

5 Widgets Flutter

Aluno: Danrlei Vieira
Professor: Everton Coimbra

Widgets escolhidos

- Transform
- DraggableScrollableSheet
- NestedScrollView
- NotificationListener
- RefreshIndicator

TransForm

TransForm

Ao contrário de RotatedBox , que aplica uma rotação antes do layout, este objeto aplica sua transformação logo antes da pintura, o que significa que a transformação não é levada em consideração ao calcular quanto espaço o child deste widget consome.

Propriedades Transform

alignment → **AlignmentGeometry**

O alinhamento da origem, em relação ao tamanho da caixa.

origin → **Offset**

A origem do sistema de coordenadas no qual aplicar a matriz.

transformHitTests → **bool**

Se deve aplicar a transformação ao executar testes de clique.

Métodos TransForm

createElement () → SingleChildRenderObjectElement

RenderObjectWidgets sempre aumenta para uma subclasse
RenderObjectElement .

debugFillProperties () → void

Adicione propriedades adicionais associadas ao nó.

DraggableScrollableSheet

DraggableScrollableSheet

DraggableScrollableSheet é um widget que contém um widget de rolagem, permitindo arrastá-lo e redimensioná-lo até que um determinado limite seja alcançado e, em seguida, habilitar a rolagem no conteúdo interno.

Propriedades DraggableScrollableSheet

initialChildSize → **double**

O valor fracionário inicial da altura do contêiner pai a ser usado ao exibir o widget.

maxChildSize → **double**

O valor fracionário máximo da altura do contêiner pai a ser usado ao exibir o widget.

minChildSize → **double**

O valor fracionário mínimo da altura do contêiner pai a ser usado ao exibir o widget.

Métodos

DraggableScrollableSheet

createState () → _DraggableScrollableSheetState

Cria o estado mutável para este widget em um determinado local na árvore.

createElement () → StatefulElement

Cria um StatefulElement para gerenciar a localização deste widget na árvore.

NestedScrollView

NestedScrollView

Uma visualização de rolagem dentro da qual podem ser aninhadas outras visualizações de rolagem, com suas posições de rolagem sendo intrinsecamente vinculadas.

O caso de uso mais comum para este widget é uma visualização rolável com um SliverAppBar flexível contendo um TabBar no cabeçalho

Propriedades NestedScrollView

body → **Widget**

O widget a ser mostrado dentro de NestedScrollView .

controller → **ScrollController**

Um objeto que pode ser usado para controlar a posição para a qual a exibição de rolagem externa é rolada.

scrollDirection → **Axis**

O eixo ao longo do qual a visualização de rolagem rola.

Métodos NestedScrollView

toStringDeep () → String

Retorna uma representação de string deste nó e seus descendentes

debugFillProperties () → void

Adicione propriedades adicionais associadas ao nó.

NotificationListener

NotificationListener

O widget NotificationListener escutará as notificações que estão borbulhando na árvore. As notificações irão acionar o retorno de chamada onNotification apenas se seu runtimeType for um subtipo de T.

Para despachar notificações, use o método Notification.dispatch .

Propriedades NotificationListener

runtimeType → **Type**

Uma representação do tipo de tempo de execução do objeto.

onNotification → **NotificationListenerCallback < T >**

Chamado quando uma notificação do tipo apropriado chega a este local na árvore.

Métodos NotificationListener

build(BuildContext context) → Widget

Descreve a parte da interface do usuário representada por este widget.

toStringShallow () → String

Retorna uma descrição detalhada de uma linha do objeto.

RefreshIndicator

RefreshIndicator

Um widget que suporta o idioma "deslizar para atualizar".

Quando a rolagem termina, se o indicador foi arrastado longe o suficiente para se tornar completamente opaco, o retorno de chamada `onRefresh` é chamado. Espera-se que o retorno de chamada atualize o conteúdo do rolável e, em seguida, conclua o `Future` que ele retorna. O indicador de atualização desaparece após a conclusão do `Future` do `callback`.

Propriedades RefreshIndicator

backgroundColor → **Color**

A cor de fundo do indicador de progresso. O ThemeData.canvasColor do tema atual por padrão.

edgeOffset → **double**

O deslocamento onde RefreshProgressIndicator começa a aparecer no início do arrasto.

triggerMode → **RefreshIndicatorTriggerMode**

Define como este RefreshIndicator pode ser acionado quando os usuários ultrapassam a rolagem.

Métodos RefreshIndicator

toStringShort () → String

Uma breve descrição textual deste widget.

createState () → RefreshIndicatorState

Cria o estado mutável para este widget em um determinado local na árvore.