

一、前言

骑迹 是一款为骑行爱好者设计的全方位移动端应用程序，专注于提升用户的运动体验、社交互动和装备管理能力。其核心价值在于结合骑行记录、社交功能以及二手装备交易，为骑行用户提供一站式服务平台。本文档旨在全面解析 骑迹 的功能模块、技术架构、用户体验设计和运行注意事项，为产品的开发和维护提供全方位的支持。

二、产品整体框架与需求

2.1 产品信息结构图

主要模块结构：

- 首页**：综合展示跑步入口、推荐社区帖子及热门二手交易装备。
- 骑行记录**：记录用户的运动轨迹，包括标准骑行、定时骑行及定距离骑行模式。
- 骑行社区**：用户可通过社区发帖功能分享骑行故事、发起活动及参与话题讨论。
- 二手交易**：用户可发布、搜索及购买骑行相关的二手装备。

信息流结构图说明：

产品整体通过首页串联各主要功能模块，用户从登录后即可进入个性化推荐页面，快速浏览社区动态或二手交易商品，或者直接切换至骑行记录功能完成运动。

2.2 产品功能需求表

功能模块	功能点	优先级	备注
骑行记录	- 标准骑行：记录时间、距离和实时位置轨迹	高	支持实时地图展示和速度分析
	- 定时骑行：用户设定时间目标	高	提供语音提示功能
	- 定距离骑行：用户设定骑行目标距离	高	数据精确到小数点后两位
	- 历史记录与个人成就查看	高	支持分享至社区或第三方平台
骑行社区	- 发帖功能：支持文字、图片及表情发布	高	评论与点赞功能增强用户黏性
	- 活动发布与参与：赛事、排行榜及新闻动态	高	可设置提醒
	- 搜索与关注：搜索骑行主题并关注用户	高	便于用户建立社交圈
二手交易	- 商品分类浏览：分类筛选及排序功能	中	提供价格、评价等排序方式
	- 商品发布与交易：用户可发布商品并完成购买	高	支持物流跟踪与状态更新

2.3 产品用户主流程

- 注册登录：**用户通过手机号或第三方平台（微信、QQ）完成注册或登录。
- 选择功能模块：**进入首页后，用户可选择骑行记录、社区或二手交易模块。
- 骑行记录流程：**用户选择骑行模式，开始记录运动数据，结束后保存并上传至个人主页或社区。
- 社区互动流程：**用户浏览帖子、点赞或评论，也可发布自己的骑行动态或发起活动。
- 二手交易流程：**浏览商品，筛选或搜索心仪装备，完成交易或发布商品信息。

2.4 产品架构

2.4.1 产品架构图

- 前端层：**Android 原生开发，采用 MVVM 模式，支持数据双向绑定；
- 服务层：**后端基于 Spring Boot 提供 RESTful 接口服务；
- 数据层：**MySQL 存储用户及骑行记录，Redis 作为缓存提高访问效率；
- 第三方集成：**集成高德地图 API 用于骑行轨迹记录，支付宝/微信支付用于二手交易功能。

2.4.2 产品功能矩阵

功能模块	目标用户	关键价值
骑行记录	普通骑行爱好者	提供精准的运动记录和分析
社区互动	热衷分享的社交型用户	建立骑行圈子和互动
二手交易	对装备升级有需求的用户	提供低成本的装备获取途径

三、页面 UI 设计与交互

3.1 配色色卡及解释说明

- 主色调：**绿色（#4CAF50）
 - 象征环保与健康，突出骑行运动主题。
- 辅助色：**蓝色（#2196F3）
 - 用于功能提示与交互反馈，增强信息层次感。
- 背景色：**浅灰色（#F5F5F5）
 - 减少视觉疲劳，提供舒适的阅读体验。

