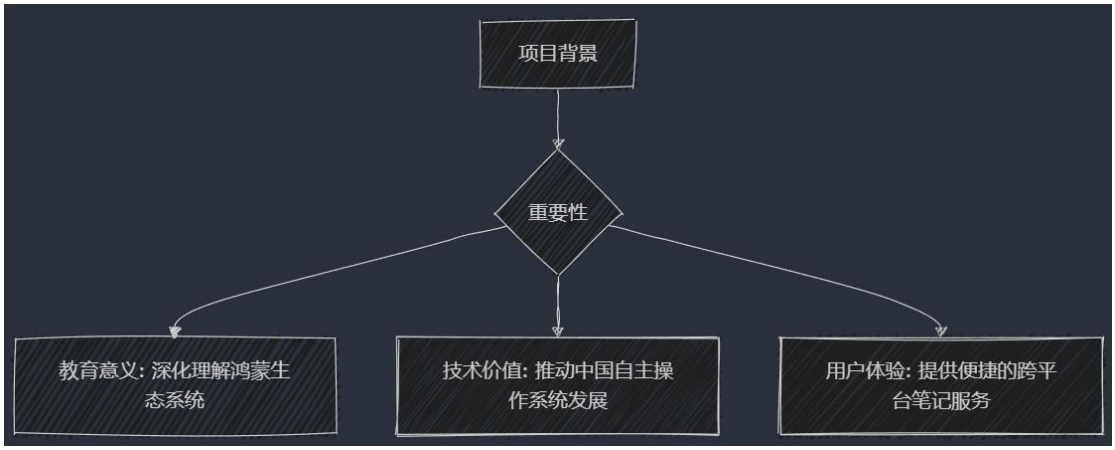
目录

第一章 概述

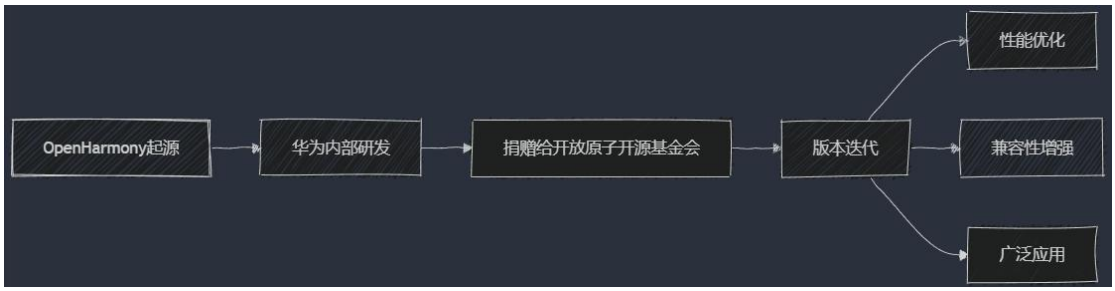
1.1 项目背景与意义



图表 1 项目背景

MyNote 应用的开发旨在填补当前市场中轻量级、高效笔记管理工具的空缺，尤其针对 HarmonyOS 和 OpenHarmony 系统用户。该项目不仅为用户提供了一个便捷的跨平台笔记服务，同时也促进了开发者对鸿蒙生态系统的理解和掌握，对于推动中国自主操作系统的发展有着重要的教育和技术价值。此外，通过使用现代化的开发语言和技术栈（如 ArkTs、Kotlin），以及高效的数据库管理和同步机制，MyNote 在提升用户体验的同时，也展示了如何构建高性能的应用程序。

1.2 OpenHarmony 发展历史

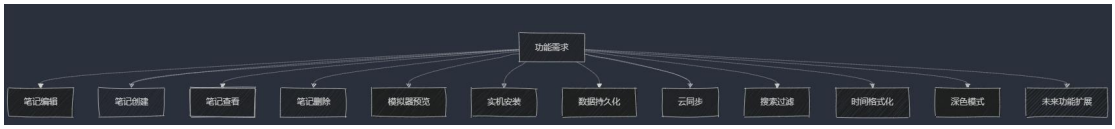


图表 2 OpenHarmony 发展历史

OpenHarmony 的发展历程可以追溯到华为公司内部的操作系统研究。面对复杂的国际环境和技术封锁，华为决定自主研发一个能够支持多种设备类型的分布式操作系统。2020 年 9 月，华为将 OpenHarmony 捐赠给开放原子开源基金会，标志着该操作系统正式成为全球开发者共同维护的开源项目。自此之后，OpenHarmony 经历了多个版本迭代，不断优化其性能和兼容性，逐步成长为一个成熟的全场景智能终端操作系统，广泛应用于智能家居、物联网等多个领域。

