

你说的 `onMomentumScrollEnd` 在 WAP (react-native-web) 里不触发 —— 这是真的，而且是 RN Web 的老问题，核心原因是：

Web 的 ScrollView 根本没有“momentum scroll (惯性滚动)”这个事件模型，RN Web 模拟不了。

所以：

- iOS / Android 原生：有 `inertia` → 有 `momentum` → 有 `onMomentumScrollEnd`
- WAP / RN Web：没有 `momentum` 事件 → `onMomentumScrollEnd` 不会触发

🔥 那在 WAP 上应该用什么替代 `onMomentumScrollEnd`？

✓ 正确答案：监听 `onScroll` + 在滚动停止后手动触发“滚动结束”事件
(因为浏览器只有 scroll，没有 momentum)

💡 浏览器停止滚动判定 = “最后一次 scroll 后，N 毫秒没有再次触发 scroll”

常用方式：

- 每次 `onScroll` 调用时重置一个 `setTimeout`
- 超过 80~120ms 无 scroll → 判断为“滚动结束”

✅ 给你最稳的 WAP 兼容轮播方案（核心代码只有 12 行）

```
const scrollRef = useRef(null);
let scrollEndTimer = null;

const onScroll = (e) => {
  const x = e.nativeEvent.contentOffset.x;
```

```

// 每次滚动都清理延迟
if (scrollEndTimer) clearTimeout(scrollEndTimer);

// 100ms 内无滚动 → 视为 scrollEnd
scrollEndTimer = setTimeout(() => {
  handleScrollEnd(x);
}, 120);

const handleScrollEnd = (x) => {
  // 算最近页
  const page = Math.round(x / width);

  // 自动吸附
  scrollRef.current?.scrollTo({
    x: page * width,
    animated: true
  });
};

ScrollView:
<ScrollView
  ref={scrollRef}
  horizontal
  onScroll={onScroll}
  scrollEventThrottle={16}
>
  {children}
</ScrollView>

```

👍 为什么这是 WAP 最稳方案?

方案	WA P	原因
snapToInterval	✗	RN Web 没实现 scroll snapping
onMomentumScrollEnd	✗	Web 无 momentum event
onScrollEndDrag	✗	手松开并不代表滚动结束
“定时器检测滚动停止”	✓	兼容所有浏览器，实际项目中最常用

◎ 进一步改进：避免频繁吸附（优化体）

验)

加入“最小移动阈值”：

```
if (Math.abs(x - currentPage * width) < 5) return;
```

加入“最大惯性抖动过滤”：

```
if (Math.abs(x - lastX) < 1) return; // 忽略低速滚动
```

加入“人工阻止滑太多页”：

```
const maxPage = currentPage + 1;  
const minPage = currentPage - 1;
```

```
const clampedPage = Math.min(maxPage, Math.max(minPage,  
page));
```

📌 最终你会得到：

- ✓ 手动拖动顺畅
- ✓ 惯性自然
- ✓ 滚动停止自动吸附
- ✓ WAP/iOS/Android 全平台兼容
- ✓ 不卡顿
- ✓ 不需要 reanimated 或 gesture-handler

如果需要，我可以给你：

- 无限循环轮播 (WAP 兼容)
- 多列轮播 (两列/三列显示)
- 自动轮播 (切换页面)
- 左右切换按钮版本
- 超高性能版本 (去掉 re-render)

你想要哪种？