示例使用场景：

- 骑行记录页面：绿色用于实时速度和距离显示，蓝色用于按钮和提示。
- 社区页面：浅灰色背景搭配图片动态，提升信息的对比度。

3.2 页面交互设计

骑行记录页面：

- 实时轨迹跟踪，用户可切换不同视图（地图/数据）。
- 骑行完成后自动弹出记录保存窗口，支持一键分享。

社区互动页面：

- 发帖功能支持文字与图片的实时预览，点赞后按钮颜色变化。
- 帖子列表支持左右滑动以切换热门和最新分类。

二手交易页面：

- 商品详情页面包含缩略图滑动功能和用户评价展示。
- 搜索栏动态提示关键词，帮助用户快速定位需求。

四、数据库设计

4.1 数据表总览目录

编号	数据表名称	用途
1	用户表 (user)	存储用户的基本注册信息
2	骑行记录表 (ride)	存储用户的骑行运动数据
3	帖子表 (post)	存储社区用户发布的帖子内容
4	商品表 (product)	存储二手交易商品信息
5	商品分类表 (category)	存储商品的分类信息
6	评论表 (comment)	存储帖子及商品的用户评论
7	订单表 (order)	存储用户购买商品的订单记录
8	支付记录表 (payment)	存储订单支付的详细信息
9	消息通知表 (message)	存储系统或用户之间的消息通知
10	用户关系表 (relation)	存储用户之间的关注和社交关系
11	活动表 (activity)	存储骑行活动信息
12	活动参与表 (participation)	存储用户参与活动的记录
13	用户成就表 (achievement)	存储用户获得的骑行成就
14	举报记录表 (report)	存储用户举报帖子、评论或商品的记录
15	系统日志表 (log)	存储系统操作日志记录
16	广告位管理表 (ad_slot)	管理应用中的广告位及展示内容
17	设备绑定表 (device_binding)	存储用户绑定的设备信息，如码表或传感器
18	路线记录表 (route)	存储用户规划或上传的骑行路线
19	徽章表 (badge)	存储应用内的徽章信息及解锁条件
20	意见反馈表 (feedback)	存储用户提交的意见反馈信息

4.2 数据表详细设计

1. 用户表 (user)

字段	类型	备注
user_id	INT	主键, 自增
username	VARCHAR(50)	用户名
password	VARCHAR(50)	加密后的密码
email	VARCHAR(100)	用户邮箱
phone	VARCHAR(15)	用户手机号
avatar_url	VARCHAR(255)	用户头像 URL 地址
created_at	TIMESTAMP	用户注册时间

2. 骑行记录表 (ride)

字段	类型	备注
ride_id	INT	主键, 自增
user_id	INT	外键, 关联用户表
distance	FLOAT	骑行距离 (公里)
duration	TIME	骑行时长
start_time	DATETIME	骑行开始时间
route_map	TEXT	保存轨迹的地图数据 (JSON)

3. 帖子表 (post)

字段	类型	备注
post_id	INT	主键, 自增
user_id	INT	外键, 关联用户表
title	VARCHAR(100)	帖子标题
content	TEXT	帖子内容
image_url	TEXT	帖子图片 URL (多图 JSON)
created_at	TIMESTAMP	发帖时间

4. 商品表 (product)

字段	类型	备注
product_id	INT	主键, 自增
user_id	INT	外键, 关联用户表
category_id	INT	外键, 关联分类表
name	VARCHAR(100)	商品名称
price	DECIMAL(10, 2)	商品价格
stock	INT	商品库存数量
description	TEXT	商品描述
image_url	TEXT	商品图片 URL (多图 JSON)
is_sold	BOOLEAN	是否已售出 (默认 FALSE)
created_at	TIMESTAMP	商品上架时间

5. 商品分类表 (category)

字段	类型	备注
category_id	INT	主键, 自增
name	VARCHAR(50)	分类名称
description	TEXT	分类描述

6. 评论表 (comment)