1.3 应用场景和需求

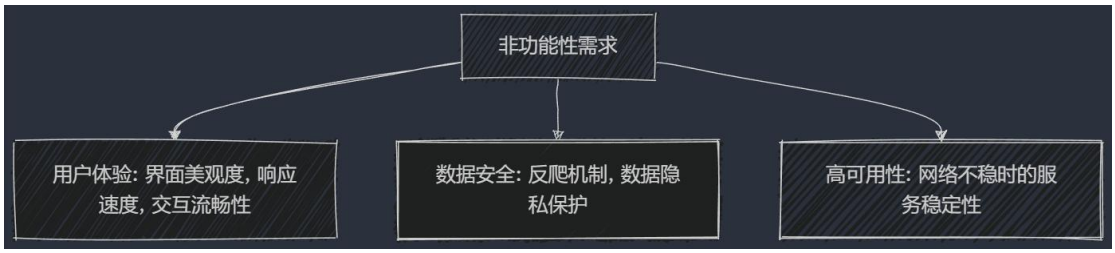
1.3.1 功能需求



图表 3 功能需求

MyNote 应用的功能需求涵盖了从基础的笔记编辑、创建、查看、删除等操作，到高级的数据持久化和云同步功能。它不仅需要满足用户的日常记录需求，还要确保数据的安全性和可靠性。为了提高用户体验，MyNote 还实现了搜索过滤功能，让用户可以通过关键词快速查找所需内容；时间格式化功能则让新建或更新的时间显示更加人性化。同时，考虑到不同用户群体的需求差异，MyNote 支持深色模式切换，并且计划在未来版本中加入附件添加、字体格式调整等功能，以提供更丰富的编辑体验。

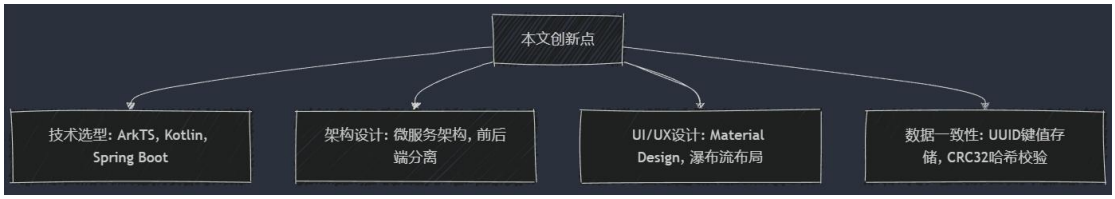
1.3.2 非功能性需求



图表 4 非功能需求

非功能性需求主要包括三个方面：一是用户体验，包括界面美观度、响应速度、交互流畅性等；二是数据安全，例如采用令牌桶算法进行限流控制，防止恶意爬虫攻击，同时保证用户数据的隐私性和完整性；三是高可用性，即确保即使在网络不稳定的情况下，用户也能正常访问自己的笔记信息。为此，MyNote 采用了 PostgreSQL 作为后端数据库，因其良好的并发处理能力和 ACID 兼容特性而被选中，从而保障了服务的稳定性和可靠性。

1.3 本文创新点

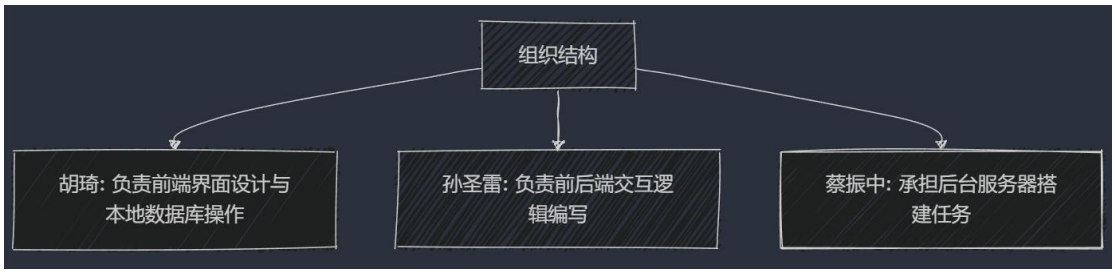


图表 5 本文创新点

本文提出的 MyNote 应用在多个方面展现了创新之处。首先，在技术选型上，选择了新兴的语言和技术栈，如 ArkTS 用于前端开发，Kotlin 用于服务器端开发，结合 Spring

Boot 框架实现高效稳定的后端服务。其次，在架构设计方面，采用了微服务架构，使得前后端分离更加彻底，提高了系统的可扩展性和维护性。再次，UI/UX 设计理念紧跟潮流，参考 Material Design 风格，配合瀑布流布局，使应用既美观又实用。最后，针对云端和本地数据的一致性问题，MyNote 引入了 UUID 键值存储机制，以及 CRC32 哈希校验，确保了数据同步过程中的准确性和效率。

1.4 组织结构



图表 6 组织结构

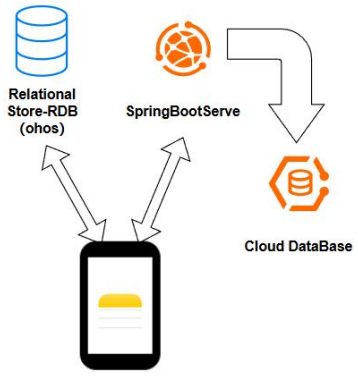
本项目的开发团队由三名成员组成，各自负责不同的模块，形成了紧密协作的工作模式。F主要负责基于 ArkUI 的软件页面开发，包括笔记编辑、创建、用户登录界面的设计与测试，以及利用 SQLite 实现高效的本地数据持久化。S专注于前后端交互逻辑编写，确保本地与远程数据库之间的高效同步。C则承担后台服务器搭建任务，基于 SpringBoot3 使用 Kotlin 语言开发云同步服务器，并选择 PostgreSQL 作为数据库，以提供高性能的并发处理能力。三人分工明确，各司其职，共同完成了整个应用程序的设计与实现，同时也积累了宝贵的团队合作经验。

