1. 功能八： 导出便签
   1. **功能介绍**

**功能概述：**主界面的菜单栏中选择“导出文件”，在Android手机提供SD卡支持的情况下，能够将小米便签中的便签内容逐个转化为“.txt”格式的文本文档，并存储到SD卡中。

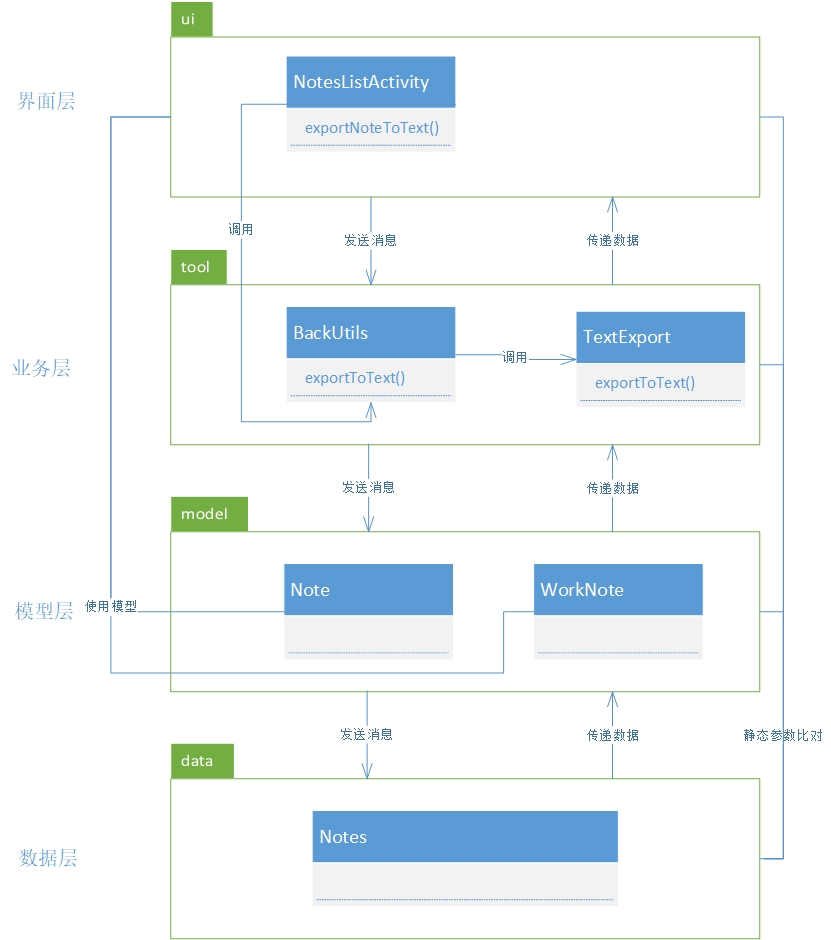
**操作步骤：**在主列表界面点击右下角菜单栏，点击“导出文本”按钮，软件自动检测Android手机是否有SD卡，有SD卡则导出文本到路径“mnt/sdcard/MIUI/notes”路径下；反之如果SD卡未挂载、SD卡正在被占用、系统出错则弹出错误信息提示弹窗。下图为成功导出界面和未成功导出界面（由于测试物理机、虚拟机均没有SD卡，成功用例图系网上查找）

* 1. **业务请求与响应过程**

导出便签功能（下简称“功能”）的界面层由包ui中NotesListActivity类调用exportNoteToText()方法实现。exportNoteToText()中新建了一个不可修改的BackupUtils备份工具类实例；新建一个异步任务AsyncTask执行具体的备份功能，在后台调用BackupUtils的exportToText()方法，在界面层接收下层返回的结果，并对执行结果进行弹窗提示。

