4.4 Calendar

1. 日历有四个是视图模式，分别是：日视图，月视图，周视图，日程视图。月视图主要对应的是month目录下，日程视图对应的是在agenda目录下。而日视图和周视图则显示对应在dayview这个类里面。这些视图都是Fragment的形式存在，并由AllInOneActivity来统管。
2. CalendarApplication这个类主要做了两件事情：（1）。设置Preferences的默认值通过的方法是setDefaultValues(this)；（2）。初始话一些自定义的东西init(getAssets())，从参数可以大概知道，要读入的文件存在assets目录下，根据init方法可以知道只是当存在calendar\_extensions.properties文件的时候，才会加载进来。
3. 根据AndroidManifest.xml，我们可以看到，AllInOneActivity是整个日历的入口类，几乎所有日历界面或是数据的交互，都是由这个类来穿针引线的，是一个名符其实的总管。在他的oncreate（）方法中，初始化了界面和显示。数据部分大概做两个事情：（1）账户检查，如果是第一次进入或者当前没有account的时候，会去加载googgle账户；（2）创建一个非常重要的CalendarController实列，并且通过registerFirstEventHandler（）注册第一个EventHandler。这个CalendarController（c）非常重要，他是整个Calendar的指挥官，下面会专门分析这个类。初始化界面的时候，第一步是获取要显示的视图模式是当前时间。第二步，根据视图模式创建一个actionbar。第三步，调用initFragments（）来加载对应视图的Fragments。setMainPane（）方法来加载最主要的视图界面，就是tab下面的那个视图，也是各种视图的最大区分。在setMainPane方法中，可以看到各种视图的Fragments替换。
4. 这里补充一些Fragments使用中常用的使用方法。在一个activity中要使用Fragments，要管理fragments，需使用FragmentManager，要获取它，需在activity中调用方法getFragmentManager()获取其对象，并用beginTransaction()开启fragment的事务管理。常用的方法：使用方法findFragmentById()或findFragmentByTag()，获取activity中已存在的fragment。Add（）添加。Remove（）移除，Replace（）替换fragment。什么是一个事务呢？一个事务是在同一时刻执行的一组动作（很像数据库中的事务）。你可以用add(),remove(),replace()等方法构成事务，最后使用commit()方法提交事务。在调用commint()之前，你可以用addToBackStack()把事务添加到一个后退栈中，这样做的作用是，当你回退的时候，事务中所做的操作可以依次返回（这叫做事务回溯）），否则不能回退。注意：你只能在activity处于可保存状态的状态时，比如running中，onPause()方法和onStop()方法中提交事务，否则会引发异常。这是因为fragment的状态会丢失。如果要在可能丢失状态的情况下提交事务，请使用commitAllowingStateLoss()。
5. CalendarController在日历中的作用，我们知道android中有一个非常重要的设计模式—MVC，日历就是采用的这种设计模式，那么现在看这个类的名字就可以知道他的作用了，他就是对应mvc中的 c层——Controller层，负责分发事件。总体的流程如下:

各个视图 view层

CalendarController Controller层

Handevent（）事件处理 Mode层

1）Envent事件分成两类，涉及caledar Envent(如 edit caledar Envent)和不涉及caledar Envent（如界面的滑动事件等）。CalendarController为view层提供了sendEventrelatedEvent和sendEvent（）来处理这些事件。

2）CalendarController向model层提供了一个eventhandle接口，来作为向下分发事件的统一接口。