字段	类型	备注
comment_id	INT	主键, 自增
post_id	INT	外键, 关联帖子表
product_id	INT	外键, 关联商品表
user_id	INT	外键, 关联用户表
content	TEXT	评论内容
created_at	TIMESTAMP	评论时间

7. 订单表 (order)

字段	类型	备注
order_id	INT	主键, 自增
user_id	INT	外键, 关联用户表
product_id	INT	外键, 关联商品表
quantity	INT	购买数量
total_price	DECIMAL(10, 2)	订单总金额
order_status	VARCHAR(20)	订单状态
created_at	TIMESTAMP	下单时间

8. 支付记录表 (payment)

字段	类型	备注
payment_id	INT	主键, 自增
order_id	INT	外键, 关联订单表
payment_method	VARCHAR(20)	支付方式
payment_status	VARCHAR(20)	支付状态
amount	DECIMAL(10, 2)	支付金额
paid_at	TIMESTAMP	支付时间

9. 消息通知表 (message)

字段	类型	备注
message_id	INT	主键, 自增
user_id	INT	外键, 关联用户表 user_id
content	TEXT	消息内容
type	VARCHAR(20)	消息类型 (系统、用户消息等)
is_read	BOOLEAN	是否已读 (默认 FALSE)
created_at	TIMESTAMP	消息发送时间

10. 用户关系表 (relation)

字段	类型	备注
relation_id	INT	主键, 自增
follower_id	INT	外键, 关联用户表 user_id
followed_id	INT	外键, 关联用户表 user_id
created_at	TIMESTAMP	关系建立时间

11. 活动表 (activity)

字段	类型	备注
activity_id	INT	主键, 自增
name	VARCHAR(100)	活动名称
description	TEXT	活动简介
start_time	DATETIME	活动开始时间
end_time	DATETIME	活动结束时间
location	VARCHAR(255)	活动地点

12. 活动参与表 (participation)

字段	类型	备注
participation_id	INT	主键, 自增
activity_id	INT	外键, 关联活动表 activity_id
user_id	INT	外键, 关联用户表 user_id
status	VARCHAR(20)	参与状态 (已报名、已完成)

13. 用户成就表 (achievement)

字段	类型	备注
achievement_id	INT	主键, 自增
user_id	INT	外键, 关联用户表 user_id
title	VARCHAR(100)	成就标题
description	TEXT	成就描述
obtained_at	TIMESTAMP	获得时间

14. 举报记录表 (report)

字段	类型	备注
report_id	INT	主键, 自增
reporter_id	INT	外键, 关联用户表 user_id
target_id	INT	被举报对象的 ID (如帖子、评论或商品)
target_type	VARCHAR(20)	举报对象类型 (帖子、评论等)
reason	TEXT	举报原因
created_at	TIMESTAMP	举报时间

15. 系统日志表 (log)

字段	类型	备注
log_id	INT	主键, 自增
user_id	INT	外键, 关联用户表 user_id
action	VARCHAR(100)	用户操作描述
created_at	TIMESTAMP	操作时间

16. 广告位管理表 (ad_slot)

字段	类型	备注
ad_slot_id	INT	主键, 自增
position	VARCHAR(50)	广告位位置 (如首页顶部)
content_url	VARCHAR(255)	广告内容 URL 地址
start_time	TIMESTAMP	广告投放开始时间
end_time	TIMESTAMP	广告投放结束时间

17. 设备绑定表 (device_binding)

字段	类型	备注
binding_id	INT	主键, 自增
user_id	INT	外键, 关联用户表 user_id
device_type	VARCHAR(50)	设备类型 (码表、传感器等)
device_name	VARCHAR(100)	设备名称

字段	类型	备注
serial_number	VARCHAR(100)	设备序列号
binding_time	TIMESTAMP	绑定时间

18. 路线记录表 (route)

字段	类型	备注
route_id	INT	主键, 自增
user_id	INT	外键, 关联用户表 user_id
route_name	VARCHAR(100)	路线名称
start_location	VARCHAR(255)	起点坐标
end_location	VARCHAR(255)	终点坐标
distance	FLOAT	路线长度 (公里)
created_at	TIMESTAMP	路线创建时间

