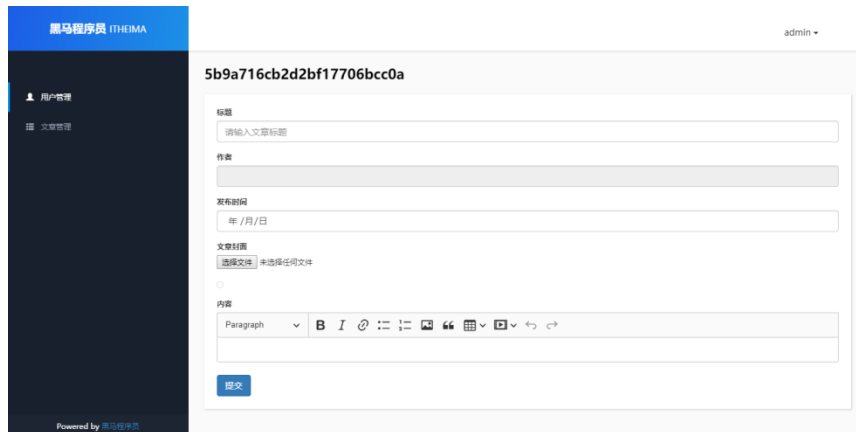
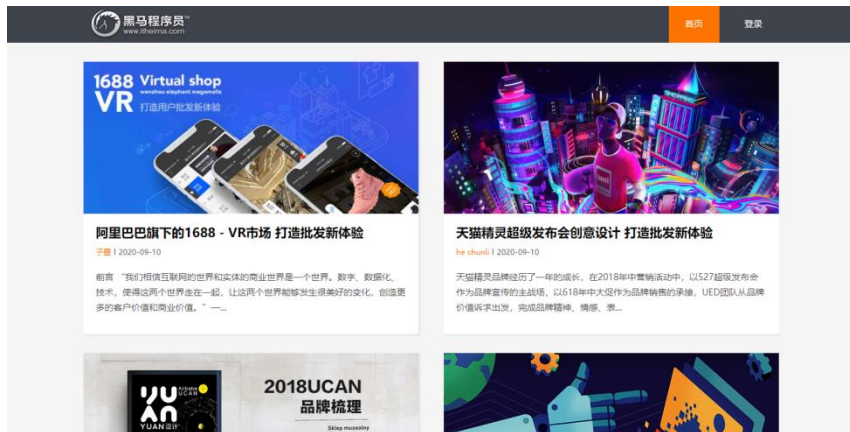
- ◆ 项目环境搭建
- ◆ 项目功能实现
- ◆ 项目包含的知识点

1. 项目环境搭建

1.1 项目介绍

多人博客管理系统

1. 博客内容展示
2. 博客管理功能





1. 项目环境搭建

1.2 案例初始化

1. 建立项目所需文件夹

- public 静态资源
- model 数据库操作
- route 路由
- views 模板

2. 初始化项目描述文件

- npm init -y

3. 下载项目所需第三方模块

```
npm install express mongoose art-template express-art-template
```

4. 创建网站服务器

■ 1. 项目环境搭建

1.2 案例初始化

5. 构建模块化路由

6. 构建博客管理页面模板

■ 2. 项目功能实现

2.1 登录

1. 创建用户集合，初始化用户
 1. 连接数据库
 2. 创建用户集合
 3. 初始化用户
2. 为登录表单设置请求地址、请求方式以及表单项name属性
3. 当用户点击登录按钮时，客户端验证用户是否填写了登录表单
4. 如果其中一项没有输入，阻止表单提交
5. 服务器端接收请求参数，验证用户是否填写了登录表单
6. 如果其中一项没有输入，为客户端做出响应，阻止程序向下执行

