

TIP:

- 1、按照 GitHub 上传的规范，此文档应该写在.md 文件中，但是由于实力有限，不能很好的玩转.md 文件格式，因此创作此 PDF
- 2、项目来自于互联网，只是作为学习记录自己学习 koa2 框架的过程与代码
- 3、00koa2 入门--->koa2 的基础入门
- 4、01 完成 koa2 项目--->基于 node 环境下用 koa2 框架完成电商平台后端服务的基础增删改查操作
- 5、菜的一批的，写的不好，多多见谅
- 6、本文档接口（参数）部分首字母均为小写（太懒了省的去改了）

安装&使用

- 1、在文件夹目录下打开命令行窗口，使用 `npm i` 命令下载项目所需要的依赖（需要在 node 环境下）
- 2、在 .env 文件夹中设置本机配置
 - (1) `APP_PORT=3000` //服务器的端口号
 - (2) `MYSQL_HOST = localhost` //使用本地数据库
 - (3) `MYSQL_PORT = 3306` //数据库的端口号
 - (4) `MYSQL_USER = root` //数据库的用户名
 - (5) `MYSQL_PWD = root` //数据库的密码
 - (6) `MYSQL_DB = koa2-item` //数据库的库名
 - (7) `JWT_SECRET=zzc>?!u` //token 加密方式（随便什么都行）
- 3、在 src/model 目录下，使用 `node` 命令，依次运行 model 下的文件（创建数据表）
- 4、src 目录中使用命令行窗口，使用 `node main.js` 运行服务

依赖说明

- 1、"bcryptjs": "^2.4.3",用于对密码进行加密解密
- 2、"dotenv": "^16.0.1",用于管理统一配置文件
- 3、"jsonwebtoken": "^8.5.1",用于生成和解析 token
- 4、"koa": "^2.13.4",koa 框架的核心依赖
- 5、"koa-body": "^5.0.0",解析 body 参数（支持文件）
- 6、"koa-parameter": "^3.0.1",过滤不合格参数（不要相信前端的任何数据）
- 7、"koa-router": "^10.1.1",koa2 路由依赖
- 8、"koa-static": "^5.0.0",用户浏览器直接访问的服务器的静态资源
- 9、"koa2-cors": "^2.0.6",解决跨域问题
- 10、"mysql2": "^2.3.3",数据库驱动（2 能支持 promise）
- 11、"sequelize": "^6.19.2",数据库模型框架（更简单的操作数据库）

接口文档

1、用户操作

基础接口：接口：localhost: 3000/user

1.1 用户注册

接口:post /register

参数：

username:必填， string

password:必填， string

返回项：

注册成功：code: 0， 提示信息

注册失败：失败的提示信息

接口使用要求：

Username 不能与数据库中已经注册的用户重复

1.2 用户登录

接口： post /login

参数：

Username： 必填， string

Password： 必填， string

返回项：

登录成功：提示信息+用户 token

登录失败：具体失败原因

接口使用要求：无

1.3 修改密码

接口： patch /

参数：

Password： 必填， string

返回项：

修改成功：提示信息

修改失败：具体失败原因

接口使用要求：

1、需要先进行登录操作后才能使用此接口

2、新密码与旧密码不能相同

2、地址管理

基础接口：接口：localhost: 3000/addr

2.1 添加收货地址

接口：post /

参数：

Consignee: 收件人, string, 必填

Phone: 手机号, string, 必填, 需要满足手机号规则

Address: 详细地址, string, 必填

返回项：

添加成功：提示信息

添加失败：具体失败原因

接口使用要求：

- 1、需要先进行登录操作后才能使用此接口
- 2、参数需要满足添加规范

2.2 查询个人收货地址

接口：get /

参数：无

返回项：

查询成功：个人收货地址信息

查询失败：失败原因

接口使用要求：

需要先进行登录操作后才能使用此接口

2.3 删除个人收货地址

接口：delete /: id

参数：

id: 必填, int, 使用 params 传递参数

返回项：

删除成功：提示信息

删除失败：具体失败原因

接口使用要求：

- 1、需要先进行登录操作后才能使用此接口
- 2、从数据库永久删除, 注意使用场景

2.4 设置默认地址

接口: patch /:id

参数:

id: 原地址的 id 号, 必填, int, 使用 params 传递参数

返回项:

设置成功: 提示信息

设置失败: 具体失败原因

接口使用要求:

需要先进行登录操作后才能使用此接口

2.5 修改地址

接口: 普通 /:id

参数:

id: 原地址的 id 号, 必填, int, 使用 params 传递参数

Address: 新的收货地址, 与添加收货地址的参数相同

返回项:

修改成功: 提示信息

修改失败: 具体失败原因

接口使用要求:

- 1、需要先进行登录操作后才能使用此接口
- 2、新的信息需要与原来的信息有所变化
- 3、参数需要满足添加规范

3、订单管理

基础接口: localhost:3000/orders

3.1 添加订单信息

接口: post /

参数:

Address_id: 必填, int, 收货地址的 id 号

goods_info 必填, string, 商品信息

Total: 必填, string, 商品总数

返回项:

添加成功: 提示信息

添加失败: 具体失败原因

接口使用要求：

- 1、需要先进行登录操作后才能使用此接口
- 2、订单的状态默认为 0（未支付）
- 3、参数需要满足规范

3.2 打印订单信息

接口：get /

参数：

pageNum，选填，int，默认为 1，订单的第几页

pageSize，选填，int，默认为 10，每一页展示的订单条数

status：选填，int，默认为 0，打印什么状态的订单信息

返回项：

添加成功：提示信息

添加失败：具体失败原因

接口使用要求：

- 1、需要先进行登录操作后才能使用此接口
- 2、参数需要满足规范

3.3 修改订单信息

接口：patch /:id

参数：

id：原订单的 id 号，必填，int，使用 params 传递参数

status：必填填，int，将订单修改为哪种状态

返回项：

修改成功：提示信息

修改失败：具体失败原因

接口使用要求：

- 1、需要先进行登录操作后才能使用此接口
- 2、参数需要满足规范

4、购物车管理

基础接口：localhost:3000/cart

4.1 将商品添加至购物车

接口：post /

参数:

goods_id: 商品 id 号, 必填, int

返回项:

添加成功: 提示信息

添加失败: 具体失败原因

接口使用要求:

需要先进行登录操作后才能使用此接口

4.2 获取购物车列表

接口: get /

参数:

pageNum, 选填, int, 默认为 1, 订单的第几页

pageSize, 选填, int, 默认为 10, 每一页展示的订单条数

返回项:

查询成功: 提示信息

查询失败: 具体失败原因

接口使用要求:

1、需要先进行登录操作后才能使用此接口

2、根据用户的查询请求自适应的填入 num 和 size 参数

4.3 购物车的全选&全不选

接口: get /selectAll & /unselectAll

参数: 无

返回项:

修改成功: 提示信息

修改失败: 具体失败原因

接口使用要求:

需要先进行登录操作后才能使用此接口

4.4 删除购物车列表信息

接口: delete /

参数:

ids: 必填, array[int]

返回项:

删除成功: 提示信息

删除失败: 具体失败原因

接口使用要求:

1、需要先进行登录操作后才能使用此接口

2、参数必须是数组类型

3、从数据库中永久删除，需谨慎

4.5 修改购物车列表信息

接口：patch /:id

参数：

Number ||selected：两者至少选一 number|Boolean

返回项：

修改成功：提示信息

修改失败：具体失败原因

接口使用要求：

- 1、需要先进行登录操作后才能使用此接口
- 2、需要真正修改购物车信息
- 3、参数两者至少选一

5、商品管理

基础接口：localhost:3000/goods

5.1 图片信息上传

接口：post /upload

参数：

file:必填，文件类型

返回项：

添加成功：提示信息

添加失败：具体失败原因

接口使用要求：

- 1、需要先进行登录操作后才能使用此接口
- 2、登录的用户需要有管理员权限
- 3、前端因对上传文件的类型，大小等因素进行限制

5.2 发布商品的接口

接口：post /

参数：

```
goods_name: { type: 'string', required: true },
goods_price: { type: 'number', required: true },
goods_num: { type: 'number', required: true },
```

返回项:

添加成功: 提示信息

添加失败: 具体失败原因

接口使用要求:

- 1、需要先进行登录操作后才能使用此接口
- 2、登录的用户需要有管理员权限

5.3 更新商品信息的接口

接口: `put /:id`

参数:

`id`: 必填, 原商品信息, `int`

`goods_name`: { type: 'string', required: true },

`goods_price`: { type: 'number', required: true },

`goods_num`: { type: 'number', required: true },

返回项:

添加成功: 提示信息

添加失败: 具体失败原因

接口使用要求:

- 1、需要先进行登录操作后才能使用此接口
- 2、登录的用户需要有管理员权限
- 3、修改的信息需要与原来的信息有所不同
- 4、下架（删除）的商品不能进行修改

5.4 获取所有商品信息

接口: `get /`

参数:

`pageNum`, 选填, `int`, 默认为 1, 订单的第几页

`pageSize`, 选填, `int`, 默认为 10, 每一页展示的订单条数

返回项:

查询成功: 提示信息

查询失败: 具体失败原因

接口使用要求:

- 1、需要先进行登录操作后才能使用此接口
- 2、登录的用户需要有管理员权限
- 3、根据用户的查询请求自适应的填入 `num` 和 `size` 参数
- 4、被下架（删除）的商品不进行展示

5.4 上架&下架商品（删除）

接口：get /:id/on & /:id/off

参数：

id:必填，原商品 id,int

返回项：

上架&下架成功：提示信息

上架&下架失败：具体失败原因

接口使用要求：

- 1、需要先进行登录操作后才能使用此接口
- 2、登录的用户需要有管理员权限
- 3、已上架的商品不能进行上架操作，已下架的商品不能进行下架操作
- 4、数据不会从数据库真正删除，而是被隐藏，仍需要提醒用户谨慎操作

PS: tip:

1、把模块化体现到极致，从 router 到 control 到 service；从 app 到 main；从 db 到 seq，原作者应该是一位优秀的 Java 程序员

2、代码非原创，在原作者的基础上进行二次封装

3、仅考虑到简单的接口与逻辑

4、一些中间件的功能

(1) auth, hadAdminPermission(./middleware) 检查是否登录/有管理员权限

(2) Validator 用于存放管理数据的规则限制（不要相信前端传入的任何数据，虽然我走前端但我依然不相信自己传过来的数据）

(3) user.middleware 中是不使用第三方插件的方式，对传入的数据进行管理过滤

5、db 文件夹是数据库连接文件

6、Model 文件夹是每个模块所对应的数据表文件

7、Controller 文件夹是路由文件中的函数，用户展示提示/错误信息，处理数据等

8、Service 文件夹是真正对数据操作的文件

9、Main.js 是入口文件

10、Upload 管理员上传的文件存放与此（可在 app 文件夹更改）