功能的业务层由包tool中BackUtils类调用exportToText()方法实现。exportToText()中调用了内部类TextExport的exportToText()将便签导出生成的文本写入目标地址文件中，其中通过getExportToTextPrintStream()方法创建PrintStream，该Stream包含了generateFileMountedOnSDcard()方法用以生成待存储的文件。

功能的模型层由包model中的Note类和WorkingNote类分别实现，Note类定义了便签类的抽象参数，WorkingNote类定义了对便签操作的方法（如setBgColorId()、setCheckListMode()等）。

****功能的数据层由包data中的Notes类实现，Notes类具体定义了便签类的具体参数，将存储的便签数据向上传递。

* 1. **类的作用**
     1. **界面层相关的类**

**1、类名：NoteListActivity**

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：**NoteListActivity类的成员

|  |  |
| --- | --- |
| **成员** | **说明** |
| FOLDER\_NOTE\_LIST\_QUERY\_TOKEN | 文件夹便签表查询标识常量，传递查询信号 |
| FOLDER\_LIST\_QUERY\_TOKEN | 文件夹表查询标识常量，传递查询信号 |
| MENU\_FOLDER\_DELETE | 菜单文件夹删除标识常量，传递删除信号 |
| MENU\_FOLDER\_VIEW | 菜单文件夹查看标识常量，传递查看信号 |
| MENU\_FOLDER\_CHANGE\_NAME | 菜单文件夹改名标识常量，传递改名信号 |
| PREFERENCE\_ADD\_INTRODUCTION | 访问偏好设置字符串键，应用的介绍或配置信息 |
| ListEditState | 一个枚举类型成员，表示便签列表的编辑状态，包括便签列表、子文件夹和通话记录文件夹 |
| mState | 一个类成员变量，表示当前的列表编辑状态 |
| 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 | 便签列表的适配器，用于填充便签数据到界面 |
| mNotesListView | 便签列表的ListView控件，以列表的形式展示具体数据内容，并且根据数据的长度自适应屏幕显示 |
| mAddNewNote | 一个按钮类对象，用于添加新便签 |
| mDispatch | 一个布尔值，用于跟踪事件的分发状态 |
| mOriginY | 一个整数变量，用于记录Y轴坐标的值 |
| mDispatchY | 一个整数变量，用于跟踪Y轴坐标的值 |
| mTitleBar | 一个文本视图，用于显示标题栏的文本内容 |
| mCurrentFolderId | 一个长整数变量，表示当前所在文件夹的ID |
| mContentResolver | 一个ContentResolver对象，用于访问应用数据 |
| mModeCallBack | 一个回调对象，用于处理特定模式下的操作 |
| TAG | 一个用于日志记录的标签 |
| NOTES\_LISTVIEW\_SCROLL\_RATE | 一个整数常量，表示便签列表的滚动速率 |
| mFocusNoteDataItem | 一个便签数据项，用于跟踪当前焦点的便签 |
| NORMAL\_SELECTION | 一个用于数据库查询的字符串，表示普通数据筛选条件 |
| ROOT\_FOLDER\_SELECTION | 一个用于数据库查询的字符串，表示根文件夹的筛选条件 |
| REQUEST\_CODE\_OPEN\_NODE | 请求打开结点标识码常量 |
| REQUEST\_CODE\_NEW\_NODE | 请求新建结点标识码常量 |