黑马程序员 - 博客管理员登录

邮件

密码

登录

■ 2. 项目功能实现

2.1 登录

7. 根据邮箱地址查询用户信息
8. 如果用户不存在，为客户端做出响应，阻止程序向下执行
9. 如果用户存在，将用户名和密码进行比对
10. 比对成功，用户登录成功
11. 比对失败，用户登录失败
12. 保存登录状态
13. 密码加密处理

2. 项目功能实现

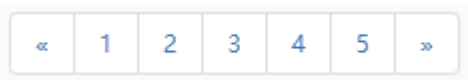
2.2 新增用户

1. 为用户列表页面的新增用户按钮添加链接
2. 添加一个连接对应的路由，在路由处理函数中渲染新增用户模板
3. 为新增用户表单指定请求地址、请求方式、为表单项添加name属性
4. 增加实现添加用户的功能路由
5. 接收到客户端传递过来的请求参数
6. 对请求参数的格式进行验证
7. 验证当前要注册的邮箱地址是否已经注册过
8. 对密码进行加密处理
9. 将用户信息添加到数据库中
10. 重定向页面到用户列表页面

2. 项目功能实现

2.3 数据分页

当数据库中的数据非常多时，数据需要分批次显示，这时就需要用到数据分页功能。



分页功能核心要素：

1. 当前页，用户通过点击上一页或者下一页或者页码产生，客户端通过get参数方式传递到服务器端
2. 总页数，根据总页数判断当前页是否为最后一页，根据判断结果做响应操作

总数据条数	每页显示数据条数	总页数
50	5	10
52	5	11

总页数：Math.ceil (总数据条数 / 每页显示数据条数)

2. 项目功能实现

2.3 数据分页

```
limit(2) // limit 限制查询数量 传入每页显示的数据数量  
skip(1) // skip 跳过多少条数据 传入显示数据的开始位置
```

数据开始查询位置 = (当前页 - 1) * 每页显示的数据条数

■ 2. 项目功能实现

2.4 用户信息修改

1. 将要修改的用户ID传递到服务器端
2. 建立用户信息修改功能对应的路由
3. 接收客户端表单传递过来的请求参数
4. 根据id查询用户信息，并将客户端传递过来的密码和数据库中的密码进行比对
5. 如果比对失败，对客户端做出响应
6. 如果密码对比成功，将用户信息更新到数据库中

■ 2. 项目功能实现

2.5 用户信息删除

1. 在确认删除框中添加隐藏域用以存储要删除用户的ID值
2. 为删除按钮添自定义属性用以存储要删除用户的ID值
3. 为删除按钮添加点击事件，在点击事件处理函数中获取自定义属性中存储的ID值并将ID值存储在表单的隐藏域中
4. 为删除表单添加提交地址以及提交方式
5. 在服务器端建立删除功能路由
6. 接收客户端传递过来的id参数
7. 根据id删除用户

■ 2. 项目功能实现

2.6 开发环境与生产环境

什么是开发环境与生产环境

环境，就是指项目运行的地方，当项目处于开发阶段，项目运行在开发人员的电脑上，项目所处的环境就是开发环境。当项目开发完成以后，要将项目放到真实的网站服务器电脑中运行，项目所处的环境就是生产环境。

为什么要区分开发环境与生产环境

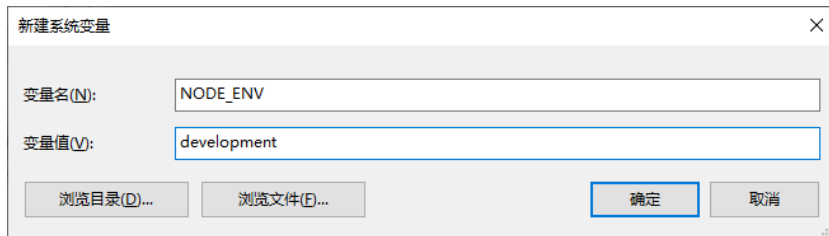
因为在不同的环境中，项目的配置是不一样的，需要在项目代码中判断当前项目运行的环境，根据不同的环境应用不同的项目配置。

