1. 主Activity .tinnoui.TinnoUIViewPager

Tab的结构图：

**TinnoUIViewPager**

**CategoriesBrowseActivity**

**PhoneStroageActivity**

Activity

FileManagerOperationActivity

AbsBaseActivity

Activity

1. 分类显示是怎么刷新的？

自定义了一个广播"android.filemanager.action.updatestorage"，该activity中动态注册的这个广播，

当TinnoUIViewPager切换时，MyOnPageChangeListener会监听到切换到该activity，然后发出这个广播，该activity接收到这个广播后，执行刷新操作。执行刷新操作的方法是scannerStorage。

1. 存储设备的信息是怎么刷新的

触发条件：

* Intent.ACTION\_MEDIA\_MOUNTED
* Intent.ACTION\_MEDIA\_UNMOUNTED
* Intent.ACTION\_MEDIA\_EJECT

调用adapter notifyDataSetChanged 刷新

StorageStatusAdapter

重新填充相关结构

mMeasureArray

mMearsureReceiverArray

mMearsureInfoArray 保存了挂载点的信息

重新获取挂载的节点

MountPointManager 的init(Context context) 方法

接收到广播

注册以上广播

mMountReceiver = MountReceiver.registerMountReceiver(getApplicationContext()); mMountReceiver.registerMountListener(mMountListener);

1. 与CategoriesBrowseActivity有关的结构

CategoriesConfig

CategoryAdapter

GridView

ListView

StorageStatusAdapter

1. 文件列表是怎么加载的

入口showDirectoryContent

FileManagerService的listFiles方法

创建task时有个参数OperationEventListener listener

创建ListFileTask 对象

Task执行周期中，会对应调用listener的方法：

onPreExecute -> mListener.onTaskPrepare();

onCancelled -> mListener.onTaskResult(…)

onProgressUpdate -> mListener.onTaskProgress(…)

onPostExecute ->mListener.onTaskResult(…)

执行ListFileTask

Listener 的onTaskResult

填充FileInfoManager 的mShowFilesInfoList

Adapter中的数据与其关联

FileInfoManager 的loadFileInfoList

mAdapter.notifyDataSetChanged();

1. 文件管理中回调的使用
2. MountReceiver 中的接口interface MountListener

public interface MountListener {

void onMounted(String mountPoint);

void onUnMounted(String unMountPoint);

void onEjected(String unMountPoint);

void onSdSwap();

}

这4个方法在MountReceiver的onReceive方法中的调用

if (INTENT\_SD\_SWAP.equals(action)) {

……

listener.onSdSwap();

……

}

if (Intent.ACTION\_MEDIA\_MOUNTED.equals(action)) {

…..

listener.onMounted(mountPoint);

…..

} else if (Intent.ACTION\_MEDIA\_UNMOUNTED.equals(action)) {

…..

listener. onUnMounted (mountPoint);

…..

}else if (Intent.ACTION\_MEDIA\_EJECT.equals(action)) {

…..

listener. onEjected (mountPoint);

…..

}

实现这个接口的一个地方AbsBaseActivity

注册：

mMountReceiver = MountReceiver.registerMountReceiver(getApplicationContext());

mMountReceiver.registerMountListener(this);//this是实现该接口的类

4个接口方法的实现

public void onSdSwap() {

……

}

public void onUnMounted(String unMountPoint) {

……

}

public void onEjected(String unMountPoint) {

……

}

public void onMounted(String mountPoint) {

……

}

当对应的广播发生时，便会执行AbsBaseActivity中对应的4个方法

2）FileManagerService 中的接口

public interface OperationEventListener {

……

void onTaskPrepare();

void onTaskProgress(ProgressInfo progressInfo);

void onTaskResult(int result);

}

注册监听器的过程，以显示文件列表为例：

带有listener参数

调用listFiles 方法

继承自自定义类BaseAsyncTask

传入listener参数

新建ListFileTask

传入listener参数

ListFileTask的构造方法中 调用基类的构造方法

继承自系统的AsyncTask

保存了listener参数

BaseAsyncTask的构造方法

至此，listener以到了目的地，调用接口方法的地方如下：

protected void onPreExecute() {

……

mListener.onTaskPrepare();

……

}

protected void onPostExecute(Integer result) {

……

mListener.onTaskResult(result);

……

}

protected void onCancelled() {

……

mListener.onTaskResult(OperationEventListener.ERROR\_CODE\_USER\_CANCEL);

……

}

protected void onProgressUpdate(ProgressInfo... values) {

……

mListener.onTaskProgress(values[0]);

……

}

AbsBaseActivity中的ListListener 如下

调用：

showDirectoryContent 方法中的

mService.listFiles(this.getClass().getName(), mCurrentPath, new ListListener());

定义：

protected class ListListener implements FileManagerService.OperationEventListener {

public void onTaskResult(int result) {……}

public void onTaskPrepare() {……}

public void onTaskProgress(ProgressInfo progressInfo) {……}

}

这种采用接口的方式的好处，可以使代码实现高度的复用，同样使用者可以不用关心其他的细节，只关注对应的状态即可