NoteListActivity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 |
| onActivityResult(int requestCode, int resultCode, Intent data) | requestCode 是请求码，用于标识请求的来源resultCode是结果码，表示操作是否成功；data是包含返回数据的Intent对象 | 用于处理从其他Activity返回的结果，例如打开便签或创建新便签后的操作 | 处理Activity返回结果的方法，当其他Activity完成操作后返回结果时会调用此方法；否则调用父类onActivityResult方法来处理其他情况 |
| setAppInfoFromRawRes() | / | 初始化应用程序的介绍信息 | 首先获取SharedPreferences对象，检查是否已经插入过介绍信息，如果没有插入过则执行以下操作：从应用的原始资源（R.raw.introduction）中读取介绍信息；创建一个新的便签并将读取的介绍信息设置为便签内容；保存便签，并将插入介绍信息的标志设置为true，以避免重复插入 |
| onStart() | / | Activity的生命周期方法，在Activity即将可见时调用 | 调用startAsyncNotesListQuery()方法开始异步查询便签列表数据 |
| initResources() | / | 初始化Activity的资源和视图 | 初始化了mContentResolver、mBackgroundQueryHandler、mCurrentFolderId等成员变量，以及便签列表的视图（mNotesListView）和适配器（mNotesListAdapter）等 |
| updateMenu() | / | 更新多选模式下的操作菜单 | selectedCount 表示已选择的项的数量，通过更新下拉菜单的标题和全选/取消全选菜单项的状态 |
| onPrepareActionMode(ActionMode mode, Menu menu) | ActionMode mode：表示当前的多选模式操作栏；Menu menu：表示多选操作栏的菜单 | 操作模式即将显示时调用的方法，预加载 | 不加载 |
| onActionItemClicked(ActionMode mode, MenuItem item) | ActionMode mode：表示当前的多选模式操作栏；MenuItem item：表示被点击的菜单项 | 处理操作模式下菜单项的点击事件 | 不加载 |
| onDestroyActionMode(ActionMode mode) | ActionMode mode：表示当前的多选模式操作栏 | 在操作模式结束时调用的方法；用于清理多选模式相关的设置，恢复正常模式 | 清楚索引，修改相关设置 |
| finishActionMode() | / | 结束操作模式 | 调用mActionMode.finish() |
| onItemCheckedStateChanged(ActionMode mode, int position, long id, boolean checked) | ActionMode mode：表示当前的多选模式操作栏；int position：表示被选中或取消选中的便签项的位置；long id：表示被选中或取消选中的便签项的ID；boolean checked：表示便签项是否被选中 | 用于更新已选择项的状态和更新菜单 | 调用NitesListAdapter.setCheckedItem() |
| onMenuItemClick(MenuItem item) | MenuItem item：表示被点击的菜单项 | 处理下拉菜单中的菜单项点击事件 | 根据菜单项的ID执行不同的操作，如删除或移动已选择的便签 |
| startAsyncNotesListQuery() | / | 异步查询并加载便签列表数据 | 使用 mBackgroundQueryHandler 发起异步查询，根据当前文件夹ID来加载相应的便签数据 |
| showFolderListMenu(Cursor cursor) | cursor 是包含文件夹数据的光标对象 | 用于显示文件夹列表的对话框，并允许用户选择将便签移动到哪个文件夹 | 调用 DataUtils.batchMoveToFolder 方法将选定的便签移动到所选文件夹 |
| createNewNote() | / | 创建新的便签 | 创建一个新的便签编辑界面的 Intent，并指定当前文件夹ID |
| batchDelete() | / | 批量删除便签或将它们移动到回收站 | 异步执行的任务，根据同步模式，直接删除便签 |
| deleteFolder(long folderId) | folderId 表示要删除的文件夹的ID | 删除文件夹及其相关操作 | 根据同步模式，直接删除文件夹 |
| openNode(NoteItemData data) | data是一个 NoteItemData对象，表示要打开的便签数据 | 打开便签 | 创建一个 Intent 以查看指定便签的内容 |
| openFolder(NoteItemData data) | 同上 | 打开文件夹 | 根据所选文件夹的ID切换当前文件夹，加载该文件夹的便签列表 |
| onClick(View v) | v是触发点击事件的视图 | 处理点击事件 | 调用 createNewNote() 方法创建新便签 |
| showSoftInput() | / | 显示软键盘 | 调用相关方法 |
| hideSoftInput(View view) | view 是一个视图对象，用于获取窗口令牌以隐藏软键盘 | 隐藏软键盘 | 调用相关方法 |
| showCreateOrModifyFolderDialog(boolean create) | create 是一个布尔值，表示是创建文件夹还是修改文件夹 | 显示创建或修改文件夹的对话框 | 弹出一个弹窗，将输入参数传入函数 |
| onBackPressed() | / | 处理返回按钮的点击事件 | 进入switch-case，根据相关条件执行相关操作 |
| updateWidget(int appWidgetId, int appWidgetType) | appWidgetId 表示小部件的ID，appWidgetType 表示小部件的类型 | 更新小部件 | 根据小部件的类型（2x2 或 4x4），构建适当的 Intent 并发送广播以更新小部件 |
| onContextMenuClosed(Menu menu) | menu表示菜单对象 | 当上下文菜单关闭时的回调 | 移除长按文件夹后的上下文菜单监听器 |
| onContextItemSelected(MenuItem item) | item表示菜单项目对象 | 处理上下文菜单中的选项 | 根据用户的选择，执行不同的操作，如查看文件夹、删除文件夹或更改文件夹名称 |
| onPrepareOptionsMenu(Menu menu) | menu表示菜单对象 | 在准备选项菜单时的回调 | 根据当前应用状态（便签列表、子文件夹、通话记录文件夹），设置不同的选项菜单 |
| onOptionsItemSelected(MenuItem item) | item表示菜单项目对象 | 处理选项菜单中的选项选择 | 根据用户选择执行不同的操作，如创建新文件夹、导出便签到文本文件、启动同步或设置 |
| onSearchRequested() | / | 处理搜索请求 | 启动搜索操作，用于搜索便签 |
| **exportNoteToText()** | **/** | **将便签导出为文本文件** | **使用 BackupUtils 类执行导出操作，并根据结果显示相应的对话框提示** |
| isSyncMode() | / | 检查是否启用了同步模式 | 根据用户的设置检查是否配置了同步帐户 |
| startPreferenceActivity() | / | 启动应用的设置界面 | 跳转到设置界面，允许用户配置应用的各种参数 |
| startQueryDestinationFolders() | / | 开始查询可作为目标的文件夹列表 | 构建查询，获取可以移动便签的文件夹列表，并在查询完成后显示选择目标文件夹的对话框 |
| onItemLongClick(AdapterView<?> parent, View view, int position, long id) | parent：表示触发长按事件的父级适配器视图; view：表示用户长按的特定列表项视图; position：表示被长按的列表项在适配器中的位置; id：表示被长按的列表项的唯一标识符 | 处理长按列表项的事件 | 如果被长按的列表项是文件夹（Notes.TYPE\_FOLDER），则将文件夹上下文菜单创建监听器设置为 mFolderOnCreateContextMenuListener。这样，当用户点击该文件夹时，将弹出文件夹上下文菜单 |