第 2 章 整体框架及功能模块设计

2.1 概述

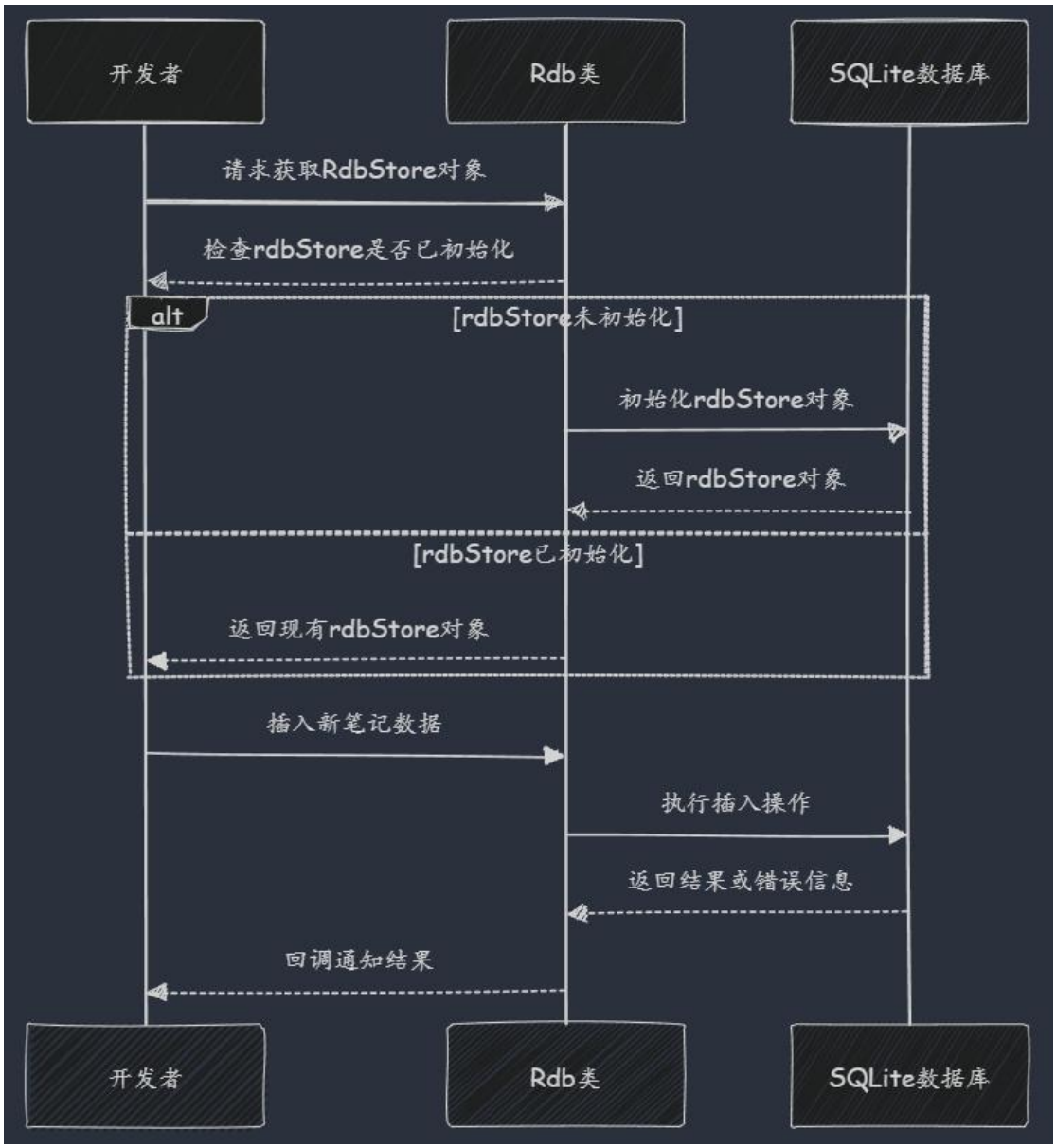
MyNote 是一个轻量级基于 ArkUI，使用鸿蒙生态的应用开发语言 ArkTs，运行在 HarmonyOS、OpenHarmony 系统的软件，参考 md 设计风格。云服务器使用由 kotlin 编写的 java 应用，即兼具了 Java 对各大开源系统的跨平台支持，也通过 Kotlin 所提供的现代化语法糖进行迅捷开发。云服务器使用令牌桶算法进行限流、并且 User、Note 持久化对象键值使用 UUID，具有很好的反爬机制，保证了服务高可用。