19. 徽章表 (badge)

字段	类型	备注
badge_id	INT	主键, 自增
name	VARCHAR(100)	徽章名称
description	TEXT	徽章描述
unlock_condition	TEXT	解锁条件描述
icon_url	VARCHAR(255)	徽章图标 URL 地址

20. 意见反馈表 (feedback)

字段	类型	备注
feedback_id	INT	主键, 自增
user_id	INT	外键, 关联用户表 user_id
content	TEXT	用户反馈内容
feedback_type	VARCHAR(50)	反馈类型 (Bug、建议等)
created_at	TIMESTAMP	提交时间
is_resolved	BOOLEAN	是否已解决 (默认 FALSE)

五、接口档案与代码管理（重构版）

1. 用户表 (user)

函数名	方法	路径	函数说明	参数说明
registerUser	POST	/api/v1/user/register	注册新用户	username (String): 用户名, password (String): 密码, email (String): 邮箱
loginUser	POST	/api/v1/user/login	用户登录	username (String): 用户名, password (String): 密码
getUserById	GET	/api/v1/user/{id}	根据用户ID获取信息	id (Integer): 用户唯一标识
updateUserProfile	PUT	/api/v1/user/{id}/update	更新用户信息	username (String), email (String), phone (String), profileImage (String): 头像路径

2. 骑行记录表 (ride)

函数名	方法	路径	函数说明	参数说明
createRideRecord	POST	/api/v1/ride/create	新建骑行记录	userId (Integer): 用户ID, distance (Float): 骑行距离, duration (Time): 骑行时长

函数名	方法	路径	函数说明	参数说明
<code>getRideRecordById</code>	GET	<code>/api/v1/ride/{id}</code>	根据记录 ID 查询	<code>id</code> (Integer): 骑行记录 ID
<code>listRideRecordsByUser</code>	GET	<code>/api/v1/ride/user/{userId}</code>	获取用户骑行记录	<code>userId</code> (Integer): 用户 ID
<code>deleteRideRecord</code>	DELETE	<code>/api/v1/ride/{id}</code>	删除骑行记录	<code>id</code> (Integer): 骑行记录 ID

3. 帖子表 (post)

函数名	方法	路径	函数说明	参数说明
<code>createPost</code>	POST	<code>/api/v1/post/create</code>	新建帖子	<code>userId</code> (Integer): 用户 ID, <code>content</code> (String): 帖子内容, <code>images</code> (List): 图片列表
<code>getPostById</code>	GET	<code>/api/v1/post/{id}</code>	获取帖子详情	<code>id</code> (Integer): 帖子 ID

函数名	方法	路径	函数说明	参数说明
<code>listPosts</code>	GET	<code>/api/v1/post/list</code>	获取所有帖子	<code>page</code> (Integer): 页码, <code>size</code> (Integer): 每页条数
<code>deletePost</code>	DELETE	<code>/api/v1/post/{id}</code>	删除帖子	<code>id</code> (Integer): 帖子 ID

4. 商品表 (product)

函数名	方法	路径	函数说明	参数说明
<code>addProduct</code>	POST	<code>/api/v1/product/add</code>	添加商品	<code>userId</code> (Integer): 用户 ID, <code>name</code> (String): 商品名称, <code>price</code> (Float): 价格, <code>categoryId</code> (Integer): 类别 ID
<code>searchProducts</code>	GET	<code>/api/v1/product/search</code>	搜索商品	<code>keyword</code> (String): 搜索关键字, <code>categoryId</code> (Integer): 类别 ID, <code>minPrice</code> (Float): 最低价格
<code>getProductDetails</code>	GET	<code>/api/v1/product/{id}</code>	获取商品详情	<code>id</code> (Integer): 商品 ID
<code>deleteProduct</code>	DELETE	<code>/api/v1/product/{id}</code>	删除商品	<code>id</code> (Integer): 商品 ID