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

1. **类的作用：**

一个备份工具类，能够处理备份和导出便签数据到文本文件的功能，用于导出Android便签应用程序数据到文本文件的工具类

1. **类的精读：**

BackupUtils类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| TAG | 用于日志记录的标签，通常用于在应用中标识日志消息的来源 |
| sInstance | 用于实现单例模式的静态成员变量，用于存储BackupUtils的单一实例 |
| STATE\_SD\_CARD\_UNMOUONTED | 表示SD卡当前未挂载的状态 |
| STATE\_BACKUP\_FILE\_NOT\_EXIST | 表示备份文件不存在的状态 |
| STATE\_DATA\_DESTROIED | 表示数据损坏或格式不正确的状态 |
| STATE\_SYSTEM\_ERROR | 表示备份或恢复操作中发生运行时异常的状态 |
| STATE\_SUCCESS | 表示备份或恢复操作成功完成的状态 |
| mTextExport | TextExport类的实例，用于处理将便签数据导出为文本的操作 |
| TextExport.TEXT\_FORMAT | 存储用于格式化文本导出的字符串数组 |
| TextExport.mContext | 用于访问应用的资源和内容提供器 |
| TextExport. mFileName | 导出的文本文件名 |
| TextExport. mFileDirectory | 导出的文本文件目录 |