2.2 系统整体架构及功能模块设计



图表 7 项目整体架构

2.3 Relational Store（RDB） 设计



图表 8 RDB **设计**

数据库定义

类 Note

id: number - 笔记的唯一标识符。

updateTime: number - 笔记最后更新的时间戳。

title: string - 笔记标题。

content: string - 笔记内容。

RDB 类设计

抽象的 Rdb 类

rdbStore: relationalStore.RdbStore | null - 管理关系数据库的方法属性或为 null。

tableName: string - 数据库表名称。

sqlCreateTable: string - 创建 SQLite 表的语句字符串。

columns: Array<string> - 列名字段的数组。

构造函数

创建表的 SQL 语句、列名字段的数组。

方法 getRdbStore

获取操作数据库的 RdbStore 对象，接收一个回调函数作为参数。此方法检查 rdbStore 是否已初始化；如果没有，则获取上下文并初始化 rdbStore。

插入数据方法 insertData

接受要插入的数据（键值对形式）和一个回调函数作为参数。如果 rdbStore 存在，则执行插入操作，并通过回调传递结果或错误信息。

操作接口与持久化

关系数据库操作接口（RdbStore）

用于创建表、增删改查等基本操作。

谓词（RdbPredicates）

用于过滤查询条件。

结果集（ResultSet）

保存过滤后的查询结果。

2.4 PostgreSQL 设计

2.4.1 PostgreSQL 概述

PostgreSQL 是一种特性非常齐全的自由软件的对象-关系型数据库管理系统（ORDBMS），是以加州大学计算机系开发的 POSTGRES，4.2 版本为基础的对象关系型数据库管理系统。POSTGRES 的许多领先概念只是在比较迟的时候才出现在商业网站数据库中。PostgreSQL 支持大部分的 SQL 标准并且提供了很多其他现代特性，如复杂查询、外键、触发器、视图、事务完整性、多版本并发控制等。同样，PostgreSQL 也可以用许多方法扩展，例如通过增加新的数据类型、函数、操作符、聚集函数、索引方法、过程语言等。另外，因为许可证的灵活，任何人都可以以任何目的免费使用、修改和分发 PostgreSQL。[3]

2.4.2 具体数据库实体设计

```
create table users
(
    id                  uuid not null
        primary key,
    hashed_password varchar(255),
    salt                varchar(6),
    user_name           varchar(255)
        unique
);
```

hashed_password:

数据类型为 varchar(255)，存储用户的哈希密码。
salt:
数据类型为 varchar(6)，存储用于加密密码的盐值。
user_name:
数据类型为 varchar(255)，存储用户的用户名。
具有唯一约束 (unique)，确保每个用户名在整个数据库中都是唯一的。

```
create table notes
(
    created_date            timestamp(6) with time zone,
    last_modified_date timestamp(6) with time zone,
    id                          uuid not null
        primary key,
    user_id                 uuid
        constraint fkechaouoa6kus6k1dpix1u91c
            references users,
    title                       varchar(255),
    content                 oid
);
```

created_date:
数据类型为 timestamp(6) with time zone，记录笔记的创建日期和时间，精度为微秒级别，并带有时区信息。
last_modified_date:
数据类型为 timestamp(6) with time zone，记录笔记最后修改的日期和时间，精度为微秒级别，并带有时区信息。
id:
数据类型为 uuid，是主键 (primary key)，确保每个笔记都有一个唯一的标识符。
user_id:
数据类型为 uuid，表示创建该笔记的用户的 ID。
有一个外键约束 (constraint fkechaouoa6kus6k1dpix1u91c references users)，确保 user_id 必须引用 users 表中存在的 id 值。
title:
数据类型为 varchar(255)，存储笔记的标题。
content:
数据类型为 oid，存储笔记的内容。oid 是 PostgreSQL 中的对象标识符类型，通常用于存储二进制大对象（BLOBs）。

2.5 Springboot Server 设计

2.5.1 Springboot 模块设计

本模块是一个基于 **Kotlin** 和 **Spring Boot 3** 构建的高并发笔记同步服务器，旨在

为多设备用户提供快速、稳定、安全的笔记同步功能。项目采用现代化的技术栈和设计模式，结合高效的缓存机制、灵活的数据存储方案和全面的权限认证功能，确保在高并发场景下提供优异的性能和稳定性。此外，项目集成了开放接口文档和多种开发工具，以提高开发效率和代码可维护性。

2.5.2 详细设计

2.5.2.1 数据库与 ORM

项目采用 PostgreSQL 作为数据库，理由在于其开源、高性能以及对复杂查询的优秀支持，特别适合存储笔记这样结构化数据复杂且需要高可靠性的场景。为简化数据库操作，选择 Spring Data JPA 提供的 ORM 功能，通过实体映射数据库表，开发者可以专注于业务逻辑而无需关注 SQL 细节。同时，Kotlin 的扩展函数和注解功能使得 JPA 的使用更加流畅。

2.5.2.2 缓存与性能优化

为应对笔记同步中可能存在的大量读写操作，项目集成了 Spring Cache 和 Caffeine。Caffeine 是一款高性能的本地缓存库，其基于访问频率和时间的智能驱逐策略，可以有效减少数据库查询压力并提升响应速度。与 Spring Cache 的集成也非常简单，可以通过注解的方式快速实现缓存的启用与配置。

