

项目申请书

跨平台 NFC 读写套件的维护与升级

申请人信息

基础信息	姓名：江晨炜
	邮箱：cheverjonathan@gmail.com
	GitHub：https://github.com/Chever-John
教育经历	江苏科技大学
	软件工程，学士，预计 2022 年 7 月毕业
开源经历	2020.09-now VSCode 中文社区主持人
	2020.09-now 自己的开源项目 (Ruoyi-Vue-PostgreSQL)

二零二一年五月

一、项目信息

项目名称

跨平台 NFC 读写套件的维护与升级

项目描述

项目基于 Flutter 框架，实现了跨平台的 NDEF 格式解析库、移动端 NFC 框架和 NFC 信息读取应用 NFSee。

目前，Flutter 已经发布了 2.0 版本，引入了较多的变化，迁移到新框架有一定的工作量。同时，在使用过程中，这些库积累了一些 bug 和功能请求，需要进一步的修复和实现以及维护。

背景介绍

项目的主要使用的技术是 Flutter 框架、Dart 语言、NFC 技术。

Flutter 是谷歌的移动 UI 框架，可以快速在 iOS 和 Android 上构建高质量的原生用户界面。Flutter 可以与现有的代码一起工作，起到快速上线的作用。在全世界，Flutter 正在被越来越多的开发者和组织使用。

Dart 是一个为全平台构建快速应用的客户端优化的编程语言，免费且开源。Dart 使用针对用户界面的创造进行优化的语言进行开发。可以进行反复地修改，在正在运行的应用中使用热重载立刻看到修改。Dart 可编译为移动端、桌面端及后端的 ARM & x64 的二进制文件，或是为 Web 平台编译 Javascript。项目中的 NFSee 便是由 Dart 语言进行开发的。

目前，各类移动设备，包括 Android 与 IOS，都增加了对 NFC 通信的支持；但各类应用对于 NFC 的支持都很不完善，缺少一个统一跨平台的支持库。针对以上需求，清华大学 TNUA 协会创建了本项目，并对其进行维护。

本项目的主要工作是将目前的三个项目(NDEF 库、flutter_nfc_kit、NFSee)均迁移到 2.0 版本。并维护好这三个库，修复好 bug。此外对 wiki 库进行更新维护。

申请人基本信息

申请人喜欢 coding，热爱分享技术，热衷于开源社区。在校期间有多次大型 web 项目经验，且大都有上线部署，被老师同学们使用。平日也会开发一些有趣的工具供大家使用，这是一种非常美妙的感觉，乐在其中。

希望这一次的项目能够成功通过申请。如果通过的话，必将竭力为开源社区贡献属于自己的一份力量。

二、项目方案

本次项目的关键点在于 Flutter 的迁移工作，现特为此任务制定方案。

迁移到空安全(null-safety)

基本步骤

1. **等待**项目所依赖的软件包迁移完成（如有必要，由项目开发者本人自行升级依赖）。
2. **迁移**项目的包的代码，使用 Android Studio 之类的交互式迁移工具完成。
3. **静态分析**包的代码。
4. **测试**项目的代码，确保可用。
5. 如果是依赖项目的话，进行项目的**发布**。

具体操作

1. 执行 **flutter pub outdated --mode=null-safety**，检查项目依赖的库是否都支持了空安全。

Package Name	Current	Upgradable	Resolvable	Latest
direct dependencies:				
cupertino_icons	^0.1.3	-	-	√1.0.3
flutter_nfc_kit	^2.1.0 (git)	-	-	^2.2.1 (git)
interactive_webview	^0.1.2	-	-	^0.1.2
intl	^0.16.1	-	-	√0.17.0
intl_translation	^0.17.10+1	-	-	^0.17.10+1
moor	^3.4.0	-	-	√4.3.1
moor_ffi	^0.8.0	-	-	^0.8.0
package_info	^0.4.3+2	-	-	√2.0.0
path	^1.8.0-nullsafety.1	-	-	√1.8.0
path_provider	^1.6.24	-	-	√2.0.1
url_launcher	^5.7.10	-	-	√6.0.4
webview_flutter	^1.0.6	-	-	√2.0.7
dev_dependencies:				
build_runner	^1.10.2	-	-	√2.0.3
flutter_launcher_icons	^0.7.5	-	-	√0.9.0
moor_generator	^3.3.1	-	-	√4.3.1

No resolution was found. Try running `flutter pub upgrade --null-safety --dry-run` to explore why.

由上图可见，项目所依赖的库大都已经支持了 null-safety 特性。还有几个库，如 intl_translation，需要等待其更新版本。

2. 首先将当前版本的库进行最新版本的升级，如需要将 moor 库从不支持 null-safety 的 3.4.0 版本升级到 4.3.1 版本。在此过程中考虑到依赖之间的关系。
3. 如果有必要，如到项目的完结，还存在没有支持 null-safety 特性的依赖，选择尝试自行进行更新 or 先砍掉相关功能（非重要功能）。
4. 发布已升级的项目（包括依赖和安装包）。

三、开发时间规划

时间	工作
7.1 之前	通过查阅收集资料，并尝试解决项目的 github 中 issue 和与 flutter 相关的 issues，更好的熟悉社区和环境。与社区项目导师沟通交流，然后整理可能的可选任务。
7.1-7.21	实现上一阶段的任务，开始和社区项目导师认真讨论任务完成的状况，并进行相应的调整。
7.22-8.7	至少完成 flutter_nfc_kit 库和 ndef 库的升级。
8.8-8.14	进行两个库的测试上线发布工作，此外 NFSee 的升级工作应该已经做到三分之一。
8.15	中期评审。
8.16-8.31	完成 NFSee 的升级工作，且开始着手 Wiki 文档的编写。进行项目的结尾工作。