BackupUtils类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数解释 | 方法作用 | 实现过程 |
| getInstance(Context context) | 传入一个Context参数，用于初始化该实例 | 获取BackupUtils的单例实例 | 传入一个 Context参数，获取单例实例 |
| TextExport(Context context) | 一个 Context 参数 | 初始化 TextExport 实例 | / |
| externalStorageAvailable() | / | 检查外部存储是否可用 | / |
| exportToText() | / | 导出便签数据为文本文件的方法 | 调用TextExport对象的函数 |
| getExportedTextFileName() | / | 获取导出的文本文件名 | 调用TextExport对象的函数 |
| getExportedTextFileDir() | / | 获取导出的文本文件目录 | 调用TextExport对象的函数 |
| TextExport. getFormat(int id) | id表示传入的id | 根据给定的格式ID返回对应的格式化字符串 | 返回为该id的格式化字符串 |
| TextExport.exportFolderToText(String folderId, PrintStream ps) | folderId是文件夹的id，PrintStream输出流 | 导出指定文件夹的便签到文本 | 创建文件夹游标，游标移动到传入id的文件夹 |
| TextExport. exportNoteToText(String noteId, PrintStream ps) | noteId便签的ID；PrintStream输出流 | 导出指定便签的内容到文本 | 创立数据游标，根据传入的id移动到该便签，开始导出 |
| TextExport. exportToText() | / | 导出便签数据为文本文件 | 获取文件夹id，调用exportFolderToText方法；获取便签id，调用exportNoteToText方法 |
| TextExport. getExportToTextPrintStream() | / | 获取一个 PrintStream 对象 | 获取输出地址，创建备份文件，初始化一个ps对象 |
| generateFileMountedOnSDcard(Context context, int filePathResId, int fileNameFormatResId) | 一个Context对象，一个文件路径资源ID；一个文件名格式资源 ID | 生成一个在SD卡上的文件用于存储导出的数据 | 将SD卡的根目录路径追加到 sb 中；将资源ID filePathResId 对应的字符串追加到 sb 中 |

* + 1. **模型层相关的类**

**1、 类名：**Note

1. **类的作用：**

管便签（notes）的数据，包括创建、编辑、同步等操作。它维护便签的元数据和文本数据以及通话数据，并将这些数据同步到Android应用的数据库中

1. **类的精读：**

Note类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNoteDiffValues | 一个ContentValues对象，用于存储便签的元数据的差异 |
| mNoteData | 一个NoteData对象，用于管理便签的文本数据和通话数据 |
| TAG | 一个用于日志记录的常量字符串 |