2.5.2.3 权限认证

安全性是笔记同步服务器的关键之一，项目使用 Sa-Token 作为权限认证方案。相比其他认证框架（如 Spring Security），Sa-Token 更轻量化，配置更加灵活，尤其在分布式环境下表现出色。它支持多种认证模式，包括会话管理和 Token 验证，能够满足多种场景下的安全需求。此外，其简单易用的 API 大大降低了开发的复杂度。

2.5.2.4 API 数据校验

为了确保客户端提交的数据合法性，项目引入了 Spring Validation。这一库基于 Bean Validation 规范，可以通过注解的方式轻松定义数据校验规则，从而保证服务端逻辑的健壮性。例如，校验用户输入的笔记标题是否为空或内容长度是否超过限制，都可以通过简单的注解实现，减少了手动校验的冗余代码。

2.5.2.5 开放接口文档

项目集成了 Knife4j 以生成符合 OpenAPI 3 标准的接口文档。Knife4j 是 Swagger 的

扩展版本，提供了更加丰富的功能和友好的用户界面，使开发者和第三方客户端可以方便地查看和测试 API。这不仅提升了开发效率，也便于后续的功能扩展和合作开发。

2.5.2.6 开发工具与库

为了简化开发过程，项目引入了 Hutool 工具库，这是一个功能丰富、简单易用的 Java 工具类库，涵盖了从字符串处理到文件操作的多种功能，显著提升了代码的可读性和开发效率。此外，Jackson 模块用于处理 JSON 数据，与 Kotlin 的数据类完美兼容，为序列化和反序列化提供了便捷的解决方案。

2.5.2.7 构建与依赖管理

项目通过 Gradle 的 Kotlin DSL 进行构建配置，这种方式与 Kotlin 语言本身无缝衔接，语法更加简洁明了。此外，io.spring.dependency-management 插件自动管理 Spring 生态中的依赖版本，减少了手动调整依赖版本可能引入的问题。

第 3 章 核心算法

3.1 CSC32 散列函数

3.1.1 概述

CRC32 是一种校验和/散列算法，常用于内核和互联网校验。 它与 MD5 校验和算法非常相似。

从设置了所有位（0xffffffffffff）的 32 位校验和开始。 这有助于为 "0 "字节的输入字符串提供 0 以外的输出值。 在循环中： 根据下一个输入数据（通常是一个字节）和上一个 CRC 值的低 N 位（N 是所操作数据的大小，通常是 8 位字节），在表格中查找一个 "多项式"（实际上只是一个 32 位值）。

将上一个 32 位 CRC 值向下移动 N 位。 将 "多项式 "与移动后的 CRC 值进行排他-OR，生成一个新值。 循环结束后，将计算出的 CRC 值与 0xffffffff 进行排他-OR（这与对 CRC 值进行二进制 NOT 相同）。 这就是最终的 CRC32 结果。[1]

3.1.2 应用

用于本地和云端数据同步时的完整性验证，确保数据一致性。

使用 CRC32 的主要原因是它在性能、输出长度和存储成本之间取得了良好的平衡。CRC32 计算速度快，适合资源受限的环境，相比复杂哈希算法（如 SHA-256 或 MD5）开销更小。它始终生成一个 32 位的整数，节省存储空间并简化存储结构，同时避免了直接存储 UUID 带来的高存储成本和低索引效率问题。虽然 CRC32 的碰撞概率比复杂

哈希算法高，但在 UUID 已经具有较高唯一性的情况下，碰撞概率仍然很低。

3.2 令牌桶限流算法

3.2.1 描述

令牌桶算法是网络流量整形（Traffic Shaping）和速率限制（Rate Limiting）中最常使用的一种算法。典型情况下，令牌桶算法用来控制发送到网络上的数据的数目，并允许突发数据的发送。[2]

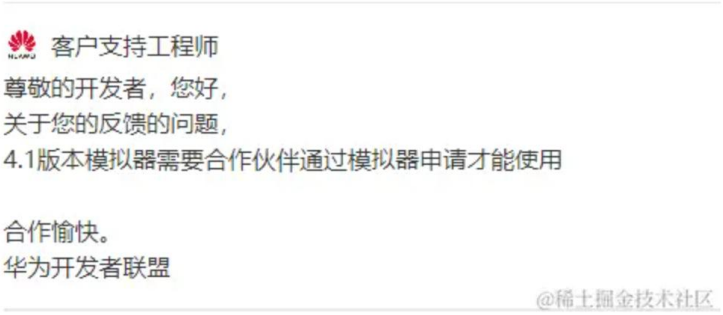
用于云服务器的限流，确保系统在高并发场景下的稳定性。常用于控制系统对资源的访问速度，防止过载。它通过模拟一个装满令牌的桶来工作，每当一个请求到来时，必须从桶中取出一个令牌才能被处理。如果桶中没有令牌，请求将被丢弃或延迟处理，直到有足够的令牌可用

