1. 功能二： 删除便签
   1. **功能介绍**

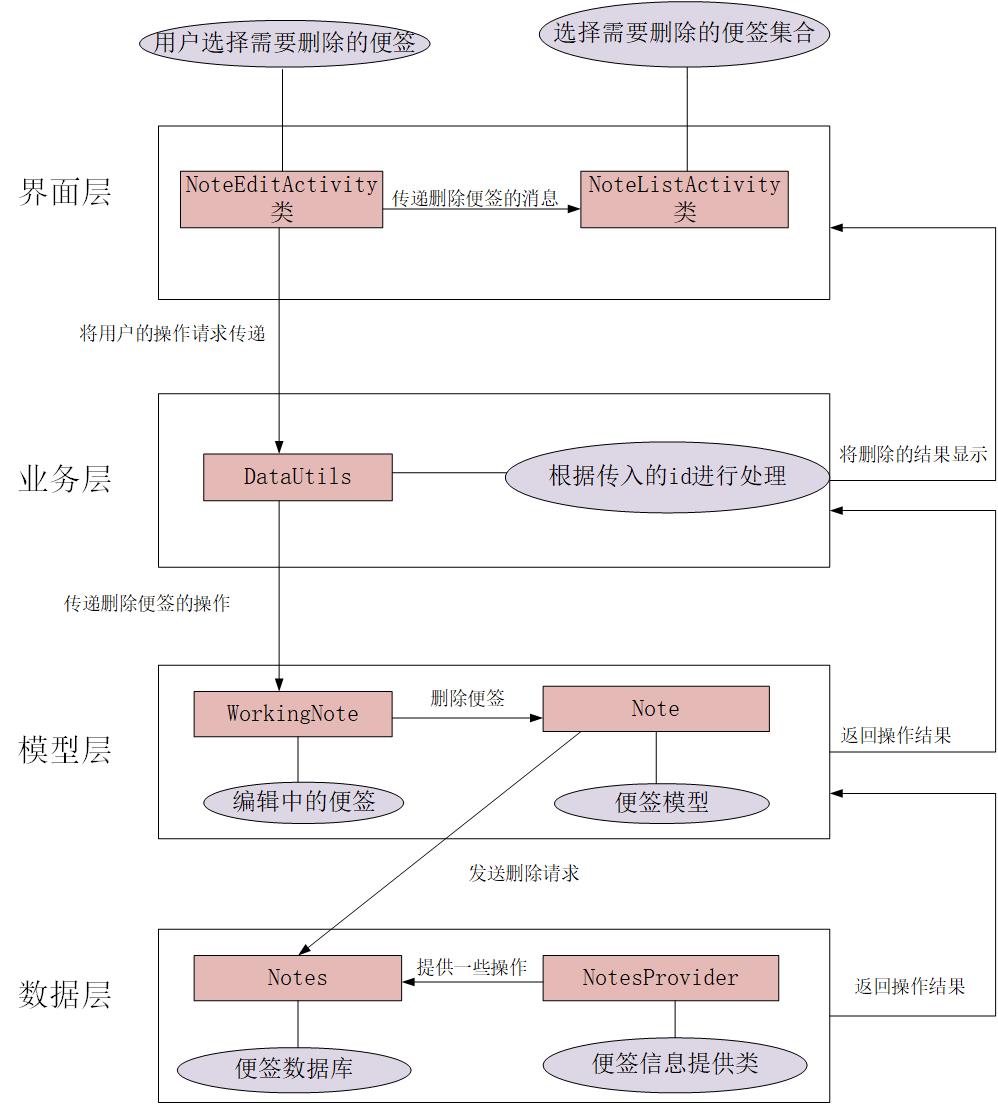
**功能概述：**将一条便签从本地数据库中删除

**操作步骤：**在便签列表界面长按某一便签，会进入选择界面，选中想要删除的便签，点击下方垃圾桶即可进行删除，能够对便签集进行有效管理

**业务请求与响应过程**

在删除便签的过程中，四个层级各自的作用大致如下：

1. **界面层**：用户在小米便签应用的界面上选择要删除的便签，并点击删除按钮或选择删除选项时，触发界面层的事件处理。
2. **业务层**：界面层接收到用户操作后，将请求传递给业务层。业务层负责处理业务逻辑，包括验证用户操作的合法性、构造请求对象、调用模型层接口等。
3. **模型层**：业务层将删除便签的请求传递给模型层。模型层表示应用程序的核心组成部分，处理与数据相关的操作。它负责与数据层进行交互，执行删除便签的操作，并将结果返回给业务层。
4. **数据层**：模型层通过调用数据层的接口，向数据库或存储系统发送删除便签的请求。数据层负责实际的数据操作，包括获取、保存、更新和删除数据等。在删除便签的过程中，数据层会根据请求中的标识符或其他信息，从数据库或存储系统中删除相应的便签数据，并返回操作的结果给模型层。