2. 项目功能实现

2.6 开发环境与生产环境

如何区分开发环境与生产环境

通过电脑操作系统中的系统环境变量区分当前是开发环境还是生产环境。

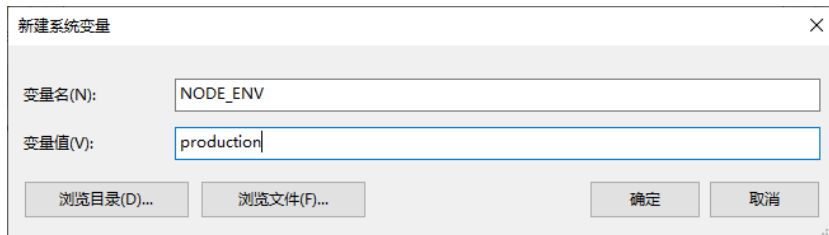


新建系统变量

变量名(N): NODE_ENV

变量值(V): development

浏览目录(D)... 浏览文件(F)... 确定 取消



新建系统变量

变量名(N): NODE_ENV

变量值(V): production

浏览目录(D)... 浏览文件(F)... 确定 取消

2. 项目功能实现

2.6 开发环境与生产环境

如何区分开发环境与生产环境

```
if (process.env.NODE_ENV == 'development') {  
    // 开发环境  
} else {  
    // 生产环境  
}
```

■ 2. 项目功能实现

2.7 第三方模块config

作用：允许开发人员将不同运行环境下的应用配置信息抽离到单独的文件中，模块内部自动判断当前应用的运行环境，并读取对应的配置信息，极大提供应用配置信息的维护成本，避免了当运行环境重复的多次切换时，手动到项目代码中修改配置信息

使用步骤

1. 使用npm install config命令下载模块
2. 在项目的根目录下新建config文件夹
3. 在config文件夹下面新建default.json、development.json、production.json文件
4. 在项目中通过require方法，将模块进行导入
5. 使用模块内部提供的get方法获取配置信息

2. 项目功能实现

2.7 第三方模块config

将敏感配置信息存储在环境变量中

1. 在config文件夹中建立custom-environment-variables.json文件
2. 配置项属性的值填写系统环境变量的名字
3. 项目运行时config模块查找系统环境变量，并读取其值作为当前配置项属于的值

```
{  
  "db": {  
    "pwd": "APP_PWD"  
  }  
}
```

编辑系统变量

变量名(N): APP_PWD

变量值(V): itcast

浏览目录(D)... 浏览文件(F)... 确定 取消

■ 2. 项目功能实现

2.8 文章评论

1. 创建评论集合
2. 判断用户是否登录，如果用户登录，再允许用户提交评论表单
3. 在服务器端创建文章评论功能对应的路由
4. 在路由请求处理函数中接收客户端传递过来的评论信息
5. 将评论信息存储在评论集合中
6. 将页面重定向回文章详情页面
7. 在文章详情页面路由中获取文章评论信息并展示在页面中



3. 项目包含的知识点

3.1 密码加密 bcrypt

哈希加密是单程加密方式: 1234 => abcd

在加密的密码中加入随机字符串可以增加密码被破解的难度。

// 导入bcrypt模块

```
const bcrypt = require('bcrypt');
```

// 生成随机字符串 gen => generate 生成 salt 盐

```
let salt = await bcrypt.genSalt(10);
```