第 4 章 问题与解决方案

4.1 Dev Studio 无法启动模拟器

最终结论：

官方客服回复时间点2024-05-08 17:07:52 GMT+08:00，我估计未来成熟了应该会全面放开



图表 9 dev studio4.1 版本模拟器

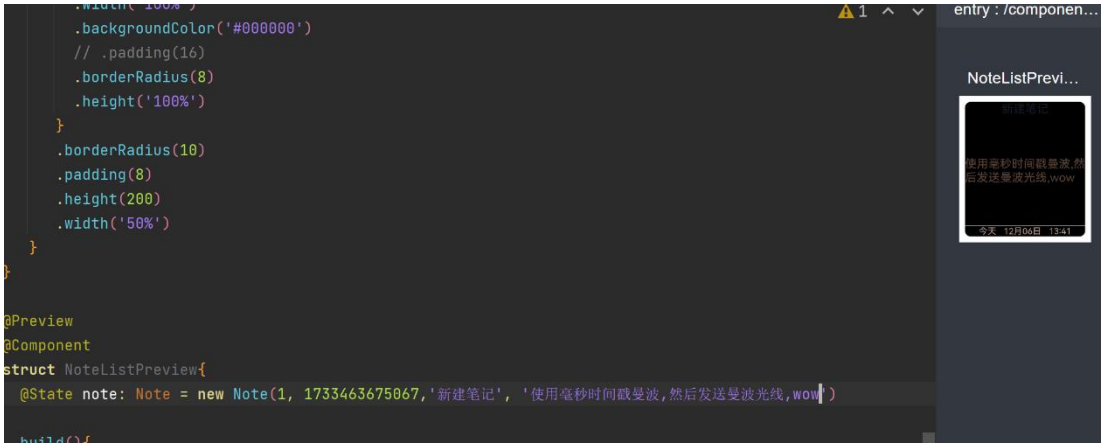
解决方案：鸿蒙 DevEco Studio 4.1 Release-模拟器启动方式错误软件版本：DevEco Studio 4.1 - 掘金。从 4.1 版本换到 5.0，最后换到 4.0 版本。因为 4.1 还不开放模拟器，而 5.0 实机调试只能对 harmony Next 系统设备。

4.2 Api 无法使用类似 uniapp 的 tabBar

解决方案：干脆不用了，因为 api9 版本暂时没有 tabBar 实现，直接用 route 跳转页面。即右上角的 setting 图标跳转 mine.ets。

4.3 笔记组件挤在一块，不好调试

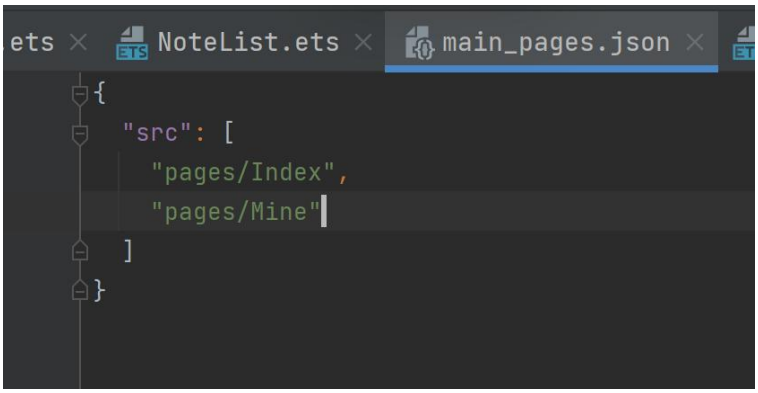
解决方案：将背景颜色设为黑色，然后使用 Dev Studio 预览查看组件结构。如下图所示：最后发现是因为最外层的 Column 没有设置 height('100%')



图表 10 组件问题

4.4 点击 setting 图标，无法跳转 Mine.ets 页面。

解决方案：原来是没有在 src/main/resources/base/profile/main_pages.json 中声明 Mine.ets。声明一下就行。

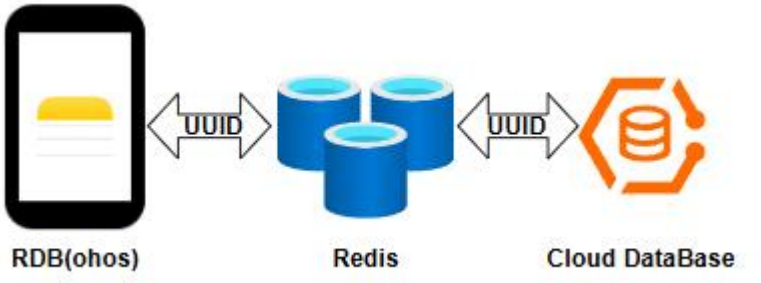


图表 11 页面配置

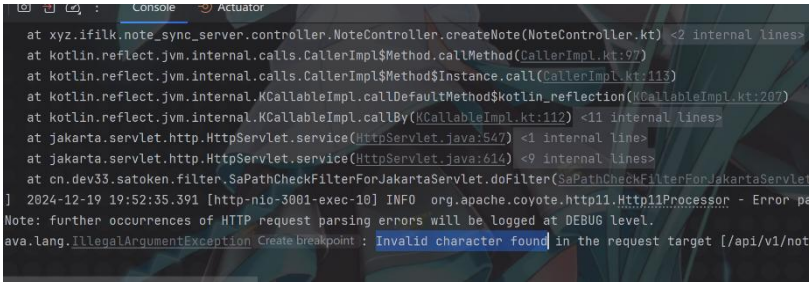
学习 arkui 开发时，尝试申请“获取所有 app”权限，尽管在配置文件中有申请，但是没有效果。

