



# 20251028\_korean\_word\_learning\_TechSpec\_v1 CN

创建者	z yx
创建时间	@2025年10月28日 16:12
类别	PRD
上次编辑者	z yx
上次更新时间	@2025年10月30日 22:37
状态	完成

## 技术与结构

- 框架：Flutter 3.x
- 数据源：Notion API（用户自己的 internal integration）
- 本地存储：`isar` 或 `sqflite`（任一，其它不依赖服务端）

项目	sqflite	isar
数据模型	关系型（表+SQL）	对象型（类+字段）
底层实现	原生 SQLite	纯 Dart（C++ 核心引擎）
性能	稳定，中等	极快（可数十倍提升）
语法	SQL 语句	链式API
Web支持	无	有
复杂查询	强	中等
易用性	略复杂	简洁

- 安全存储：`flutter_secure_storage`（保存 Notion Token）
- 网络：`dio` + 拦截器（重试、速率限制429处理）
- 路由/状态：`go_router` + `riverpod`
- 构建分发：本地 `flutter build apk`，直接给 APK

## 同步模型（无服务端前提）

- 启动时：用本地缓存渲染 → 后台增量拉取 Notion → 合并到本地
- 写入时：本地先写入草稿 → 调用 Notion API 成功后写“已同步”标记
- 冲突：以 Notion `last_edited_time` 为准；本地有未同步且远端已改时提示用户选择
- 标识：使用 Notion `page_id` 作为主键；本地表含 `updatedAtLocal`，`updatedAtRemote`，`syncState`

## Notion 集成方式

- 单用户安全做法：用户在 Notion 创建 Internal Integration，拿到 `secret_***`，在 App 首次启动粘贴，一次性保存到本机 Keystore。不要把 token 硬编码进 App。
- 授权对象：把目标数据库或页面“Share”给该 Integration。
- 速率限制：每秒 3 请求（保守），分页 `page_size=50`，用 `has_more + next_cursor` 拉全量。

## 数据库设计（示例）

以“任务清单”为例：

Notion 数据库字段： `Title` (title), `Status` (select), `Due` (date), `Tags` (multi\_select), `Notes` (rich\_text)

本地表字段：

```
id (string, notion page_id)
title (string)
status (string)
due (datetime nullable)
tags (string[]) // JSON存
notes (string) // 简文本提取
updatedAtRemote (datetime)
updatedAtLocal (datetime)
syncState (enum: synced | pending | conflict | failed)
```

## 关键页面

- 设置页：输入与测试 Notion Token；选择/绑定数据库（支持粘贴数据库URL，App 解析 database\_id）
- 单词学习页：
  - 列表页：本地渲染，支持筛选与搜索（本地执行，必要时远端搜索）
  - 详情页：创建/编辑；保存后本地落盘并排队同步
- 同步队列视图：显示失败项与重试

## 依赖清单（最小）

```
dio, riverpod, go_router, freezed, json_serializable,
flutter_secure_storage, isar(或 sqflite), connectivity_plus
```

## 打包与分发

- Debug 自测： `flutter run`
- 生成 APK： `flutter build apk --release`
- 分发：直接发 APK 或放网盘；必要时用 `bundletool` 生成 `apks` 做多ABI

## 隐私与安全

- 不收集除必要数据外的信息
- 提供“清除本地缓存与密钥”的入口
- 无远端日志，崩溃仅本地提示（如需日志，写本地文件手动分享）

## 4 周执行表（Android 本地分发）

W1 需求→字段对齐→UI草图→Flutter 工程与本地DB打通→设置页录入 Notion Token

W2 Notion 读：数据库绑定→分页拉取→本地缓存与列表页

W3 Notion 写：新增/修改/删除→冲突策略→离线队列

W4 QA 与边界：速率限制/失败重试/空态错误态→生成 APK→交付

## 你现在需要提供

1. 应用名、包名
2. v1 的实体与字段（例如“任务”：标题/状态/截止/标签/备注），或给出 Notion 数据库链接与字段截图
3. 是否只读还是可写双向同步

## 我可立即交付的内容

- Flutter 空工程骨架 + 依赖清单
- Notion 同步适配层接口与最小实现（读取列表、创建、更新、删除）
- 本地表结构与迁移脚本
- 设置页 Token 输入与检测流程
- 列表/详情基础 UI
- APK 打包脚本与检查表

## 注意事项

- Internal Integration Token 仅保存本机。备份设备会丢失，需要用户重输。
- 多人协作暂不支持。后续若扩展多人，需引入后端或走 Notion OAuth+PKCE。
- Windows 版后续可复用业务层。替换安全存储为 Windows 凭据库，UI 自适配后即可。