## 关于firebase 的使用

<https://www.jianshu.com/p/a8008b18eb9f>

Firebase 官网

优势： NoSQL, 自动解决后端架构问题

使用方法：

注册，添加

在项目中直接 getinstance 就可以

## Stripe API 的 资料

中文博客

<https://blog.csdn.net/u012482647/article/details/98942190>

官方文档

<https://stripe.com/docs/api>

## 解读APPetit

Mylibrary： 三个不同界面的通用内容

数据库架构

字符串常量

一些自己定义的view，里面包含了对一些手势的不同解读

Scrollview

Mapview

通用函数

Utilities

### Rider: 针对骑手的应用

/NavigationFragments

/Home

显示里程数

/Fragments

切换status，对接 google map api

/Profile

/Services

/SplashActivity: 进来后启动的第一个界面 无实际意义

/MainActivity: 主界面 用于登陆，注册新用户

/FragmentManager:

用于管理页面切换，内部为BottomNavigationView，切换对象为上面三个Fragment

单开一个线程用于track当前location 这里面并没有实现

Customer：针对顾客的应用

/UI

各种个样的view

/ViewHolder

/Adapter

RecyclerView的相关配件

/item

只对用户游泳的一些数据库配置

这个项目的缺点：

把所有后端逻辑都写在了 activity 里面

改进：UI 和 业务逻辑分离

## Our Project大致框架（初步设想）

Com/dash/dashnext

/MyLibrary

/Values.java

定义字符串常量

数据库映射

通用函数

/Customer

MainActivity.java

CustomerActivity.java Not sure if necessary

/UI

/home

/profile

/history

/order

/StripeAPI

/Repository

factory.java

CustomerRepository.java

/Database

/Delivery

MainActivity.java

DeliveryActivity.java Not sure if necessary

/UI

/home

/profile

/order

/CurrentOrder

/AvailableOrder

设想：分为上下两部分

上部分是当前的即使order

下部分是未来的scheduled order

/Repository

/StripeAPI

/Database

感觉database可以合体放在外面

/Initial

/UI:定义初始界面

/MainActvity: customer starter page, can switch to ./Delivery

/SignIn: customer sign in

/SignUp: customer sign up, redirect to signin

/ForgetPassword: customer find password, redirect to signin

/Delivery: delivery starter page, can switch to ./MainActivity

/DeliverySignIn

/DeliverySignUp

/DeliveryForgetPassword

/