同学们，晚上好，欢迎大家来到咱们享学课堂 我是本节课讲师Zero老师，能听到声音看到画面的同学 请刷波鲜花或Q个1，让Zero老师确保网络没问题

好了，我们的课题是XXX，如果有需要我们享学课堂晚期视频资料的 可以咨询伊娜老师，VIP咨询也可以加伊娜老师的QQ， 有技术问题 可以加我的QQ

好了，Zero老师先在这做个互动，有第一次来到享学课堂和第一次来听Zero老师讲课的同学，请刷波鲜花或Q个1

因为还是有很多同学第一次来听zero老师的课，那么zero老师先来做个自我介绍

当然 我们享学课堂还有很多其他优秀的老师

解释导航的含义

目录介绍

activity四种启动模式，然后做下小结

然后将因为activity跳转很麻烦 所以用fragment

然后讲fragment的问题

再对比activity vs fragment

引出更好的解决方案 navigation

然后先介绍下jetpack

然后讲navigation

navigation跳转是如何进行的

NavHostFragment -> NavController -> Navinflater -> 解析navigation.xml -> Destination -> navigate() -> activity/fragment的navigate

1. 口头禅 注意不要有
2. 记得录屏
3. 记得课堂互动

在正式的介绍Activity的启动模式之前，我们首先要了解一些旁边的知识，这些知识如果说模糊不清，那么在讨论启动模式的时候会一头雾水（笔者亲身感悟）。

# 简介

1.一个应用程序通常会有多个Activity，这些Activity都有一个对应的action（如MainActivity的action），我们可以通过action来启动对应Activity（隐式启动）。

<action android:name="android.intent.action.MAIN" />

2.一个应用程序可以说由一系列组件组成，这些组件以进程为载体，相互协作实现App功能。

3.任务栈（Task Stack）或者叫退回栈（Back Stack）介绍：

3.1.任务栈用来存放用户开启的Activity。

3.2.在应用程序创建之初，系统会默认分配给其一个任务栈（默认一个），并存储根Activity。

3.3.同一个Task Stack，只要不在栈顶，就是onStop状态：

3.4.任务栈的id自增长型，是Integer类型。

3.5.新创建Activity会被压入栈顶。点击back会将栈顶Activity弹出，并产生新的栈顶元素作为显示界面（onResume状态）。

3.6.当Task最后一个Activity被销毁时，对应的应用程序被关闭，清除Task栈，但是还会保留应用程序进程（狂点Back退出到Home界面后点击Menu会发现还有这个App的框框。个人理解应该是这个意思），再次点击进入应用会创建新的Task栈。

## TaskAffinity

affinity是Activity内的一个属性（在ManiFest中对应属性为taskAffinity）。默认情况下，拥有相同affinity的Activity属于同一个Task中

Task也有affinity属性，它的affinity属性由根Activity（创建Task时第一个被压入栈的Activity）决定

在默认情况下（我们什么都不设置），所有的Activity的affinity都从Application继承。也就是说Application同样有taskAffinity属性

Application默认的affinity属性为Manifest的包名

## allowTaskReparenting

标记此属性的Activity实例所在的当前应用  在退居后台后，决定是否将标记的Activity从启动的Task移动到相同affinity的Task（需要和taskaffinity属性配合使用）：

true：可以移动。

false（默认）：不能移动

## alwaysRestainTaskState

首先这个属性，只针对根Activity起作用。

标记应用的Task师傅保持原来的状态：

true：总保持

false（默认）：不保证维持原来状态

这个我们这么解释：我们的App如果长期在后台带着不使用，我们的系统会对应用的Task进行清理，而清理过程中，如果我们对根Activity标记了这个属性，那么他会保留我们根Activity的状态。

这个没把饭举例子，相信大家是可以理解的。

## clearTaskOnLaunch

这个属性也只是针对根Activity起作用的。

每次唤醒应用是，是否清楚除跟Activity的所有Activity：

true：清楚

false（默认）：不清除

## finishOnTaskLaunch

这个属性跟clearTaskLaunch很像，只不过他只是销毁标记的Activity，而clearTaskOnLaunch是除了根Activity全部销毁。

## noHistory

标记的Activity（包括根Activity）跳转到其他Activity时，Task不将其压入栈。

这个我来解释一下，比如说我们从MainActivity跳转到TwoActivity，那么Task应该讲MainActivity压入栈，但是如果给MainActivity标记了noHistory属性，那么他就不会被压入栈。

standard：标准启动模式（默认启动模式），每次都会启动一个新的activity实例。

singleTop：单独使用使用这种模式时，如果Activity实例位于当前任务栈顶，就重用栈顶实例，而不新建，并回调该实例onNewIntent()方法，否则走新建流程。

singleTask：这种模式启动的Activity只会存在相应的Activity的taskAffinit任务栈中，同一时刻系统中只会存在一个实例，已存在的实例被再次启动时，会重新唤起该实例，并清理当前Task任务栈该实例之上的所有Activity，同时回调onNewIntent()方法。

singleInstance：这种模式启动的Activity独自占用一个Task任务栈，同一时刻系统中只会存在一个实例，已存在的实例被再次启动时，只会唤起原实例，并回调onNewIntent()方法。