1. 功能一： 添加提醒
   1. **功能介绍**

**功能概述：**对于特定便签可以选择增添提醒，使得对应时间的时候自动显示的闹钟图标，提醒用户当前时间的事项。

**操作步骤：**便签编辑界面选择“添加提醒”选项，弹出时间选择框，确定提醒时间后便签显示闹钟图标，标志提醒时间。

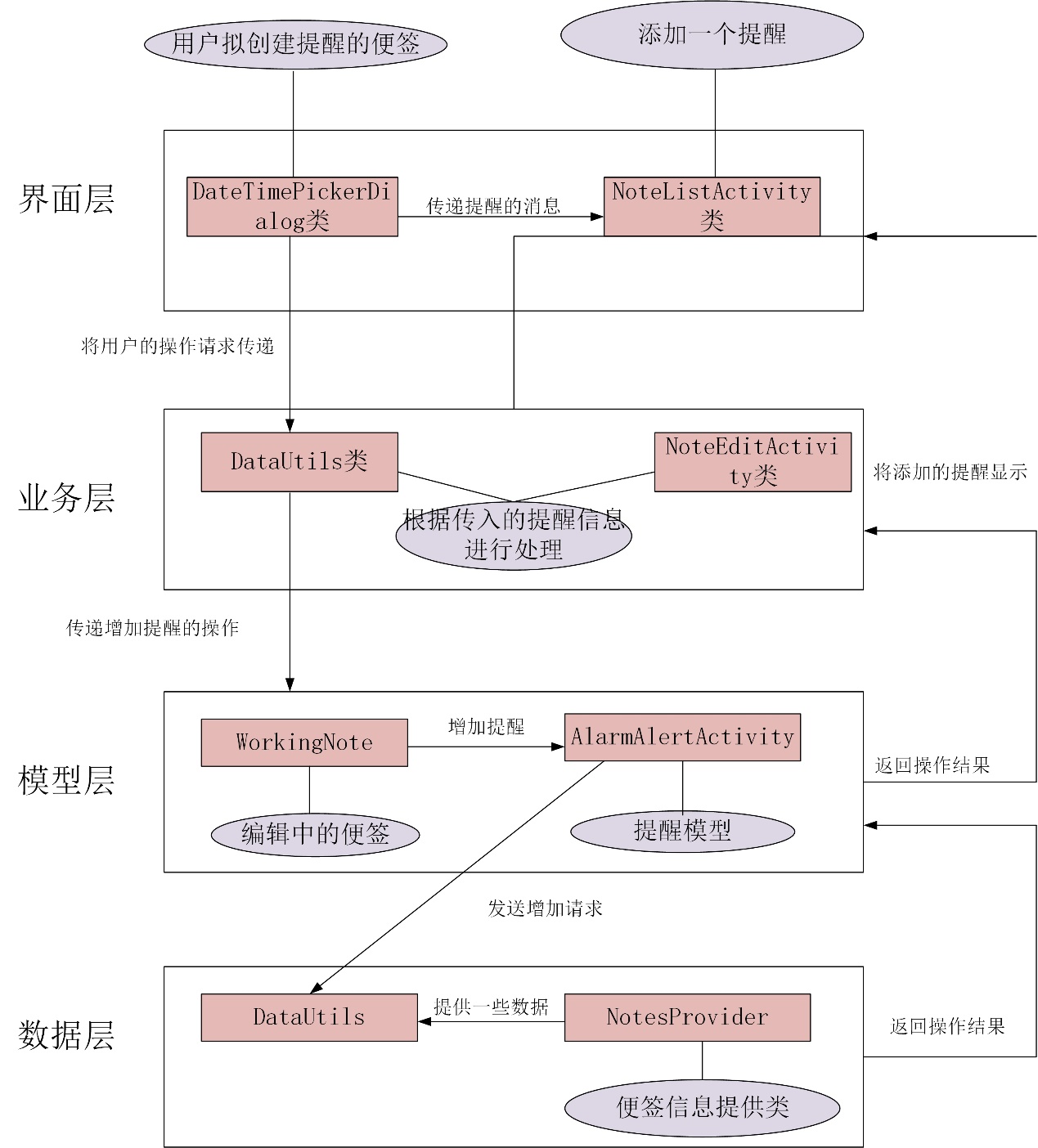