Note类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数解释 | 方法作用 | 实现过程 |
| getNewNoteId(Context context, long folderId) | context：Android应用程序的上下文；folderId表示便签所属的文件夹的ID | 创建一个新的便签ID并插入数据库中，返回新便签的ID | 创建一个ContentValues对象values，用于存储便签的元数据，初始化values，从URI中提取新便签的ID并返回 |
| Note() | / | 构造函数，初始化mNoteDiffValues和mNoteData对象 | / |
| setNoteValue(String key, String value) | String key：元数据的键名；String value：元数据的键值 | 设置便签元数据的键值对，用于标记元数据的本地修改 | 将键值对存储到mNoteDiffValues，并更新LOCAL\_MODIFIED和MODIFIED\_DATE的值 |
| setTextData(String key, String value) | String key：文本数据的键名；String value：文本数据的键值 | 设置文本数据的键值对 | 将键值对存储到mNoteData的文本数据中 |
| setTextDataId(long id) | id：文本数据的ID | 设置文本数据的ID | 将ID存储到mNoteData中 |
| getTextDataId() | / | 获取文本数据的ID | 返回mNoteData中的文本数据ID |
| setCallDataId(long id) | id：通话数据的ID | 设置通话数据的ID | 将ID存储到mNoteData中 |
| setCallData(String key, String value) | key：通话数据的键名；value：通话数据的键值 | 设置通话数据的键值对 | 将键值对存储到mNoteData的通话数据中 |
| isLocalModified() | / | 判断便签是否在本地有修改 | 检查mNoteDiffValues和mNoteData是否包含本地修改的数据，如果有则返回true，否则返回false |
| syncNote(Context context, long noteId) | context：Android应用程序的上下文；noteId：要同步的便签的ID | 同步便签到数据库中 | 如果没有本地修改，直接返回true表示同步成功；如果有本地修改，首先尝试更新便签的元数据 |
| NoteData. isLocalModified() | / | 判断文本数据或通话数据是否在本地有修改 | / |
| NoteData.setTextDataId(long id) | id表示传入的id | 设置文本数据的ID | / |
| NoteData. setCallDataId(long id) | id表示传入的id | 设置通话数据的ID | / |
| NoteData. setCallData(String key, String value) | String key：元数据的键名；String value：元数据的键值 | 设置通话数据的键值对 | / |
| NoteData. setTextData(String key, String value) | String key：元数据的键名；String value：元数据的键值 | 设置文本数据的键值对 | / |
| pushIntoContentResolver(Context context, long noteId) | context：Android应用程序的上下文；noteId：要同步的便签的ID | 将文本数据或通话数据同步到数据库中 | 根据是否存在本地修改，构建数据库操作，然后执行同步操作 |

* 1. **类名：**WorkingNote

1. **类的作用：**

定义创建、加载、保存和管理便签一些属性的具体操作，包括文本内容、提醒时间、背景颜色等属性。允许用户创建新的便签、加载已有的便签，以及保存对便签的修改。

1. **类的精读：**

WorkingNote类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| mNote | Note对象，用于管理便签的元数据和内容 |
| mNoteId | 便签的唯一标识符 |
| mContent | 便签的文本内容 |
| mMode | 便签的模式 |
| mAlertDate | 便签的提醒日期 |
| mModifiedDate | 便签的最后修改日期 |
| mBgColorId | 便签的背景颜色ID |
| mWidgetId | 小部件（Widget）的ID |
| mWidgetType | 小部件类型 |
| mFolderId | 便签所属的文件夹ID |
| mContext | Android应用程序的上下文 |
| TAG | 用于记录日志的标签 |
| mIsDeleted | 表示便签是否已被标记为删除 |
| mNoteSettingStatusListener | 便签设置状态的监听器 |
| DATA\_PROJECTION | 查询数据时使用的投影 |
| NOTE\_PROJECTION | 查询便签时使用的投影 |

