

下面用中文详细说明 **React Native ScrollView** 做轮播（每次滚动 1 列）在安卓上横竖屏切换后索引错位 的常见原因与稳定解决方案。

## ✓ 问题本质

你现在遇到的典型症状：

- 竖屏滚到 `index = 5`
- 切到横屏后 `ScrollView` 保留了「像素级偏移位置」
- 由于横屏宽度变大，`contentOffset.x` 对应的 `index` 改变
- 结果：`index` 漂移、错位、对不上
- 安卓更明显，因为 RN 安卓不会自动重算 `contentOffset`

核心原因：`ScrollView` 的 `offset` 是像素单位，而你的索引是列单位，屏幕宽度变化后 `offset` 无效。

## ✓ 最可靠的解决方案：监听横竖屏变化，重新计算 `offset`

当屏幕尺寸变化后，你必须 基于最新屏幕宽度 重置滚动位置。

## ✓ 推荐实现方式（最稳定，不会错位）

### ✓ 1. 监听屏幕尺寸变化

RN 推荐用 `useWindowDimensions()` 或 `Dimensions.addListener`。

更稳的是 `useWindowDimensions`：

```
const { width, height } = useWindowDimensions();
```

每次横竖屏切换，组件会自动 re-render。

### ✓ 2. 在尺寸变化后，重新 `scrollTo` 到当前 `index`

核心代码：

```
useEffect(() => {
  if (!scrollRef.current) return;
  // 重新定位 offset
  scrollRef.current.scrollTo({
    x: currentIndex * width,
    animated: false,
  });
}, [width]); // 监听屏幕宽度变化
```

✅ 这样能保证：

- 不管之前 offset 是多少
- 屏幕变大变小之后
- 都会重新定位到正确 index

不会出现索引漂移。

### ✅ 3. 防止安卓 ScrollView 残留 offset 导致闪烁

安卓 ScrollView 会在 re-layout 后尝试保持之前的 offset，会导致闪跳。

解决方案：给 ScrollView 加以下属性：

```
<ScrollView
  ref={scrollRef}
  horizontal
  pagingEnabled
  onLayout={() => {
    // 安卓 onLayout 后强制对齐一次
    scrollRef.current?.scrollTo({
      x: currentIndex * width,
      animated: false,
    });
  }}
/>
```

### ✅ 4. 完整代码模板（可直接用）

```
import React, { useRef, useEffect, useState } from
"react";
import { ScrollView, View, useWindowDimensions } from
"react-native";
```

```

export default function Carousel({ data }) {
  const scrollRef = useRef(null);
  const { width } = useWindowDimensions();
  const [currentIndex, setCurrentIndex] = useState(0);

  const onMomentumScrollEnd = (e) => {
    const newIndex =
Math.round(e.nativeEvent.contentOffset.x / width);
    setCurrentIndex(newIndex);
  };

  useEffect(() => {
    if (scrollRef.current) {
      scrollRef.current.scrollTo({
        x: currentIndex * width,
        animated: false,
      });
    }
  }, [width]); // 横竖屏切换时会触发

  return (
    <ScrollView
      ref={scrollRef}
      horizontal
      pagingEnabled
      showsHorizontalScrollIndicator={false}
      onMomentumScrollEnd={onMomentumScrollEnd}
      onLayout={() => {
        scrollRef.current?.scrollTo({
          x: currentIndex * width,
          animated: false,
        });
      }}
    >
      {data.map((item, index) => (
        <View key={index} style={{ width }}>
          {item}
        </View>
      ))}
    </ScrollView>
  );
}

```

## ✅ 其它可选方案（按需求选择）

### ✅ 方案 A：横竖屏直接重置到 `index = 0`

最简单粗暴，但用户可能不喜欢体验。

```
useEffect(() => {  
    setCurrentIndex(0);  
}, [width]);
```

### ✅ 方案 B：给 `ScrollView` 设置 `removeClippedSubviews={false}`

可以减少安卓 re-layout 引起的跳动，但不能单独解决索引漂移。

### ✅ 方案 C：使用专门的轮播库（更稳定）

如果 `ScrollView` 麻烦，可以直接用：

✅ `react-native-snap-carousel`

✅ `react-native-reanimated-carousel`（官方推荐，性能最好）

它们都已经处理了横竖屏 `offset` 重置逻辑。

## ✅ 总结（最关键一句话）

安卓横竖屏切换后必须根据新的宽度重新计算 `offset`，否则 `index` 会漂移。

解决办法就是：

✅ 监听横竖屏变化

✅ 重新定位 `scrollTo(x = currentIndex * newWidth)`

这是最稳的方法。

如果你愿意，我可以帮你：

✅ 把你当前的代码贴出来，我直接帮你修改成稳定版本

✅ 或帮你改成无限循环轮播版本

✅ 或支持两列显示 & 每次滚动一列的版本