5. 分类表 (category)

函数名	方法	路径	函数说明	参数说明
<code>listCategories</code>	GET	<code>/api/v1/category/list</code>	获取分类列表	无
<code>createCategory</code>	POST	<code>/api/v1/category/create</code>	添加新分类	<code>name</code> (String): 分类名称, <code>description</code> (String): 分类描述
<code>deleteCategory</code>	DELETE	<code>/api/v1/category/{id}</code>	删除分类	<code>id</code> (Integer): 分类 ID

6. 评论表 (comment)

函数名	方法	路径	函数说明	参数说明
<code>addComment</code>	POST	<code>/api/v1/comment/add</code>	添加评论	<code>postId</code> (Integer): 帖子 ID, <code>userId</code> (Integer): 用户 ID, <code>content</code> (String): 评论内容
<code>getCommentsByPostId</code>	GET	<code>/api/v1/comment/{postId}</code>	获取帖子评论	<code>postId</code> (Integer): 帖子 ID

函数名	方法	路径	函数说明	参数说明
<code>deleteComment</code>	DELETE	/api/v1/comment/{id}	删除评论	<code>id</code> (Integer): 评论 ID

7. 消息表 (message)

函数名	方法	路径	函数说明	参数说明
<code>sendMessage</code>	POST	/api/v1/message/send	发送私信	<code>senderId</code> (Integer): 发送者用户 ID, <code>receiverId</code> (Integer): 接收者用户 ID, <code>content</code> (String): 消息内容
<code>getMessagesByUserId</code>	GET	/api/v1/message/{userId}	获取用户消息	<code>userId</code> (Integer): 用户 ID
<code>deleteMessage</code>	DELETE	/api/v1/message/{id}	删除消息	<code>id</code> (Integer): 消息 ID

8. 交易记录表 (transaction)

函数名	方法	路径	函数说明	参数说明
<code>createTransaction</code>	POST	/api/v1/transaction/create	创建交易记录	<code>userId</code> (Integer): 用户 ID, <code>productId</code> (Integer): 商品 ID, <code>price</code> (Float): 商品价格

函数名	方法	路径	函数说明	参数说明
getTransactionById	GET	/api/v1/transaction/{id}	获取交易详情	<code>id</code> (Integer): 交易 ID
<code>listUserTransactions</code>	GET	/api/v1/transaction/user/{userId}	获取用户交易记录	<code>userId</code> (Integer): 用户 ID

9. 活动表 (event)

函数名	方法	路径	函数说明	参数说明
<code>createEvent</code>	POST	/api/v1/event/create	创建活动	<code>title</code> (String): 活动标题, <code>description</code> (String): 活动描述, <code>startTime</code> (DateTime): 活动开始时间
<code>getEventById</code>	GET	/api/v1/event/{id}	获取活动详情	<code>id</code> (Integer): 活动 ID
<code>listEvents</code>	GET	/api/v1/event/list	获取活动列表	<code>page</code> (Integer): 页码, <code>size</code> (Integer): 每页条数
<code>deleteEvent</code>	DELETE	/api/v1/event/{id}	删除活动	<code>id</code> (Integer): 活动 ID

10. 活动报名表 (event_registration)

函数名	方法	路径	函数说明	参数说明
registerForEvent	POST	/api/v1/event/register	报名参加活动	userId (Integer): 用户 ID, eventId (Integer): 活动 ID
getRegistrationsByUser	GET	/api/v1/event/registrations/{userId}	获取用户报名的所有活动	userId (Integer): 用户 ID
getRegistrationsByEvent	GET	/api/v1/event/registrations/{eventId}	获取某个活动的所有报名	eventId (Integer): 活动 ID
cancelRegistration	DELETE	/api/v1/event/registration/{id}	取消活动报名	id (Integer): 报名记录 ID

