

Recyclerview 更新日志（中文）

参考：[支持库修订历史记录](#)

1. [Android 支持库，修订版 24.2.0](#) (2016 年 8 月)

API 更新

- 新的 [DiffUtil](#) 类可以计算两个集合之间的差异，并分派适合 [RecyclerView.Adapter](#) 使用的更新操作列表。
- 添加了 [RecyclerView.OnFlingListener](#) 以支持对滚屏做出响应的自定义行为。
[SnapHelper](#) 类专门用于实现子视图吸附，[LinearSnapHelper](#) 类则对此实现进行了扩展，提供类似于 [ViewPager](#) 的中心对齐吸附。

2. [Android 支持库，修订版 23.3.0](#) (2016 年 4 月)

[v7 recyclerview 库](#)的变更：

- 修复了在可见项目范围缩小时 [RecyclerView](#) 不会调用滚动回调的问题。（[问题 200987](#)）
- 修复了 [RecyclerView](#) 处于线性布局、应用加权和包含图像时会冻结的问题。（[问题 203276](#)）
- 修复了 [OrientationHelper.getStartAfterPadding\(\)](#) 中发生崩溃的问题。（[问题 180521](#)）
- 修复了使用 `android:nestedScrollingEnabled` 时发生崩溃的问题。（[问题 197932](#)）

3. [Android 支持库，修订版 23.2.1](#) (2016 年 3 月)

[v7 recyclerview 库](#)的变更：

- 修复了与测量值规范方法有关的各种问题。（[问题 201856](#)）
- 缩短了计算布局或滚动时 [RecyclerView](#) 不允许适配器发生更改的锁定期间。（[问题 202046](#)）
- 修复了对视图之外的项目调用 [notifyItemChanged\(\)](#) 时发生崩溃的问题。（[问题 202136](#)）
- 修复了 [RecyclerView.LayoutManager](#) 在同一个测量值传递中添加和移除视图时发生崩溃的问题。（[问题 193958](#)）

4. [Android 支持库，修订版 23.2.0](#) (2016 年 2 月)

[v7 RecyclerView 库](#)的变更：

- [RecyclerView](#) 现在提供名称为 *AutoMeasure* 的选择功能，该功能可让 [RecyclerView.LayoutManager](#) 轻松地封装内容或处理由 [RecyclerView](#) 的父项提供的各种测量值规范。它支持 [RecyclerView](#) 的所有现有动画功能。
- 如果您有自定义的 [RecyclerView.LayoutManager](#)，请调用 `setAutoMeasureEnabled(true)` 以开始使用新的 *AutoMeasure* API。默认情况下，所有内置的 [RecyclerView.LayoutManager](#) 对象均会启用自动测量。
- [RecyclerView.LayoutManager](#) 不再忽略某些 [RecyclerView.LayoutParams](#) 设置，例如滚动方向上的 `MATCH_PARENT`。
注：这些解除的限制可能会导致布局中出现异常行为。请确保您指定正确的布局参数。
- 使用负载信息更新 [RecyclerView.ViewHolder](#) 时，[DefaultItemAnimator](#) 现在将停用动画变更。
- 现在，您可以修改 [ItemTouchHelper](#) 逃逸速度以控制滑动灵敏度。要使滑动更容易或更难，请替换 `getSwipeEscapeVelocity(float defaultValue)` 并修改 `defaultValue`。

5. [Android 支持库，修订版 23.1.1](#) (2015 年 11 月)

[v7 recyclerview 库](#)的变更：

- 修复了执行 `ItemTouchHelper` 实用工具类提供的滑动清除操作后添加项目时发生崩溃的问题。（[问题 190500](#)）

6. [Android 支持库，修订版 23.1.0](#) (2015 年 10 月)

[v7 recyclerview 库](#)的变更：

- 向 `ItemAnimator` 类中添加了改进的动画 API 以更好地进行自定义：
- 变更动画不再执行可启用项目内容动画的 `ViewHolder` 对象的两个副本。而且，`ItemAnimator` 对象将决定是重复使用相同的 `ViewHolder` 对象还是创建一个新对象。
- 新的信息记录 API 可以让 `ItemAnimator` 类灵活地在布局生命周期中的正确时间点收集数据，然后将这些信息传入动画回调。
- 为向后不兼容的 API 变更提供了简单的过渡计划：

- 如果您之前已扩展了 `ItemAnimator` 类，可以将基础类更改为 `SimpleItemAnimator`，如此一来，您的代码将像之前一样工作。
`SimpleItemAnimator` 类通过封装新的 API 提供旧的 API。
- 某些方法已从 `ItemAnimator` 类中移除。以下代码不再编译：

```
recyclerView.getItemAnimator().setSupportsChangeAnimations(false)
```

您可以将其替换为以下代码：

```
ItemAnimator animator = recyclerView.getItemAnimator();
if (animator instanceof SimpleItemAnimator) {
    ((SimpleItemAnimator) animator).setSupportsChangeAnimations(false);
}
```

7. [Android 支持库，修订版 22.1.0](#) (2015 年 4 月)

[v7 recyclerview 库](#)的变更：

- 添加了 [SortedList](#) 类，用于按列表顺序显示项目并提供列表变更通知。
- 添加了 [SortedListAdapterCallback](#) 类，此类可以将已排序的列表与 [RecyclerView.Adapter](#) 类绑定。

8. [Android 支持库，修订版 22](#) (2015 年 3 月)

[v7 recyclerview 库](#)的变更：

- 向 [RecyclerView](#) 类中添加了 [getlayoutPosition\(\)](#) 和 [getadapterPosition\(\)](#) 方法。
- 弃用了 [RecyclerView](#) 类中的 [getClassChildPosition\(\)](#) 和 [findViewHolderForPosition\(\)](#) 方法。
- 弃用了 [RecyclerView.ViewHolder](#) 类中的 [getPosition\(\)](#) 方法。
- 弃用了 [RecyclerView.LayoutParams](#) 类中的 [getViewPosition\(\)](#) 方法。

9. [Android 支持库，修订版 21.0.2](#) (2014 年 11 月)

[v7 recyclerview 库](#)的变更：

- 向 [RecyclerView](#) 类中添加
了 [TOUCH_SLOP_DEFAULT](#) 和 [TOUCH_SLOP_PAGING](#) 常量，用于支持分页的
触摸溢出配置。

10. [Android 支持库，修订版 21](#) (2014 年 10 月)

新 [v7 recyclerview 库](#) :

- 添加了 [RecyclerView](#) 小部件，它提供了灵活的列表视图，从而为大型数据集提供有限窗口。