

文 FragmentTransaction与Fragment生命周期的关系

android fragment 食用淡水鱼 2014年08月27日发布

在Android中，对 `Fragment` 的操作都是通过 `FragmentTransaction` 来执行。而从 `Fragment` 的结果来看，`FragmentTransaction` 中对 `Fragment` 的操作大致可以分为两类：

- 显示： `add()` `replace()` `show()` `attach()`
- 隐藏： `remove()` `hide()` `detach()`

对于每一组方法，虽然最后产生的效果类似，但方法背后带来的副作用以及对 `Fragment` 的生命周期的影响都不尽相同。

add() vs. replace()

只有在 `Fragment` 数量大于等于2的时候，调用 `add()` 还是 `replace()` 的区别才能体现出来。当通过 `add()` 连续两次添加 `Fragment` 的时候，每个 `Fragment` 生命周期中的 `onAttach()`-`onResume()` 都会被各调用一次，而且两个 `Fragment` 的 `View` 会被同时 `attach` 到 `containerView`。

```
Fragment#1 *****onAttach()*****
Fragment#1 *****onCreate()*****
Fragment#1 *****onCreateView()*****
Fragment#1 *****onActivityCreated()*****
Fragment#1 *****onStart()*****
Fragment#1 *****onResume()*****
Fragment#2 *****onAttach()*****
Fragment#2 *****onCreate()*****
Fragment#2 *****onCreateView()*****
Fragment#2 *****onActivityCreated()*****
Fragment#2 *****onStart()*****
Fragment#2 *****onResume()*****
```

同样，退出 `Activity` 时，每个 `Fragment` 生命周期中的 `onPause()`-`onDetach()` 也会被各调用一次。

```
Fragment#1 *****onPause()*****
Fragment#2 *****onPause()*****
Fragment#1 *****onStop()*****
Fragment#2 *****onStop()*****
Fragment#1 *****onDestroyView()*****
Fragment#1 *****onDestroy()*****
Fragment#1 *****onDetach()*****
Fragment#2 *****onDestroyView()*****
Fragment#2 *****onDestroy()*****
Fragment#2 *****onDetach()*****
```

但当使用 `replace()` 来添加 `Fragment` 的时候，第二次添加会导致第一个 `Fragment` 被销毁，即执行第二个 `Fragment` 的 `onAttach()` 方法之前会先执行第一个 `Fragment` 的 `onPause()`-`onDetach()` 方法，同时 `containerView` 会 `detach` 第一个 `Fragment` 的 `View`。

```
Fragment#1 *****onAttach()*****
Fragment#1 *****onCreate()
Fragment#1 *****onCreateView()
Fragment#1 *****onActivityCreated()
Fragment#1 *****onStart()
Fragment#1 *****onResume()
Fragment#1 *****onPause()
Fragment#1 *****onStop()
Fragment#1 *****onDestroyView()
Fragment#1 *****onDestroy()
Fragment#1 *****onDetach()
Fragment#2 *****onAttach()
Fragment#2 *****onCreate()
Fragment#2 *****onCreateView()
Fragment#2 *****onActivityCreated()
Fragment#2 *****onStart()
Fragment#2 *****onResume()
```

show() & hide() vs. attach() & detach()

调用 `show()` & `hide()` 方法时，`Fragment` 的生命周期方法并不会被执行，仅仅是 `Fragment` 的 `View` 被显示或者隐藏。而且，尽管 `Fragment` 的 `View` 被隐藏，但它在父布局中并未被 `detach`，仍然是作为 `containerView` 的 `childView` 存在。相比较下，`attach()` & `detach()` 做的就更彻底一些。一旦一个 `Fragment` 被 `detach()`，它的 `onPause()`-`onDestroyView()` 周期都会被执行。

```
Fragment#2 *****onPause()
Fragment#2 *****onStop()
Fragment#2 *****onDestroyView()
```

同时 `Fragment` 的 `View` 也会被 `detach`。在重新调用 `attach()` 后，`onCreateView()`-`onResume()` 周期也会被再次执行。

```
Fragment#2 *****onCreateView()
Fragment#2 *****onActivityCreated()
Fragment#2 *****onStart()
Fragment#2 *****onResume()
```

remove()

其实看完上面的分析，`remove()` 方法基本也就明白了。相对应 `add()` 方法执行 `onAttach()`-`onResume()` 的生命周期，`remove()` 就是完成剩下的 `onPause()`-`onDetach()` 周期。

```
Fragment#2 *****onPause()
Fragment#2 *****onStop()
Fragment#2 *****onDestroyView()
Fragment#2 *****onDestroy()
Fragment#2 *****onDetach()
```

总结

除了上面这些核心的api外，`FragmentTransaction` 还提供了更多的方法以丰富 `Fragment` 的操作，如为 `Fragment`

的显示和隐藏添加动画。具体使用方法可以参阅api文档。

<http://developer.android.com/reference/android/app/FragmentTransaction.html>

2014年08月27日发布

0 推荐

收藏

你可能感兴趣的文章

[android ViewPager和Fragment](#) 6 收藏, 421 浏览

[Android Fragment简记](#) 8 收藏, 5.3k 浏览

[Android Fragment生命周期——多屏幕支持](#) 1.5k 浏览

讨论区

使用评论询问更多信息或提出修改意见，请不要在评论里回答问题

提交评论



评论支持部分 Markdown 语法： ****bold**** **_italic_** [link] (<http://example.com>) > 引用 `code` - 列表 。 ×
同时，被你 @ 的用户也会收到通知

本文隶属于专栏

[食用淡水鱼](#)

欢迎光临！

 食用淡水鱼
作者

关注专栏

相关收藏夹

 Android
9 个条目 | 0 人关注

分享扩散：

...