**业务请求与响应过程**

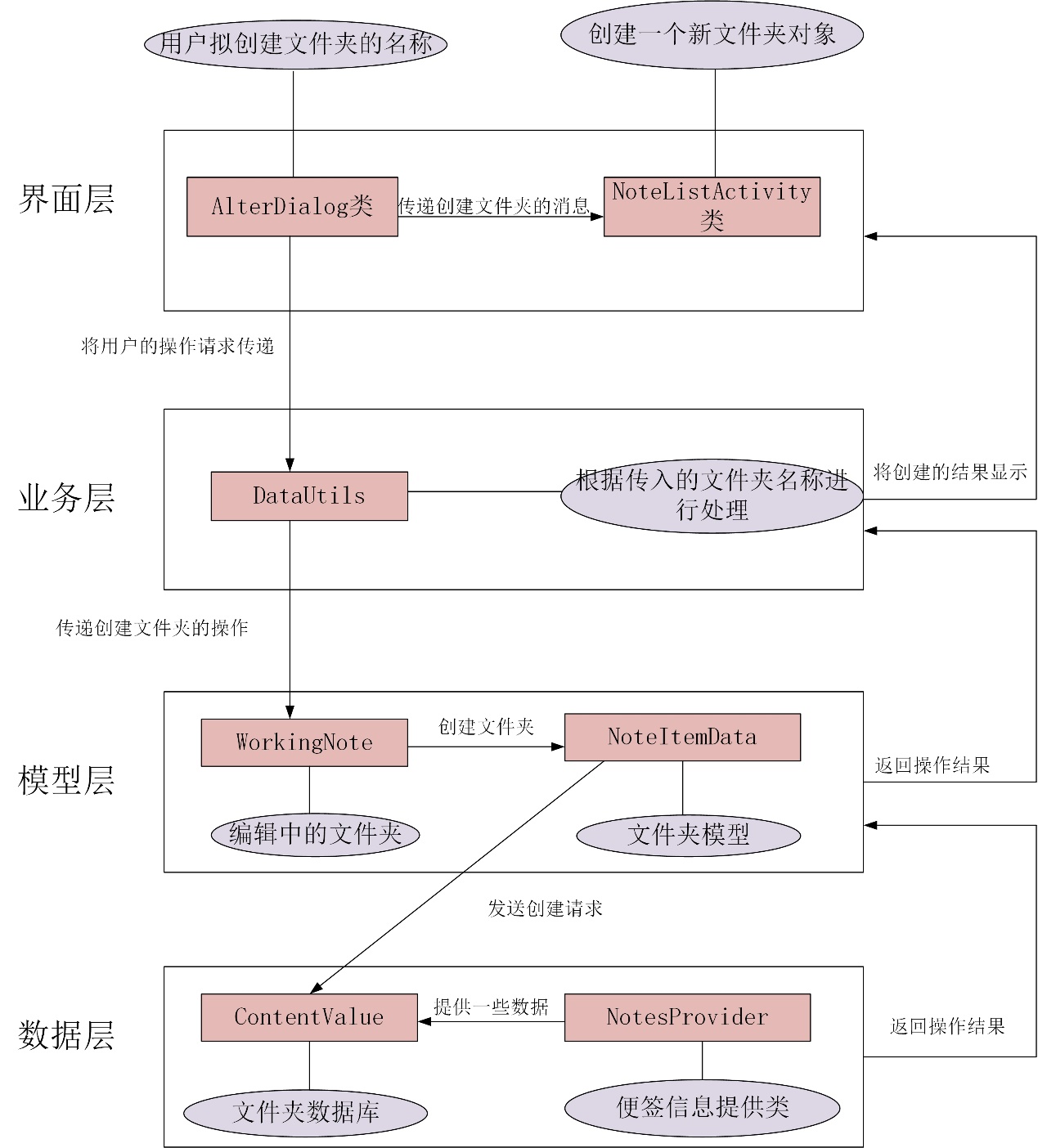
添加提醒功能（下简称“功能”）的界面层主要由包ui中DateTimePickerDialog类通过调用show ()方法实现。show ()中使用AlertDialog.Builder构建了一个对话框，显示一个包含文件夹名称输入框和确认取消按钮的对话框。

