

Flutter 扩展NestedScrollView (一)Pinned头引起的bug解决



法的空间 LV.5

2018年11月14日 22:27 · 阅读 20308

[Extended NestedScrollView](#) 相关文章

- [Flutter 扩展NestedScrollView \(一\)Pinned头引起的bug解决](#)
- [Flutter 扩展NestedScrollView \(二\)列表滚动同步解决](#)
- [Flutter 扩展NestedScrollView \(三\)下拉刷新的解决](#)

为什么想要自己来定义NestedScrollView呢？

FlutterCandies QQ群:181398081

要从我提交的2个issue开始讲：

1. [当中的Pinned为true的Sliver组件对body里面滚动组件的影响](#)
2. [当在里面放上tabview, 并且tab是缓存状态的时候, 会出现滚动会互相影响的问题](#)

没有任何进展, 用一个表情表达Flutter小组的意思



不过还好, 有源码, 还好我喜欢看源码。。这一篇的篇幅估计很多, 请先买好瓜子汽水前排坐好, 开车了。。

pub v6.0.0

NestedScrollView 是一个复杂的组件, 它跟Sliver 系列是一伙的, 最下层是个CustomScrollView.

Sliver系列的东东很多, 我们下面来一一介绍一下。

CustomScrollView

是Sliver组件的老祖宗, 全部的Sliver都放在这个里面。

[SliverList](#), which is a sliver that displays linear list of children.

[SliverFixedExtentList](#), which is a more efficient sliver that displays linear list of children that have the same extent along the scroll axis. 比SliverList多一个就是相同的行高。这样性能会更好

[SliverPrototypeExtentList](#) SliverPrototypeExtentList arranges its children in a line along the main axis starting at offset zero and without gaps. Each child is constrained to the same extent as the prototypeItem along the main axis and the SliverConstraints.crossAxisExtent along the cross axis.

[SliverGrid](#), which is a sliver that displays a 2D array of children. 可以设置每行的个数的Grid

[SliverPadding](#), which is a sliver that adds blank space around another sliver.

[SliverPersistentHeader](#) A sliver whose size varies when the sliver is scrolled to the leading edge of the viewport. This is the layout primitive that SliverAppBar uses for its shrinking/growing effect.

非常好用的组件, SliverAppBar就是用这个实现的。这个组件的特点是可以创建出随着滑动变化的可以Pinned的元素, 大家经常用的什么吸顶组件可以用这个很方便的构建, 后面我会使用这个写一个自定义效果的SliverAppBar。

[SliverAppBar](#), which is a sliver that displays a header that can expand and float as the scroll view scrolls.

[SliverToBoxAdapter](#) 当你想把一个非Sliver的Widget放在CustomScrollView里面的时候, 你需要用这个包裹一下。

SliverSafeArea A sliver that insets another sliver by sufficient padding to avoid intrusions by the operating system. For example, this will indent the sliver by enough to avoid the status bar at the top of the screen.为了防止各种边界的越界, 比如说越过顶部的状态栏

SliverFillRemaining sizes its child to fill the viewport in the cross axis and to fill the remaining space in the viewport in the main axis. 使用这个它会填充完剩余viewport里面的全部空间

SliverOverlapAbsorber,SliverOverlapAbsorberHandle 这个上面2个是官方专门为了解决我们今天主角**NestedScrollView**中Pinned 组件对Body 里面Scroll 状态影响的, 但官方做的不够完美。