* 1. **类的作用**
     1. **界面层相关的类**

**1、类名：NoteEditActivity**

1. **类的作用：**

这个类可以显示便签的内容，允许用户编辑便签，包含一个文本编辑框，使用户可以输入、编辑和格式化便签的文本内容，保存便签的修改，提供界面交互，处理用户的交互操作

**类的精读：**

表1 类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| HeadViewHolder | 用于管理小米便签中编辑界面的头部视图，用于优化列表或视图的性能，它持有对子视图的引用，以便在需要时快速访问。 |
| mNoteEditor; | 用于接收用户输入的文本，以便创建新的便签或编辑现有的便签内容 |
| mNoteEditorPanel; | 用于显示编辑便签界面的面板视图 |
| mWorkingNote; | 包含便签的相关信息 |

表2 类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| onCreate | super.onCreate  (savedInstanceState)：  调用父类的 onCreate 方法以执行基本的活动创建操作。  this.setContentView  (R.layout.note\_edit)：  设置当前活动的布局为 note\_edit.xml，这将决定界面的呈现样式。 | 用于在活动（Activity）创建时进行初始化 | 检查是否存在已保存的活动状态和通过 getIntent() 获取到的意图。如果没有保存的状态并且初始化活动状态失败，那么调用 finish() 方法结束当前活动。 |
| onSaveInstanceState | mWorkingNote：便签保存在数据库中的形式  outState：用于存放已经被删除的便签名称 | 用于保存当前活动的状态，以便在需要时进行恢复。用于保存正在编辑的便签的相关信息。 | 通过检查 mWorkingNote 是否已经存在于数据库中来决定是否需要保存便签。  如果 mWorkingNote 还不存在于数据库中，可能意味着这是一条新的便签（此时没有便签ID），因此会调用 saveNote() 方法将该便签保存到数据库中，以生成一个便签ID。然后，将便签的ID（mWorkingNote.getNoteId()）保存到 outState 中，以便在稍后的状态恢复操作中使用。 |
| onClick | View v：需要进行鼠标点击操作的视图  Id：视图名称 | 用于处理用户在界面中点击操作的事件。 | 根据用户点击位置（R.id.btn\_set\_bg\_color）  （sBgSelectorBtnsMap）  （sFontSizeBtnsMap）  的区别，进行相应的判断和响应 |
| dispatchTouchEvent | MotionEvent：当前事件 | 用于处理触摸事件的分发 | 在函数中，会根据条件判断来决定是否隐藏一些视图，并返回一个布尔值来指示是否消费了该事件。 |
| onOptionsItemSelected | MenuItem：菜单选项 | 根据被选中的菜单项的ID，执行相应的操作 | 这段代码是一个菜单项选择的回调方法，根据被选中的菜单项的ID，执行相应的操作，具体到删除便签的这一操作时，当菜单项的ID为menu\_delete\_note时，创建一个确认删除提示对话框，并在用户点击确认时调用deleteCurrentNote()方法进行删除操作，然后结束当前活动。 |
| deleteCurrentNote | mWorkingNote：便签模型，为正在被操作的便签 | 删除当前的便签 | 首先，它检查mWorkingNote表示的便签是否存在于数据库中，通过调用existInDatabase()方法进行检查。如果便签存在于数据库中，它会创建一个 HashSet 对象 ids，用于存储要删除的便签的唯一标识符。如果便签成功从数据库中删除或移动，将会调用 mWorkingNote.markDeleted(true) 方法，将 mWorkingNote 对象标记为已删除 |

1. **类名：NotesListActivity**
2. **类的作用：**

该类是小米便签的主列表界面，在该界面可以进行文件夹和便签的相关操作，例如创建文件夹，移动便签，新建便签等功能。

1. **类的精读：**

表1 类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mBackgroundQueryHandler | 异步查询类对象，负责异步线程访问数据库取出相应的便签数据 |
| mNotesListAdapter | 用于管理便签列表展示的适配器（Adapter）实例。 |
| mNotesListView; | 用于显示便签列表的 ListView 对象。 |
| mAddNewNote; | 用于增加新的note对象 |
| BackgroundQueryHandler | 用于在后台执行异步查询操作 |

表2 类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| startAsyncNotesListQuery | mBackground  QueryHandler：异步查询类对象，负责异步线程访问数据库取出相应的便签数据 | 对便签进行按照日期的排序 | 在后台异步查询便签列表数据，并根据一些条件进行筛选，以获取与当前文件夹相关的便签数据，并按照类型和修改日期降序进行排序。 |
| batchDelete | AsyncTask：判断同步  mNotesListAdapter：用于选中便签集合  doInBackground 方法最后返回 widgets 对象。 | 批量删除便签 | 该异步任务首先通过 mNotesListAdapter 获取选中的便签的集合 widgets。  如果当前不处于同步模式（!isSyncMode()），则直接调用 DataUtils.batchDeleteNotes() 方法来删除选中的便签。  如果当前处于同步模式，将选中的便签移动到垃圾箱文件夹，调用 DataUtils.batchMoveToFolder() 方法。 |

* + 1. **业务层相关的类**
  1. **类名：**DataUtils

1. **类的作用：**
2. 便签数据处理工具类，封装如查找、 移动、删除数据等操作，是一种用于处理数据相关操作的实用工具类
3. **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。 |