功能的业务层由NoteEditActivity类中的setReminder()方法实现。首先，创建一个 DateTimePickerDialog 对象，传入当前系统时间作为初始时间。然后设置一个 OnDateTimeSetListener，用于监听用户设置日期和时间的操作。当用户设置完日期和时间后，调用 mWorkingNote.setAlertDate() 方法，将设置的日期和时间以及一个布尔值传递给该方法。最后显示 DateTimePickerDialog 对话框，供用户选择日期和时间

功能的模型层主要由ui包中的AlarmAlertActivity类来实现，主要包括onClick()和onDismiss()方法等。

功能的数据层主要由DataUtils类的对象来实现。包括了警报的信息，注释的内容、标题等等。





* 1. **类的作用**
     1. **界面层相关的类**

1. **类名：****DateTimePickerDialogy**

该类是小米便签的提醒界面，在该界面可以进行便签提醒的相关操作，用于显示警报提醒并处理用户的操作。

1. **类的精读：**

**DateTimePickerDialog**类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mDate | 存储日期和时间 |
| mIs24HourView | 表示是否使用24小时制显示时间 |
| mOnDateTimeSetListener | 监听用户设置日期和时间的操作 |
| mDateTimePicker | 显示日期和时间选择器 |

**DateTimePickerDialog**类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| set24HourView(boolean is24HourView) | is24HourView:判断是否使用24小时制显示时间 | 设置是否使用24小时制显示时间 | 将接收到的布尔值参数 is24HourView 赋值给成员变量 mIs24HourView，以便在更新对话框标题时使用。 |
| setOnDateTimeSetListener(OnDateTimeSetListener callBack) | callBack：回调监听器对象 | 设置日期时间设置完成后的回调监听器 | 将传入的回调监听器对象赋值给成员变量 mOnDateTimeSetListener，以便在用户设置完日期时间后触发相应的回调方法。 |
| updateTitle(long date) | Date:日期 | 根据传入的日期更新对话框的标题 | 定义一个标志变量 flag，用于指定日期和时间格式的显示方式。然后，将 mIs24HourView 是否为 true 的判断结果作为一个条件标志位添加到 flag 中。最后，调用 DateUtils 类的 formatDateTime() 方法，传入上下文、日期、标志等参数，生成格式化的日期时间字符串，并将其赋值给对话框的标题 |
| onClick(DialogInterface arg0, int arg1) | arg0：DialogInterface对象  arg1：参数 | 处理点击对话框按钮的事件。调用回调监听器的方法将设置的日期时间传递给回调方法。 | 调用回调监听器的 OnDateTimeSet() 方法，将当前选定的日期时间（通过 mDate.getTimeInMillis() 获取）作为参数传递给回调方法。 |
| onDateTimeChanged(DateTimePicker view, int year, int month, int dayOfMonth, int hourOfDay, int minute) | View：DateTimePicker对象  Year：年  Month：月  dayOfMonth：日期  hourOfDay：小时  minute：分钟 | 更新 mDate 的值，并调用 updateTitle() 方法更新对话框标题 | 更新了 mDate 对象的各个字段，以反映用户所选的日期和时间。然后，调用 updateTitle() 方法，将更新后的日期设置为对话框的标题。 |

* + 1. **业务层相关的类**

**类名：NoteEditActivity**

1. **类的作用：**

保存用户的配置信息，如当前便签的修改字体大小和背景颜色的简单的配置信息。

1. **类的精读：**

表1 类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| HeadViewHolder | 用于保存一组视图的引用，有修改日期文本，背景颜色，提醒图标，和提醒日期文本的视图。 |
| sBgSelectorBtnsMap | 保存背景颜色选择器中选项和对应颜色的映射关系 |
| sBgSelectorSelectionMap | 保存颜色id值与对应颜的映射关系 |
| mNoteBgColorSelectot | 背景颜色的选择器 |
| mWorkingNote | 当前操作的便签项 |
| mNoteEditorPanel | 操作背景颜色的修改 |
| mHeadViewPanel | 操作标题颜色的修改 |