// 使用随机字符串对密码进行加密

```
let pass = await bcrypt.hash('明文密码', salt);
```

// 密码比对

```
let isEqual = await bcrypt.compare('明文密码', '加密密码');
```

3. 项目包含的知识点

3.1 密码加密 bcrypt

bcrypt依赖的其他环境

1. python 2.x

2. node-gyp

npm install -g node-gyp

3. windows-build-tools

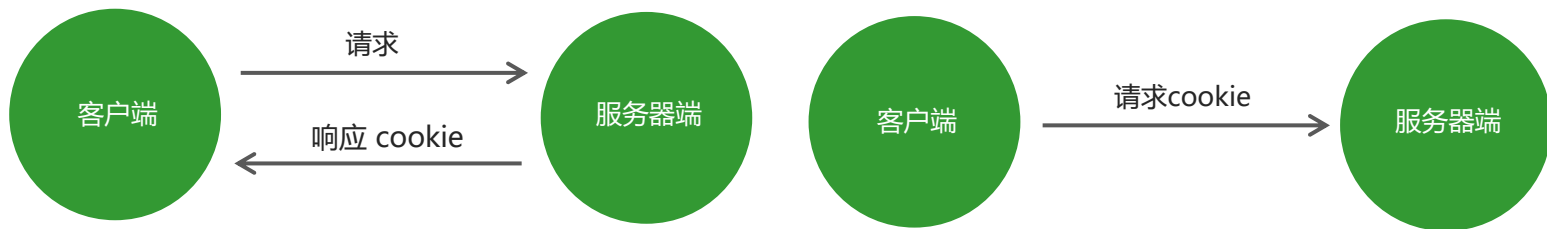
npm install --global --production windows-build-tools

3. 项目包含的知识点

3.2 cookie与session

cookie: 浏览器在电脑硬盘中开辟的一块空间, 主要供服务器端存储数据。

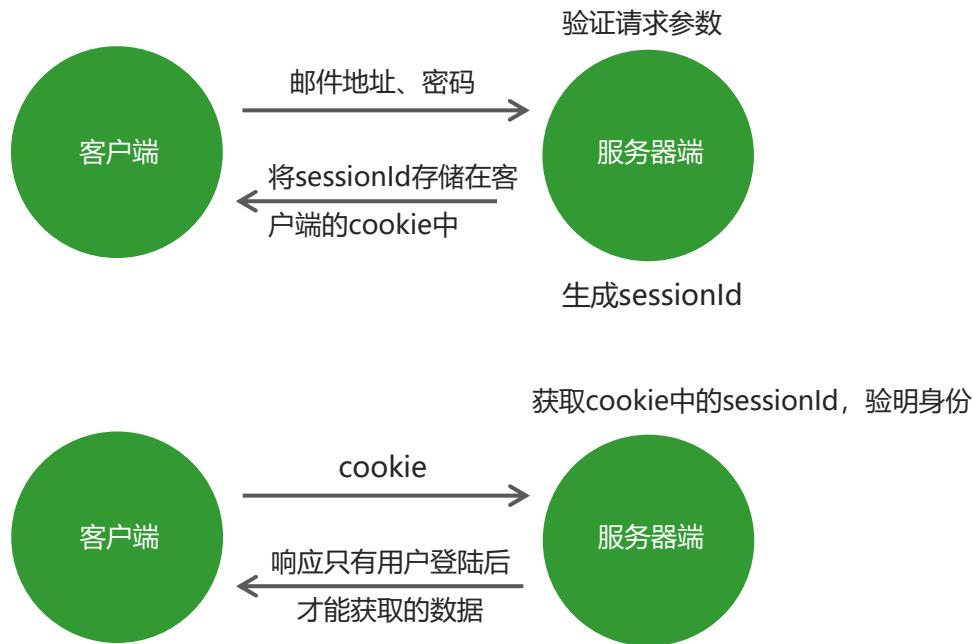
- cookie中的数据是以域名的形式进行区分的。
- cookie中的数据是有过期时间的, 超过时间数据会被浏览器自动删除。
- cookie中的数据会随着请求被自动发送到服务器端。



session: 实际上就是一个对象, 存储在服务器端的内存中, 在session对象中也可以存储多条数据, 每一条数据都有一个sessionid做为唯一标识。

3. 项目包含的知识点

3.2 cookie与session



3. 项目包含的知识点

3.2 cookie与session

在node.js中需要借助express-session实现session功能。

```
const session = require('express-session');  
app.use(session({ secret: 'secret key' }));
```