WorkingNote类的方法

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数解释 | 方法作用 | 实现过程 |
| WorkingNote(Context context, long folderId) | context：Android应用程序的上下文；folderId：便签所属的文件夹ID | 构造函数，用于创建一个新的工作便签 | / |
| WorkingNote(Context context, long noteId, long folderId) | context：Android应用程序的上下文；noteId：要加载的便签的ID；folderId：便签所属的文件夹ID | 构造函数，用于加载已有的工作便签 | / |
| loadNote() | / | 用于加载便签的元数据信息 | 创建游标，游标移动到制定，获取属性值并赋给对应便签属性 |
| loadNoteData() | / | 用于加载便签的内容数据 | 创建游标，游标移动到制定，获取文本内容并赋给 |
| createEmptyNote(Context context, long folderId, int widgetId, int widgetType, int defaultBgColorId) | context：Android应用程序的上下文；folderId：便签所属的文件夹ID；widgetId：小部件的ID；widgetType：小部件类型；defaultBgColorId：默认的背景颜色ID | 创建一个新的空白便签 | 创建一个新的工作便签，并设置其属性，包括背景颜色、小部件ID等 |
| load(Context context, long id) | context：Android应用程序的上下文；id：要加载的便签的ID | 加载现有的工作便签 | 根据传入的条件查找便签，并加载 |
| saveNote() | / | 保存便签的方法，包括新建便签和已有便签的保存 | 判断便签是否值得保存；如果便签尚未存在于数据库中，创建新便签并获得其ID；同步便签的元数据和内容数据到数据库 |
| existInDatabase() | / | 判断便签是否已存在于数据库中 | 判断NoteId是否存在 |
| isWorthSaving() | / | 判断便签是否值得保存 | 考虑是否已删除、是否存在文本内容和是否有本地修改 |
| setOnSettingStatusChanged Listener(NoteSettingChangedListener l) | 便签属性改变监听器 | 设置便签设置状态变化的监听器 | / |
| setAlertDate(long date, boolean set) | date：提醒日期的时间戳；set：是否设置提醒 | 设置提醒日期 | 如果提醒日期不同于当前日期，更新提醒日期并通知监听器 |
| markDeleted(boolean mark) | mark：是否标记为已删除 | 标记便签是否已删除 | 标记便签是否已删除，同时通知监听器 |
| setBgColorId(int id) | id：背景颜色的ID | 设置背景颜色的ID | 如果背景颜色ID不同于当前ID，更新背景颜色ID并通知监听器 |
| setCheckListMode(int mode) | mode：模式 | 设置便签的清单模式 | 如果模式不同于当前模式，更新模式并通知监听器 |
| setWidgetType(int type) | type：小部件类型 | 设置小部件的类型 | 确定widget类型是否变化，变化了就更新widget参数 |
| setWidgetId(int id) | id：小部件的ID | 设置小部件的ID | 确定widget的id是否变化，更新widget |
| setWorkingText(String text) | text：文本内容 | 设置便签的文本内容 | 如果输入内容和原文本不一致则进行内容更新 |
| convertToCallNote(String phoneNumber, long callDate) | phoneNumber：通话记录的电话号码；callDate：通话日期的时间戳 | 将便签转换为通话记录便签 | 将传入的电话号码设置为电话数据 |
| hasClockAlert() | / | 判断便签是否有提醒设置 | / |
| getContent() | / | 获取便签的文本内容 | / |
| getAlertDate() | / | 获取便签的提醒日期 | / |
| getModifiedDate() | / | 获取便签的最后修改日期 | / |
| getBgColorResId() | / | 获取背景颜色的资源ID | / |
| getBgColorId() | / | 获取背景颜色的ID | / |
| getTitleBgResId() | / | 获取标题栏背景颜色的资源ID | / |
| getCheckListMode() | / | 获取便签的清单模式 | / |
| getNoteId() | / | 获取便签的ID | / |
| getFolderId() | / | 获取便签所属的文件夹ID | / |
| getWidgetId() | / | 获取小部件的ID | / |
| getWidgetType() | / | 获取小部件的类型 | / |
| NoteSettingChangedListener | / | 内部接口，定义了便签设置状态变化的监听器方法 | / |

* + 1. **数据层相关的类**

**1、类名：Notes**

1. **类的作用：**

定义各种便签常量、数据模型以及相关URI，以便在Android应用中操作便签和相关数据；定义与便签（Notes）和相关数据操作有关的常量、数据模型以及与数据操作相关的URI（Uniform Resource Identifier）；提供了一种组织便签和相关信息的结构，并定义了用于访问和操作这些数据的标准接口；定义了数据结构类型。