解决方案：由于获取所有 app 权限较高，此处有两种解决方法：(1)换 full sdk。(2)需要修改 sdk 添加 acl 规则。其中方法 2 在尝试过后没生效。

4.4 UUID 路径参数冲突问题



问题描述： UUID 默认携带特殊字符“-”其不能作为 URL 路径参数



图表 12 URL 路径参数包含非法字符导致的解析异常

解决方法：前端传递时先去除“-”，在后端解析为 UUID 时加回“-”

第 5 章 实机测试

运行截图：

App 运行：

登录界面与主页：在页面生命周期里，使用 aboutToAppear 控制登录窗口的显示，然后调用登录方法，登录验证 token 成功后，拉取云服务器的内容传到本地数据库中。如下：

图 4- 1 app 主界面

关于”我的”界面变化：一开始比较潦草，后来使用各种 Row 和 Column 进行排列。几个 button 加上了开发者的头像，横向排列。开源仓库和后面的 icon 分别使用.align(Alignment.Start)和.align(Alignment.End)放置在首尾。

同时还使用了系统能力 '@ohos.pasteboard'与'@ohos.promptAction' ，点击开源仓库的 icon，会粘贴仓库地址并跳出 toast 提示“地址复制成功”。

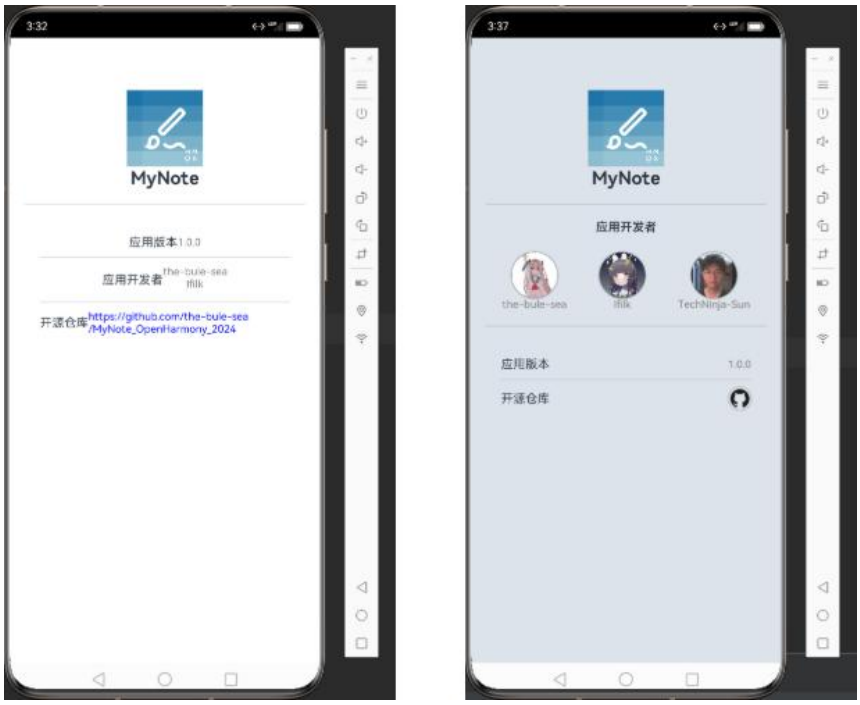


图 4- 2 "我的"界面变迁

编辑、新建笔记弹窗：对于每个 note 组件，都有 editNoteIndex 序号，点击某个后，会显示编辑弹窗，当然点击主页面的新增按钮后就会跳出新建笔记的弹窗。点击保存或者删除或者修改编辑，都会操作本地关系数据库 RDB 的数据内容，实现数据持久化。后面还添加了云同步功能。