表2 类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| setBgClorId(int id) | Id:用来修改当前颜色的值 | 修改当前便签背景颜色的ID值 | 先检查选择器是否可见，如果可见并且触摸操作不在选择器范围内则将其设置为不可见，并返回true来拦截触摸事件。 |
| dispactchTouchEvent() | ev:屏幕触碰的操作 | 颜色选择器和字体大小的选择器在屏幕上可见。同时用于在其他特殊操作下隐藏选择器 | 先检查选择器是否可见，如果可见并且触摸操作不在选择器范围内则将其设置为不可见，并返回true来拦截触摸事件。 |
| InRangeOfView(View view, MotionEvent ev) | view ：用来表示ui容器，容器的可触范围。  ev ：屏幕触碰操作 | 用来确定触碰操作是否 在给定的view范围内 | 先确定容器的范围location，在用ev的getX与getY操作判断操作点是否位于location内 |
| InitResource() | 无 | 初始化UI界面的各种元素和资源 | 通过给定id，R.id类的内容来赋值 |
| onBackPressed() | 无 | 确保用户按下返回按钮时，未保存的便签内容可以得到保存 | 先检查用户是否处于其他设置状态，否则则直接调用saveNote()的方法保存内容。同时调用super.onBackPressed方法来执行默认返回操作。 |
| clearSettingState() | 无 | 如果需要时，保证颜色选择器和字体大小选择器在不使用的时候是隐藏的 | 用判断结构，如果当前选择器为可见，则设置为不可见。 |
| onClick(View v) | v: 菜单选择器内对应向的id值 | 用来实现用户点击与界面中视图的交互，根据不同的点击内容，执行不同的点击结果。例如修改字体的大小和修改背景颜色 | View.OnclickListener的接口来监听点击事件，在onClick方法中实现对应操作 |
| onOptionsItemSelected(MenuI item) | Item;菜单点击事件 | 根据不同的菜单点击事件，使菜单可见，或执行当前菜单内对应的功能 | 通过获得item的事件id，来匹配不同的菜单内的功能，然后再执行对应功能的内容。 |
| InitNoteScreen() | 无 | 加载设置的字体大小样式等样式个性化内容到屏幕上 | 调用mNoteEditor中的setTextAppearance的方法来实现。 |
| onBackgroundColorChanged() | 无 | 修改当前的背景颜色 | 通过id值来修改对应的背景颜色 |
| setReminder() | 无 | 设置提醒 | 创建一个 DateTimePickerDialog 对象，传入当前系统时间作为初始时间。然后设置一个 OnDateTimeSetListener 监听器，在用户设置完日期和时间后调用 mWorkingNote.setAlertDate() 方法来设置提醒日期和时间。 |
| sendTo(Context context, String info) | context ：上下文对象  info ： 要分享的便签内容 | 分享便签内容给支持 Intent.ACTION\_SEND 动作和 text/plain 类型的应用程序 | 创建一个发送文本的意图，并将便签内容设置为 EXTRA\_TEXT 的额外数据，然后启动适合的应用程序 |
| createNewNote() | 无 | 用于创建新的笔记 | 保存当前正在编辑的笔记；然后，启动一个新的 NoteEditActivity 来创建一个新的笔记。 |
| deleteCurrentNote() | 无 | 删除当前笔记 | 删除当前笔记。如果笔记存在于数据库中，它会将笔记标记为已删除，并根据同步模式批量删除或移动到垃圾箱文件夹 |
| onClockAlertChanged(long date, boolean set) | date ： 提醒日期和时间  set ： 是否设置提醒 | 设置或取消闹钟提醒 | 根据是否设置提醒，创建一个隐式意图，关联到 AlarmReceiver 类，并使用笔记的 URI 数据。然后创建一个 PendingIntent 对象，并使用 AlarmManager 设置闹钟提醒。如果没有输入任何内容（即没有笔记 ID），则显示错误消息提醒用户输入内容 |
| isSyncMode() | 无 | 检查是否处于同步模式 | 检查是否处于同步模式。它根据用户在设置中配置的同步账号名称来判断。 |

* + 1. **模型层相关的类**

**类名：AlarmAlertActivity**

该类是小米便签的提醒界面，在该界面可以进行便签提醒的相关操作，用于显示警报提醒并处理用户的操作。

1. **类的精读：**

**AlarmAlertActivity**类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mNoteId | 警报的注释ID |
| mSnippet | 警报的摘要信息 |
| SNIPPET\_PREW\_MAX\_LEN | 摘要预览的最大长度 |
| mPlayer | MediaPlayer对象，用于播放警报声音 |

