

Backstack Activity, launchMode, intentFlags, taskAffinity

Activity Backstack

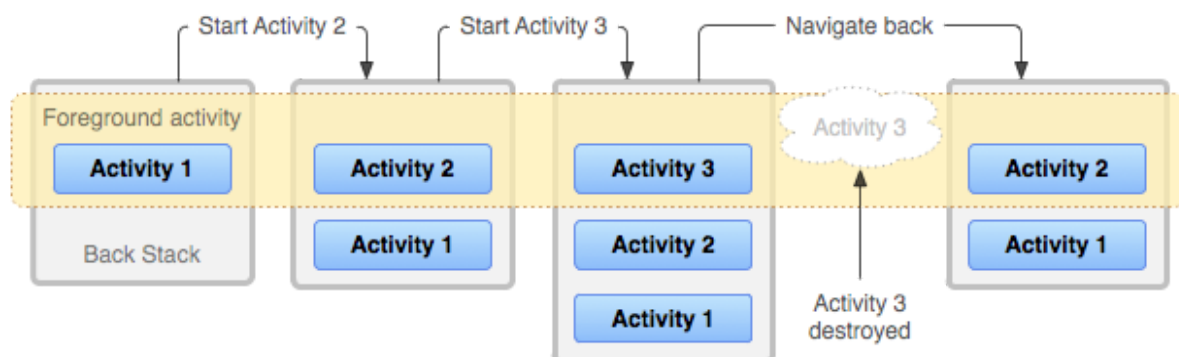
Все активити приложения, собираются в так называемый бэкстек активити. При запуске новой активити, оно добавляется на вершину стека, при нажатии кнопки назад – удаляется из стека.

<https://developer.android.com/guide/components/tasks-and-back-stack.html?hl=ru>
<https://blog.mindorks.com/android-task-and-back-stack-review-5017f2c18196>

В подавляющем большинстве приложения Android есть Activity. Часто бывает, что таких Activity очень много и нам нужно как-то переключаться между ними. Переходить на следующий экран, на предыдущий. В этом случае у нас должна быть какая-то сущность, которая запоминает в каком порядке мы переходили по экранам, чтобы при нажатии кнопки “назад”, она открыла верный экран. Такая сущность называется Back Stack. Она работает по принципу last in – first out или LIFO. Если открывается новая activity, то она добавляется на самый верх стека, а предыдущая activity уходит в stopped. Одна Activity может быть в стеке столько раз, сколько была запущена.

Также, есть такая сущность как task. Task – набор из нескольких Activity. И в каждом task есть свой back stack. Когда вы сворачиваете ваше приложение, ваш task уходит в фоновый режим. Но он обязательно восстановится, когда вы вернетесь в приложение. Но может случиться такое, что системе не будет хватать памяти и она “прибьет” ваш task, но это бывает крайне редко.

Так работает back stack:



Если нажать на кнопку back, то Activity будут удаляться из стека до тех пор, пока не останется одна корневая Activity. Если в стеке останется только корневая Activity, и вы нажмете на back, то приложение закроется и task умрет.

Все вышеописанное – это стандартное поведение Android. Но бывают случаи, когда вам нужно изменить поведение вашей task. Давайте посмотрим как это можно сделать.

Есть два способа, чтобы изменить работу вашего task:

- 1) установить атрибуты в манифесте в каждом Activity;
- 2) установить флаги для Intent, который запускает новую Activity.

Если вы будете использовать оба способа сразу, то знайте, что, при противоречии флагов из первого и второго способа, то система будет отдавать приоритет второму способу.

launchMode

Способ первый.

В манифесте для любой Activity можно задать атрибут launchMode. Вот значения, который он может принимать:

- **standard** (по умолчанию) — во время запуска Activity создается новый экземпляр в стеке. Activity может размещаться в стеке несколько раз
- **singleTop** — Activity может размещаться в стеке несколько раз. Новая запись появится если данная Activity не находится в вершине стека. Если она и так в вершине стека, то просто вызовется метод `onNewIntent()`, в который передается интент, из которого при необходимости можно извлечь данные.
- **singleTask** — Если экземпляра данной Activity нет ни в одном другом task, то создает новый task и устанавливает Activity корневой для него. Если Activity уже находится в каком-нибудь task, то откроется этот экземпляр и вызовется метод `onNewIntent()`. Activity станет корневой, и все кто был выше – удалятся. Может существовать только 1 экземпляр такой Activity.
- **singleInstance** — тоже что и `singleTask`, но для данной Activity всегда будет создаваться отдельный task и она будет в ней корневой. Т.е. любые Activity, запущенные из этой Activity, запустятся в отдельном task.

intentFlags

Способ второй.

Мы можем задавать флаги для Intent, который запускает новую Activity. Флаги более приоритетны, чем launchMode.

- **FLAG_ACTIVITY_NEW_TASK** — запускает Activity в новой task. Если уже существует task с экземпляром данной Activity, то этот task становится активным, и вызывается метод `onNewIntent()`.
- **FLAG_ACTIVITY_SINGLE_TOP** — если Activity находится в вершине стека, то вместо создания нового экземпляра в стеке вызывается метод `onNewIntent()`.
- **FLAG_ACTIVITY_CLEAR_TOP** — если экземпляр данной Activity уже существует в стеке данной task, то все Activity, находящиеся поверх нее разрушаются и этот экземпляр становится вершиной стека. Вызовется `onNewIntent()`

taskAffinity

Обычно все Activity нашего приложения работают в одном task. И мы можем изменять такое поведение и указывать, чтобы в одном приложении Activity работали в разных task, или Activity разных приложений работали в одном task. Для этого мы можем в манифесте для каждой Activity указывать название task параметром **taskAffinity**. Это строковое значение, которое не должно совпадать с названием package, т.к. стандартный task приложения называется именно как наш package. В общем случае данный параметр указывает, что Activity будет гарантированно открываться в своем отдельном task. Данный параметр актуален, если мы указываем флаг **FLAG_ACTIVITY_NEW_TASK** или устанавливаем для Activity атрибут `allowTaskReparenting=«true»`. Этот атрибут указывает, что Activity может перемещаться между tasks, который ее запустил и task, который указан в taskAffinity, если один из них становится активным.

Доп. чтение:

<https://developer.android.com/guide/components/tasks-and-back-stack.html?hl=ru>