* + 1. **模型层相关的类**

**1、 类名：Note**

1. **类的作用：**单个便签的模型
2. **类的精读：**

表1 类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mNoteDiffValues | 存储便签的差异值 |
| mNoteData | 存储便签数据 |
| TAG | private static final String TAG = "Note"; |

表2 类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| getNewNoteId | ContentValues：用于存储便签的各个属性和值  Context：上下文参数  folderId：文件夹名称 | 获取新便签的ID | 从数据库中获取一个新的便签 ID。创建一个对象，用于存储便签的各个属性和值。获取当前时间作为新便签的创建时间。  将各种属性和值添加到对象中，返回新便签的 ID。 |
| Note | ContentValues：用于存储便签的各个属性和值 | Note 类的构造函数。它用于初始化一个 Note 对象。 | 这是一个无参数的构造函数，用于创建 Note 对象。 |
| getNewNoteId | Context：上下文参数  folderId：文件夹名称 | 在数据库中创建一个新的便签，并返回该便签的ID | 创建一个待插入数据库的对象，用于设置新便签的各个属性，指定便签所属的文件夹，获取插入后自动生成的 noteId |
| setNoteValue | Key：需要被设置的属性  Value：应该设置到key中的值 | 设置便签的某个属性的值，并对相关的属性进行相应的更新。 | 通过参数 key 和 value，将指定属性的新值添加到 mNoteDiffValues 中 |

* 1. **类名：WorkingNote**

1. **类的作用：**正在编辑的便签的模型
2. **类的精读：**

表1 类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| mNote | Note for the working note |
| mNoteId | Note Id |
| mNoteSettingStatusListener | 用于监听便签设置变化的监听器 |
| DATA\_PROJECTION | 一个字符串类型的数组，其中包含了一些列名或字段名作为查询数据的投影 |
| NOTE\_PROJECTION | 包含了一组用于查询便签数据的列名 |

表2 类中的方法

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| WorkingNote | WorkingNote的一些参数需要 | 私有构造方法 | 私有构造方法 |
| createEmptyNote | Context：上下文参数  folderId：文件夹名称  widgetId：插件id  widgetType：插件类型  defaultBgColorId：  用于颜色设置 | 创建一个空的便签，并进行一些初始设置 | 创建了一个 WorkingNote 类的实例，这行代码设置了便签的背景颜色为默认颜色，设置了便签的小部件ID。  设置了，便签的小部件类型最后，该方法返回创建的 WorkingNote 实例。 |
| markDeleted | Mark：一个布尔值 | 用于设置便签的删除状态 | markDeleted 方法用于标记便签的删除状态。通过将布尔值 mark 赋值给成员变量 mIsDeleted，可以设置便签的删除状态。 |
| isWorthSaving | mIsDeleted：标记了删除操作  existInDatabase TextUtils.isEmpty  ：判断了当前便签是否存在于数据库中并且内容是否为空  existInDatabase()  mNote.isLocalModified()  判断当前便签是否在数据库中存在，并且是否有本地修改 | 用于判断是否值得保存，这可以和naveNote联动 | 检查是否标记了删除操作，如果已经删除，则可能认为不值得保存。  结合判断了当前便签是否存在于数据库中并且内容是否为空。如果便签既不在数据库中也没有内容，则可能认为不值得保存。  判断当前便签是否在数据库中存在，并且是否有本地修改。如果便签在数据库中存在但没有本地修改，则可能认为不值得保存。 |

* + 1. **数据层相关的类**

**1、类名：Notes**

1. **类的作用：**

便签数据库，用于记录便签相关属性和数据

1. **类的精读：**

表1 类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| DataConstants | 用于指示一些常量 |
| NoteColumns、DataColumns | 用于定义一些final类型的值 |
| TextNote、CallNote | 对DataColumns进行接口的实现 |

此类由于是data类，用于保存和记录数据，没有相关的方法

* 1. **类名：NotesProvider**

1. **类的作用：**便签信息提供类
2. **类的精读：**

继承了 ContentProvider （Android 中用于提供数据访问接口的基类，可以从其他应用程序中访问和共享数据。）类

用于处理小米便签数据的内容提供器。它可能提供了一些方法来实现对便签数据的增删改查操作，以及与其他应用程序进行数据共享和访问的功能。

**类中的方法**

|  |  |  |  |
| --- | --- | --- | --- |
| **方法** | **参数解释** | **方法作用** | **实现过程** |
| insert | URI：唯一标识符  ContentValues：目录 | 用于插入数据到数据库中 | 据提供的 uri，它将区分是插入便签数据还是插入其他数据。 |
| getType | URI：唯一标识符 | 判断给定的 Uri 是属于什么类型的数据 | 根据给定的 Uri 获取其类型，并返回一个字符串表示 |
| delete | URI：唯一标识符  deleteData:布尔值  selection：删除选择内容 | 一个删除便签或数据的方法 | 通过匹配 Uri 来确定要删除的是便签还是数据。根据不同的 Uri 类型，采取不同的删除操作。  在删除便签时，根据传入的选择条件和参数执行删除操作。将删除影响的行数保存到 count 变量中。 |