3. 项目包含的知识点

3.3 Joi

JavaScript对象的规则描述语言和验证器。

```
const Joi = require('joi');
const schema = {
  username: Joi.string().alphanum().min(3).max(30).required().error(new Error('错误信息')),
  password: Joi.string().regex(/^[a-zA-Z0-9]{3,30}$/),
  access_token: [Joi.string(), Joi.number()],
  birthyear: Joi.number().integer().min(1900).max(2013),
  email: Joi.string().email()
};
Joi.validate({ username: 'abc', birthyear: 1994 }, schema);
```




3. 项目包含的知识点

3.4 formidable

作用：解析表单，支持get请求参数，post请求参数、文件上传。

```
// 引入formidable模块
const formidable = require('formidable');

// 创建表单解析对象
const form = new formidable.IncomingForm();

// 设置文件上传路径
form.uploadDir = "/my/dir";

// 是否保留表单上传文件的扩展名
form.keepExtensions = false;

// 对表单进行解析
form.parse(req, (err, fields, files) => {
    // fields 存储普通请求参数
    // files 存储上传的文件信息
});
```

3. 项目包含的知识点

3.5 文件读取 FileReader

```
var reader = new FileReader();  
reader.readAsDataURL('文件');  
reader.onload = function () {  
    console.log(reader.result);  
}
```

140x140

data:image/svg+xml;base64,PD94bWwgdmlVyc2lvbj0iMS4wliBlbmNvZGluZz0iVV
RGLTgilHN0YW5kYWxvbmU9Inllcyl/PjxzdmceG1sbnM9Imh0dHA6Ly93d3cudzM
ub3JnLzlwMDAv3ZnliB3aWR0aD0iMTQwliBoZWlnaHQ9IjE0MCIgdmlld0JveD0i
MCAwIDE0MCAxNDAILHByZXNlcnZlQXNwZWNOUmF0aW89Im5vbmUiPjwhLS0K
U291cmNlIFVSTDogAG9sZGVyLmpzLzE0MHgxNDAKQ3JlYXRIZCB3aXRoiEhvbGRlci
5qcyAyLjYuMCA4KTGVhcm4gbW9yZSBhdCBodHRwOi8vaG9sZGVyanMuY29tCihjK
SAyMDEyLTIwMTUgSXZhbiBNYWxvcGluc2t5IC0gaHR0cDovL2ltc2t5LmNvCi0tPjxk
ZWZzPjxzdhlsZSB0eXBIPSJ0ZXh0L2Nzcyl+PCFbQ0RBVE...

3. 项目包含的知识点

3.6 数据分页 mongoose-sex-page

```
const pagination = require('mongoose-sex-page');  
pagination(集合构造函数).page(1).size(20).display(8).exec();
```



3. 项目包含的知识点

3.7 mongoDB数据库添加账号

1. 以系统管理员的方式运行powershell
2. 连接数据库 mongo
3. 查看数据库 show dbs
4. 切换到admin数据库 use admin
5. 创建超级管理员账户 db.createUser()
6. 切换到blog数据 use blog
7. 创建普通账号 db.createUser()
8. 卸载mongodb服务
 1. 停止服务 net stop mongod
 2. mongod --remove



3. 项目包含的知识点

3.7 mongoDB数据库添加账号

9. 创建mongodb服务

```
mongod --logpath="C:\Program Files\MongoDB\Server\4.1\log\mongod.log" --dbpath="C:\Program  
Files\MongoDB\Server\4.1\data" --install --auth
```

10. 启动mongodb服务 net start mongod

11. 在项目中使用账号连接数据库

```
mongoose.connect('mongodb://user:pass@localhost:port/database')
```



传智播客旗下高端IT教育品牌