看源码是一件好玩的事情, 大家跟我一起来吧。

`flutter\packages\flutter\lib\src\widgets\nested_scroll_view.dart`

首先我们看看第一个问题, 从官方文档中的Sample可以看到**NestedScrollView**

less 复制代码

```
DefaultTabController(  
  length: _tabs.length, // This is the number of tabs.  
  child: NestedScrollView(  
    headerSliverBuilder: (BuildContext context, bool innerBoxIsScrolled) {  
      // These are the slivers that show up in the "outer" scroll view.  
      return <Widget>[  
        SliverOverlapAbsorber(  
          // This widget takes the overlapping behavior of the SliverAppBar,  
          // and redirects it to the SliverOverlapInjector below. If it is  
          // missing, then it is possible for the nested "inner" scroll view  
          // below to end up under the SliverAppBar even when the inner  
          // scroll view thinks it has not been scrolled.  
          // This is not necessary if the "headerSliverBuilder" only builds  
          // widgets that do not overlap the next sliver.  
          handle: NestedScrollView.sliverOverlapAbsorberHandleFor(context),  
          child: SliverAppBar(  
            title: const Text('Books'), // This is the title in the app bar.  
            pinned: true,  
            expandedHeight: 150.0,  
            // The "forceElevated" property causes the SliverAppBar to show  
            // a shadow. The "innerBoxIsScrolled" parameter is true when the  
            // inner scroll view is scrolled beyond its "zero" point, i.e.  
            // when it appears to be scrolled below the SliverAppBar.  
            // Without this, there are cases where the shadow would appear  
            // or not appear inappropriately, because the SliverAppBar is  
            // not actually aware of the precise position of the inner  
            // scroll views.  
            forceElevated: innerBoxIsScrolled,  
            bottom: TabBar(  
              // These are the widgets to put in each tab in the tab bar.  
              tabs: _tabs.map((String name) => Tab(text: name)).toList(),  
            ),  
          ),  
        ),  
      ],  
    ),  
  ],  
),  
body: TabBarView(  
  // These are the contents of the tab views, below the tabs.  
  children: _tabs.map((String name) {  
    return SafeArea(  
      top: false,  
      bottom: false,  
      child: Builder(  
        // This Builder is needed to provide a BuildContext that is "inside"  
        // the NestedScrollView, so that sliverOverlapAbsorberHandleFor() can  
        // find the NestedScrollView.  
        builder: (BuildContext context) {  
          return CustomScrollView(  
            // The "controller" and "primary" members should be left  
            // unset, so that the NestedScrollView can control this  
            // inner scroll view.  
            // If the "controller" property is set, then this scroll  
            // view will not be associated with the NestedScrollView.  
            // The PageStorageKey should be unique to this ScrollView;  
            // it allows the list to remember its scroll position when  
            // the tab view is not on the screen.  
            key: PageStorageKey<String>(name),  
            slivers: <Widget>[  
              SliverOverlapInjector(  
                // This is the flip side of the SliverOverlapAbsorber above.  
                handle: NestedScrollView.sliverOverlapAbsorberHandleFor(context),  
              ),  
              SliverPadding(  
                padding: const EdgeInsets.all(8.0),  
                // In this example, the inner scroll view has  
                // fixed-height list items, hence the use of  
                // SliverFixedExtentList. However, one could use any  
                // sliver widget here, e.g. SliverList or SliverGrid.  
                sliver: SliverFixedExtentList(  

```

```

// The items in this example are fixed to 48 pixels
// high. This matches the Material Design spec for
// ListTile widgets.
itemExtent: 48.0,
delegate: SliverChildBuilderDelegate(
  (BuildContext context, int index) {
    // This builder is called for each child.
    // In this example, we just number each List item.
    return ListTile(
      title: Text('Item $index'),
    );
  },
  // The childCount of the SliverChildBuilderDelegate
  // specifies how many children this inner List
  // has. In this example, each tab has a List of
  // exactly 30 items, but this is arbitrary.
  childCount: 30,
),
),
),
],
);
},
),
);
}).toList(),
),
),
)

```

可以看到官方用一个SliverOverlapAbsorber包裹了SliverAppBar，在下面body里面，每一个list的上面都加了一个SliverOverlapInjector。实际效果就是SliverOverlapInjector的高度就等于SliverAppBar的Pinned的高度。如果不加入这些代码，当body里面的list滚动到SliverAppBar下方的时候。。依然可以继续向上滚动，也就是说body的滚动最上面点为0，而不是SliverAppBar的Pinned 高度。