11. 设备表 (device)

函数名	方法	路径	函数说明	参数说明
addDevice	POST	/api/v1/device/add	添加设备	userId (Integer): 用户 ID, deviceType (String): 设备类型, deviceName (String): 设备名称
getDevicesByUserId	GET	/api/v1/device/{userId}	获取用户所有设备	userId (Integer): 用户 ID
deleteDevice	DELETE	/api/v1/device/{id}	删除设备	id (Integer): 设备 ID

12. 购物车表 (cart)

函数名	方法	路径	函数说明	参数说明
<code>addToCart</code>	POST	<code>/api/v1/cart/add</code>	将商品添加到购物车	<code>userId</code> (Integer): 用户 ID, <code>productId</code> (Integer): 商品 ID
<code>getCartByUserId</code>	GET	<code>/api/v1/cart/{userId}</code>	获取用户购物车中的商品	<code>userId</code> (Integer): 用户 ID
<code>deleteFromCart</code>	DELETE	<code>/api/v1/cart/{id}</code>	从购物车中删除商品	<code>id</code> (Integer): 购物车记录 ID

13. 支付表 (payment)

函数名	方法	路径	函数说明	参数说明
<code>createPayment</code>	POST	<code>/api/v1/payment/create</code>	创建支付订单	<code>userId</code> (Integer): 用户 ID, <code>amount</code> (Float): 支付金额, <code>paymentMethod</code> (String): 支付方式
<code>getPaymentById</code>	GET	<code>/api/v1/payment/{id}</code>	获取支付订单详情	<code>id</code> (Integer): 支付订单 ID

函数名	方法	路径	函数说明	参数说明
<code>getPaymentsByUserId</code>	GET	<code>/api/v1/payment/user/{userId}</code>	获取用户的所有支付订单	<code>userId</code> (Integer): 用户 ID

14. 反馈表 (feedback)

函数名	方法	路径	函数说明	参数说明
<code>submitFeedback</code>	POST	<code>/api/v1/feedback/submit</code>	提交用户反馈	<code>userId</code> (Integer): 用户 ID, <code>content</code> (String): 反馈内容, <code>rating</code> (Integer): 评分
<code>getFeedbackById</code>	GET	<code>/api/v1/feedback/{id}</code>	获取反馈详情	<code>id</code> (Integer): 反馈 ID

15. 推广表 (promotion)

函数名	方法	路径	函数说明	参数说明
-----	----	----	------	------

函数名	方法	路径	函数说明	参数说明
<code>createPromotion</code>	POST	<code>/api/v1/promotion/create</code>	创建推广活动	<code>title</code> (String): 活动标题, <code>description</code> (String): 活动描述, <code>startDate</code> (DateTime): 活动开始时间
<code>getPromotionById</code>	GET	<code>/api/v1/promotion/{id}</code>	获取推广活动详情	<code>id</code> (Integer): 推广活动 ID
<code>listPromotions</code>	GET	<code>/api/v1/promotion/list</code>	获取所有推广活动	<code>page</code> (Integer): 页码, <code>size</code> (Integer): 每页条数

16. 折扣表 (discount)

函数名	方法	路径	函数说明	参数说明
<code>createDiscount</code>	POST	<code>/api/v1/discount/create</code>	创建折扣规则	<code>name</code> (String): 折扣名称, <code>rate</code> (Float): 折扣比例, <code>validity</code> (DateTime): 有效期

函数名	方法	路径	函数说明	参数说明
<code>getDiscountById</code>	GET	<code>/api/v1/discount/{id}</code>	获取折扣详情	<code>id</code> (Integer): 折扣 ID
<code>getAllDiscounts</code>	GET	<code>/api/v1/discount/all</code>	获取所有折扣规则	<code>page</code> (Integer): 页码, <code>size</code> (Integer): 每页条数

17. 订单表 (order)