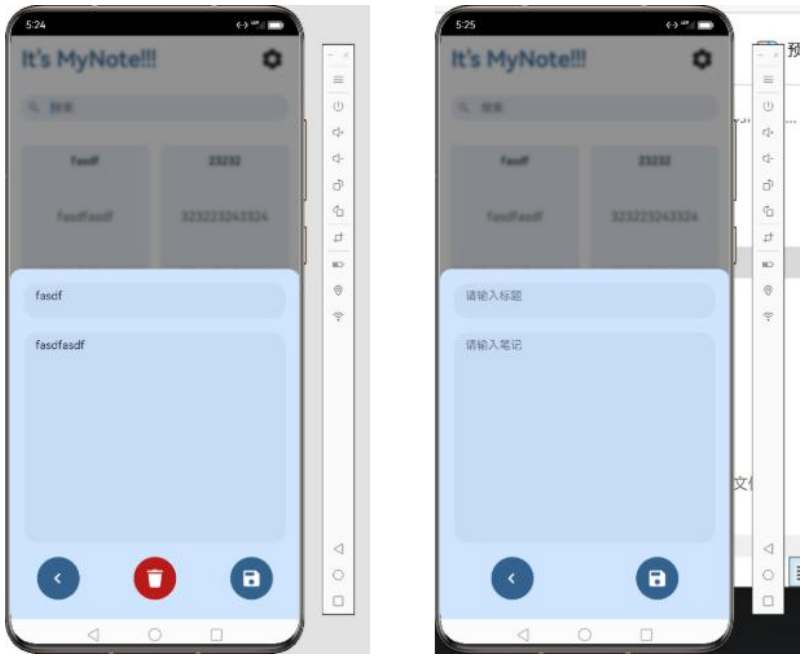


图 4- 3 编辑笔记、新建笔记弹窗

云同步后台运行：

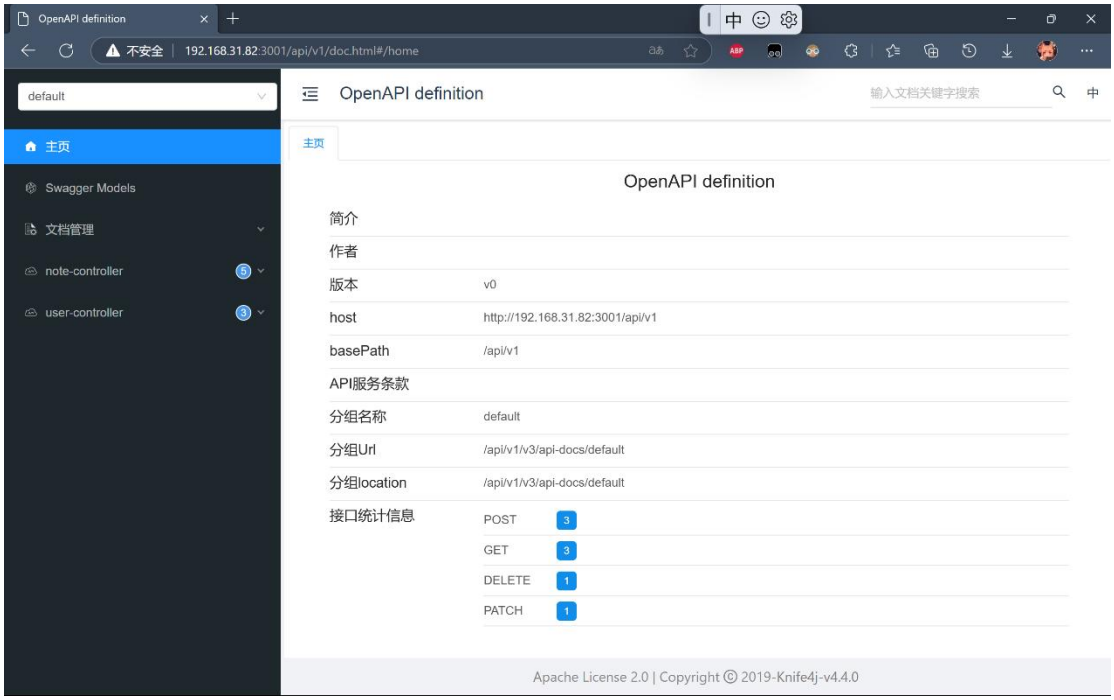


图 4- 4 后台页面

App 图标设计：整体以蓝色为基调，生成渐变图后像素化，然后使用华为官方提供的 HarmonyOS 图标库，加上“ic_screenshot_penshape”。最后，按照上架华为应用商店的要求（应用图标-视觉风格-手机/折叠屏/平板 - 华为 HarmonyOS 开发者），在要求的位置和大小添加区隔元素“HMOS”，如下：

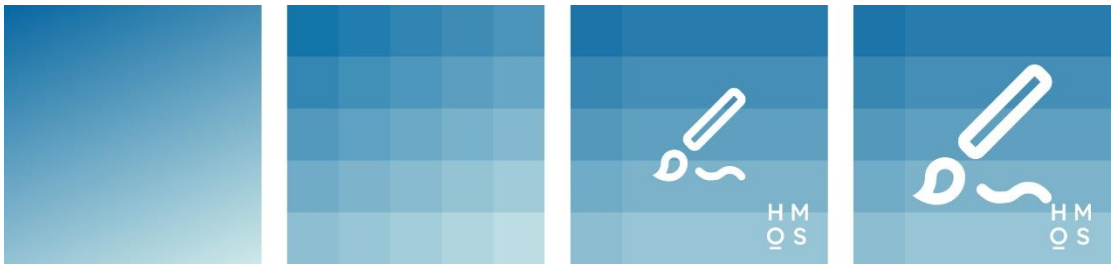


图 4- 5 app 设计

参考文献

[1] CRC32 - OSDev Wiki
[2] 令牌桶算法_百度百科
[3] MySQL 和 PostgreSQL 的对比选择-【维普期刊官网】 - 中文期刊服务平台