为什么会出现这种情况呢？这要从Sliver的老祖宗CustomScrollView说起来。可能很多人发现,这些Sliver widgets(可以滚动的那种)没有ScrollController这个东西(CustomScrollView和NestedScrollView除外)。其实当你把Sliver Widgets(可以滚动的那种)放到CustomScrollView里面的时候将由CustomScrollView来统一处理各种Sliver Widgets(可以滚动的那种)，每个Sliver Widgets(可以滚动的那种)都会attach 各自的ScrollPosition。比如说第一个列表滚动到头了，第2个列表就会开始处理对应的ScrollPosition，将出现在viewport里面的元素render出来。

在我们的主角NestedScrollView当中，有2个ScrollController。

```

class _NestedScrollController extends ScrollController {
  _NestedScrollController(
    this.coordinator, {
    double initialScrollOffset = 0.0,
    String debugLabel,
  }) : super(

```

一个是inner，一个outer。outer是负责headerSliverBuilder里面的滚动widgets inner是负责body里面的滚动widgets 当outer滚动到底了之后，就会看看inner里面是否有能滚动的东东，开始滚动。

为了解决1问题，我们这里需要来处理outer这个ScrollController里面控制的_NestedScrollPosition，问题1在于，当header里面有多pinned的widget的时候，我们outer能滚动的extent。应该要去减掉这个pinned的总的高度。这样当滚动到pinned的组件下方的时候。我们就会开始滚动inner。

在_NestedScrollPosition 里面

```

// The _NestedScrollPosition is used by both the inner and outer viewports of a
// NestedScrollView. It tracks the offset to use for those viewports, and knows
// about the _NestedScrollCoordinator, so that when activities are triggered on
// this class, they can defer, or be influenced by, the coordinator.
class _NestedScrollPosition extends ScrollPosition
  implements ScrollActivityDelegate {
  _NestedScrollPosition({
    @required ScrollPhysics physics,
    @required ScrollContext context,
    double initialPixels = 0.0,
    ScrollPosition oldPosition,
    String debugLabel,
    @required this.coordinator,
  }) : super(

```

我override了applyContentDimensions方法

```

@override
bool applyContentDimensions(double minScrollExtent, double maxScrollExtent) {

```

```

if (debugLabel == 'outer' &&
    coordinator.pinnedHeaderSliverHeightBuilder != null) {
    maxScrollExtent =
        maxScrollExtent - coordinator.pinnedHeaderSliverHeightBuilder();
    maxScrollExtent = math.max(0.0, maxScrollExtent);
}
return super.applyContentDimensions(minScrollExtent, maxScrollExtent);
}

```

pinnedHeaderSliverHeightBuilder是我从最外层传递进来的用于获取当时Pinned 为true的全部Sliver header的高度。。在这里把outer最大的滚动extent减去了Pinned 的总的高度，这样我们就完美解决了[问题.1](#)

Sample code

在我的demo里面。pinned 的高度 由 status bar + appbar + 1个或者2个tabbar 组成。这里为什么要用个function而不是直接传递个算好的高度呢？因为在我的case里面这个pinned的高度是会改变的。

```

var tabBarHeight = primaryTabBar.preferredSize.height;
var pinnedHeaderHeight =
    //statusBa height
    statusBarHeight +
    //pinned SliverAppBar height in header
    kToolbarHeight +
    //pinned tabbar height in header
    (primaryTC.index == 0 ? tabBarHeight * 2 : tabBarHeight);
return NestedScrollViewRefreshIndicator(
    onRefresh: onRefresh,
    child: extended.NestedScrollView(
        headerSliverBuilder: (c, f) {
            return _buildSliverHeader(primaryTabBar);
        },
        //
        pinnedHeaderSliverHeightBuilder: () {
            return pinnedHeaderHeight;
        },
    ),

```

kotlin 复制代码

最后放上 [Github extended_nested_scroll_view](#), 如果你有更好的方式解决这个问题或者有什么不明白的地方, 都请告诉我, 由衷感谢。

pub v6.0.0



@稀土掘金技术社区