函数名	方法	路径	函数说明	参数说明
<code>createOrder</code>	POST	<code>/api/v1/order/create</code>	创建订单	<code>userId</code> (Integer): 用户 ID, <code>products</code> (List): 商品列表, <code>totalAmount</code> (Float): 总金额
<code>getOrderById</code>	GET	<code>/api/v1/order/{id}</code>	获取订单详情	<code>id</code> (Integer): 订单 ID
<code>getOrdersByUserId</code>	GET	<code>/api/v1/order/user/{userId}</code>	获取用户所有订单	<code>userId</code> (Integer): 用户 ID

函数名	方法	路径	函数说明	参数说明
<code>cancelOrder</code>	DELETE	<code>/api/v1/order/{id}</code>	取消订单	<code>id</code> (Integer): 订单 ID

18. 消息表 (message)

函数名	方法	路径	函数说明	参数说明
<code>sendMessage</code>	POST	<code>/api/v1/message/send</code>	发送消息	<code>senderId</code> (Integer): 发送者 ID, <code>receiverId</code> (Integer): 接收者 ID, <code>content</code> (String): 消息内容
<code>getMessageById</code>	GET	<code>/api/v1/message/{id}</code>	获取消息详情	<code>id</code> (Integer): 消息 ID
<code>getMessagesByUserId</code>	GET	<code>/api/v1/message/user/{userId}</code>	获取用户所有消息	<code>userId</code> (Integer): 用户 ID

19. 邀请码表 (invitation_code)

函数名	方法	路径	函数说明	参数说明
<code>generateInvitationCode</code>	POST	<code>/api/v1/invitation/generate</code>	生成邀请码	<code>userId</code> (Integer): 用户 ID, <code>validity</code> (DateTime): 有效期

函数名	方法	路径	函数说明	参数说明
<code>validateInvitationCode</code>	POST	<code>/api/v1/invitation/validate</code>	验证邀请码	<code>code</code> (String): 邀请码, <code>userId</code> (Integer): 用户 ID

20. 地址表 (address)

函数名	方法	路径	函数说明	参数说明
<code>addAddress</code>	POST	<code>/api/v1/address/add</code>	添加用户地址	<code>userId</code> (Integer): 用户 ID, <code>address</code> (String): 地址详情, <code>isDefault</code> (Boolean): 是否为默认地址
<code>getAddressesByUserId</code>	GET	<code>/api/v1/address/{userId}</code>	获取用户所有地址	<code>userId</code> (Integer): 用户 ID
<code>deleteAddress</code>	DELETE	<code>/api/v1/address/{id}</code>	删除地址	<code>id</code> (Integer): 地址 ID
<code>updateAddress</code>	PUT	<code>/api/v1/address/update</code>	更新地址信息	<code>id</code> (Integer): 地址 ID, <code>address</code> (String): 新地址详情, <code>isDefault</code> (Boolean): 是否为默认地址

六、运行项目注意事项

1. 环境要求:

- vue
- CSS
- JS

- MySQL 数据库版本 8.0+

2. 配置要求:

- 地图功能需配置高德 API Key, 访问限制建议设为 IP 白名单。
- 支付功能需配置支付宝和微信支付的回调 URL, 确保服务器 SSL 证书正常。

3. 性能优化:

- 后端接口尽量减少复杂查询, 使用索引优化查询速度。
- 前端加载时可引入 Lottie 动画提高加载体验。

七、测试方案

7.1 测试范围

- **功能测试:** 全面覆盖骑行记录、社区发帖、商品交易功能。
- **性能测试:** 对高并发访问的二手交易模块进行压力测试。
- **安全测试:** 确保用户数据加密传输和支付接口的安全性。

7.2 测试流程

1. **单元测试:** 对每个功能模块进行独立测试。
2. **集成测试:** 模拟用户主流程的实际操作, 确保功能无缝衔接。
3. **压力测试:** 通过模拟高并发场景测试系统的稳定性。