**AlarmAlertActivity**类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate(Bundle savedInstanceState) | savedInstanceState 是一个Bundle对象，用于保存Activity的状态信息 | Activity的生命周期方法，用于初始化Activity，设置布局文件，调用initResources()方法初始化资源 | 首先调用了父类onCreate方法，然后通过setContentView方法设置了布局为R.layout.note\_list |
| isScreenOn() | / | 检查屏幕是否亮起 | 返回值：布尔值，表示屏幕是否亮起 |
| playAlarmSound() | / | 播放警报声音 | 首先获取默认警报铃声的URI。然后获取静音模式下受铃声模式影响的流。接着设置MediaPlayer的音频流类型。最后设置MediaPlayer的数据源并进行准备、循环播放和启动。 |
| showActionDialog() | / | 显示警报操作对话框。 | 创建一个AlertDialog.Builder对象。  设置对话框的标题和消息内容为警报的摘要信息。  设置对话框的确定按钮以及可能的取消按钮。  显示对话框并设置对话框消失时的监听器。 |
| onClick(DialogInterface dialog, int which) | DialogInterface dialog：被点击的对话框。  int which：被点击的按钮的标识。 | 处理对话框按钮的点击事件。 | 根据点击的按钮不同执行相应的操作，例如点击取消按钮跳转到NoteEditActivity页面。 |
| onDismiss(DialogInterface dialog) | DialogInterface dialog：被解雇的对话框。 | 对话框消失时调用，停止警报声音并关闭活动。 | 停止播放警报声音，释放MediaPlayer资源，并结束当前活动。 |
| stopAlarmSound() | / | 停止警报声音的播放。 | 停止MediaPlayer的播放，释放MediaPlayer资源。 |

* + 1. **数据层相关的类**

**1、类名：**DataUtils

1. **类的作用**

一个数据处理的工具类，能够包含了一系列用于处理笔记数据的静态方法。这些方法涵盖了笔记的批量删除、移动到文件夹、获取文件夹的笔记部件信息、检查文件夹名是否存在等功能。

1. **类的精读：**

表1 类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| TAG | public static final String TAG = "DataUtils"; |

表2 类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getUserFolderCount | ContentResolver:  查询便签内容  （Notes.CONTENT\_NOTE\_URI）获取游标（Cursor） | 获取用户文件夹的数量 | 查询便签内容获取游标  查询时指定了特定条件：  选择字段 "COUNT(\*)"，表示查询的结果是文件夹的数量。  如果游标不为空（则移动游标到第一行  尝试从游标中获取第一列的整数值，即文件夹的数量最后，无论是否发生异常，都需要关闭游标  返回获取到的文件夹数量。 |
| batchDeleteNotes | HashSet<Long>：id列表  ContentResolver: 一个内容解析器  ContentUris.  withAppendedId():用于构建便签的特定URI。  operationList操作列表 | 根据传入的便签 ID 集合进行批量删除操作 | 遍历 ids 集合中的每个便签ID，根据ID构建一个 对象，指定要删除的具体便签。  利用内容提供者操作类执行与内容提供者相关的操作，例如插入、更新、删除等。  返回结果数组 results 中的第一个结果判断操作是否成功。如果结果为空或第一个结果为空，表示删除失败。  如果删除操作成功，则返回 true，否则返回 false。 |
| moveNoteToFoler | resolver：用于与应用程序的数据提供者进行通信。  id：要移动的便签的唯一标识符。  srcFolderId：便签当前所在的文件夹的唯一标识符。  desFolderId：要移动到的目标文件夹的唯一标识符。 | 将指定的便签移动到不同的文件夹中 | 创建一个对象，用于存储要更新的便签的新值。之后更新便签的属性值。  将更新后的 values 对象应用于指定的便签。指定要更新的便签的位置。 |
| batchMoveToFolder | resolver：用于交互。  ids：包含要移动的便签的 ID。  folderId：要移动到的目标文件夹的 ID。 | 将一组便签移动到指定文件夹 | 对于 ids 中的每个便签 ID，创建一个Operation对象，并设置更新的 URI 和要修改的列的值。这里的列包括父文件夹ID和本地修改标志。  将创建的Operation对象添加到 operationList 中。  使用 resolver 对象调用 applyBatch() 方法，将 operationList 中的操作应用到 ContentProvider 中。  根据返回的结果进行判断：如果 results 为空、长度为 0 或第一个元素为空，则记录一条日志并返回 false。  如果没有抛出异常，则返回 true。否则，记录异常信息并返回 false。 |