1. 功能三： 移动便签
   1. **功能介绍**

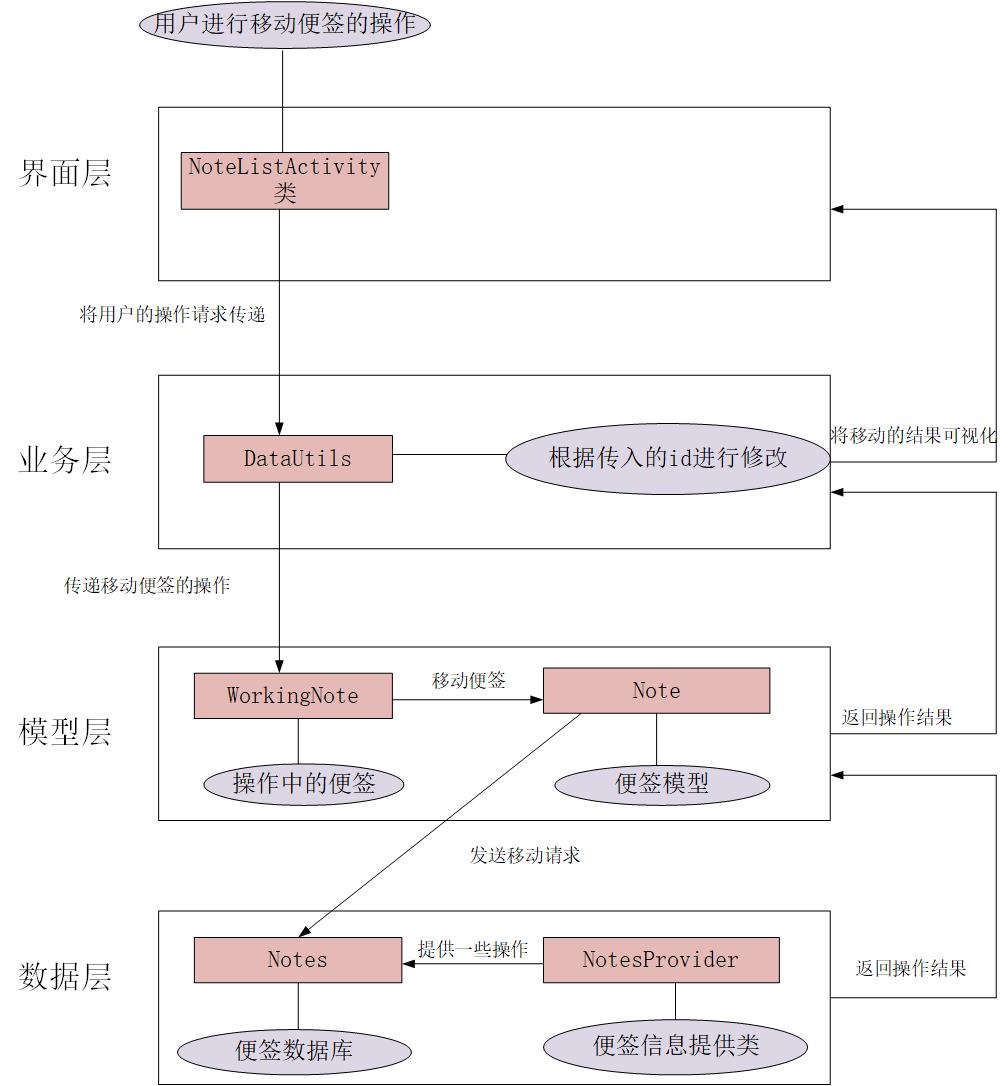
**功能概述：**将便签进行移动操作，方便管理

**操作步骤：**在便签列表界面长按某一便签，进入选择界面，选中想要移动的标签，点击左下方移动按钮，选择目标文件夹，就可以将快速创建的便签归类以进行有效管理

**业务请求与响应过程**

在移动便签的过程中，四个层级各自的作用大致如下：

1. **界面层**：界面层是用户与移动便签应用进行交互的部分。它包括应用的用户界面、用户输入的处理和展示等。用户通过界面层发送操作请求，如创建新的便签、移动便签位置等。
2. **业务层**：业务层处理来自界面层的请求，并进行相应的业务逻辑处理。它负责验证用户操作的合法性、调用合适的模型进行数据处理等。在移动便签中，业务层会根据用户请求，调用模型层的接口进行数据处理。
3. **模型层**：模型层是处理数据的核心部分。它包含处理数据的逻辑和相关的操作方法。在移动便签中，模型层负责创建、更新、删除便签数据等。它可以对数据进行操作，以满足业务层的需求。
4. **数据层**：数据层负责存储和管理移动便签应用的数据。它包括数据库、文件系统或其他数据存储机制。数据层提供了对数据的读取和存储操作，以满足模型层的需求。移动便签的数据层负责将数据存储到服务器端，并处理请求中的数据读取和更新操作。



* 1. **类的作用**
     1. **界面层相关的类**
  2. **类名：NotesListActivity**

1. **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。 |
| openFolder | mCurrentFolderId：当前文件夹的唯一标识符 | 打开一个文件夹，并根据传入的参数 对应的数据来执行相应的操作。 | 于打开一个文件夹，并根据传入的参数 NoteItemData 对应的数据来执行相应的操作 |

* + 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。 |
| 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。 |

* + 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。 |
| loadNote | mContext：目录 | 加载便签数据 | 通过提供的便签ID查询便签的数据。  如果查询结果不为 null，则将查询结果中的各个字段值分别赋给相应的成员变量。 |
| loadNoteData | mContext：目录 | 查询与特定ID 相关的数据。 | 进行数据库查询操作查询与特D相关的数据。通过循环逐个遍历查询结果中的行。  根据不同的数据类型，进行相应的操作 |

* + 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 |
| update | URI：唯一标识符  mMatcher：匹配器 | 用于更新数据库中数据 | 首先，通过传入的参数 uri  使用 mMatcher  .match(uri) 来执行一个匹配操作，判断要进行的操作类型。  根据匹配的结果，执行相应的操作：  如果更新的行数大于 0，表明有数据更新成功，则执行以下操作：  如果 updateData 为 true，表示数据项（Data）已更新，则通过调用 notifyChange() 方法通知数据项发生变化。  最后，返回更新的行数。 |