1. **类的精读：**

Note类的成员

|  |  |
| --- | --- |
| 成员 | 说明 |
| AUTHORITY | 定义了用于访问数据的内容提供者的权限 |
| TAG | 用于标识日志输出的标签 |
| TYPE\_NOTE | 普通便签的类型常量 |
| TYPE\_FOLDER | 文件夹的类型常量 |
| TYPE\_SYSTEM | 系统的类型常量 |
| ID\_ROOT\_FOLDER | 标识根文件夹的常量 |
| ID\_TEMPARAY\_FOLDER | 标识临时文件夹的常量 |
| ID\_CALL\_RECORD\_FOLDER | 标识通话记录文件夹常量 |
| ID\_TRASH\_FOLER | 标记回收站文件夹常量 |
| INTENT\_EXTRA\_ALERT\_DATE | 在Intent中作为键，传递提醒日期信息 |
| INTENT\_EXTRA\_BACKGROUND\_ID | 在Intent中作为键，用来传递背景颜色的标识符 |
| INTENT\_EXTRA\_WIDGET\_ID | 在Intent中作为键，用来传递小部件的ID |
| INTENT\_EXTRA\_WIDGET\_TYPE | 在Intent中作为键，用来传递小部件的类型信息 |
| INTENT\_EXTRA\_FOLDER\_ID | 在Intent中作为键，用来传递文件夹的ID |
| INTENT\_EXTRA\_CALL\_DATE | 在Intent中作为键，传递通话记录的日期信息 |
| TYPE\_WIDGET\_INVALIDE | 用于表示无效的小部件类型 |
| TYPE\_WIDGET\_2X | 表示小部件为2x2大小 |
| TYPE\_WIDGET\_4X | 表示小部件为4x4大小 |
| DataConstants. NOTE | 表示文本便签的MIME类型 |
| DataConstants. CALL\_NOTE | 表示通话记录的MIME类型 |
| CONTENT\_NOTE\_URI | 表示用于查询便签的URI |
| CONTENT\_DATA\_URI | 表示用于查询数据的URI |
| NoteColumns.ID | 代表唯一行的ID |
| NoteColumns.PARENT\_ID | 代表便签或文件夹的父项的ID，组织便签的层次 |
| NoteColumns.CREATED\_DATE | 代表笔记或文件夹的创建日期 |
| NoteColumns.MODIFIED\_DATE | 代表笔记或文件夹的最后修改日期 |
| NoteColumns.ALERTED\_DATE | 代表提醒日期 |
| NoteColumns.SNIPPET | 代表笔记的摘要或文件夹的名称 |
| NoteColumns.WIDGET\_ID | 代表与笔记关联的小部件的ID |
| NoteColumns.WIDGET\_TYPE | 代表小部件的类型 |
| NoteColumns.BG\_COLOR\_ID | 代表笔记的背景颜色的ID |
| NoteColumns.HAS\_ATTACHMENT | 代表笔记是否具有附件 |
| NoteColumns.NOTES\_COUNT | 代表文件夹内包含的笔记数量 |
| NoteColumns.TYPE | 代表文件的类型 |
| NoteColumns.SYNC\_ID | 代表最后同步的ID |
| NoteColumns.LOCAL\_MODIFIED | 用于指示是否在本地修改 |
| NoteColumns.ORIGIN\_PARENT\_ID | 代表在移动到临时文件夹之前的原始父项ID |
| NoteColumns.GTASK\_ID | 代表Google任务（Google Task）的ID |
| NoteColumns.VERSION | 代表版本代码 |
| TextNote.MODE | 代表文本笔记的模式 |
| TextNote.MODE\_CHECK\_LIST | 表示文本笔记的清单模式 |
| TextNote.CONTENT\_TYPE | 用于标识文本笔记的MIME类型 |
| TextNote.CONTENT\_ITEM\_TYPE | 用于标识单个文本笔记的MIME类型 |
| TextNote.CONTENT\_URI | 表示访问文本笔记数据的URI |
| CallNote.CALL\_DATE | 代表通话记录的通话日期 |
| CallNote.PHONE\_NUMBER | 代表通话记录的电话号码 |
| CallNote.CONTENT\_TYPE | 标识通话记录的MIME类型 |
| CallNote.CONTENT\_ITEM\_TYPE | 标识单个通话记录的MIME类型 |
| CallNote.CONTENT\_URI | 表示访问通话记录数据的URI |

Notes类没有方法