そのため、前回のレッスンでは、mockupのように見えるようにアプリを設計することで、すでにかなり進んでいます。

そして今、私たちがしなければならないことは、タスク項目をリストビューに追加することです。

Android Studioに進み、構造を確認できるようにすべてを展開しましょう。

現在、フローティングアクションボタンと列を備えたスキャフォールドがあります。

その列には、アプリの上部にあるコンテナと、アプリの下部を占める別のコンテナが含まれています。

リストビューを追加したいのは、この一番下のコンテナです。

これを選択して、ウィジェットツリーの適切な場所に到達できるようにします。これが課題です。

前回のモジュールでリストビューを既に作成し、それらがどのように見えるか、必要なものを知っていること、そして作成したいもののモックアップを用意しました。

Flutterのドキュメントをいくつか試してみるか、オンラインでStack Overflowを検索して、これまでに学んだことを使用して独自のリストビューを作成して、モックアップの外観を再現できるかどうかを確認してください。

そして、一度試してみたら、ここに戻って解決策を示します。

準備ができたら、ビデオを一時停止して、それを試してみてください。

大丈夫。

それでは、ソリューションを見ていきましょう。

まず、リストビューを作成する場所がコンテナであることにお気づきでしょう。

そして、この特定のコンテナにはまだ子がありません。

追加するのは、子プロパティです。

これで、この場合の子はもちろん、リストビューになり、To Doリストのセットアップとレイアウトに役立ちます。

リストビュー内には、変更できるプロパティがたくさんありますが、最も重要なのはその子です。

そのリストの中に何が入りますか？

この場合、実際には別の便利なウィジェットであるListTileを使用します。

これにより、タイトルなどのプロパティを持つ再利用可能なタイルを作成するだけで、リストの主要部分に何が入るかが非常に簡単になります。

トレイリング、最後に行くもの、または先頭に行くもの、フロントに行くものなど、onTapなどのプロパティと、コアウィジェットに通常期待されるすべてのものがあります。

そこで、タイトルを追加して、テキストウィジェットが必要な場合、このテキストウィジェットは「これはタスクです」と言います。

それでは、保存してみましょう。

リスト項目がここに表示されます。

では、小さなチェックボックスはどうでしょうか。

さて、これは最後の部分になりますので、このリストタイルの最後です。そして簡単に、チェックボックスウィジェットがあり、値を与えるだけです。

それで、あなたはそれをチェックしたりチェックを外したりしますか？

さて、チェックなしで始めましょう。

そのため、小さなチェックボックスがあります。そして、タイルを複数減らしたい場合は、先に進んで別のタイルをコピーしてみましょう。

リストには2つのタスクがあります。

しかし、タイルを入れると、少し奇妙に見えるようになります。

モックアップを見ると、すべてがきちんと揃っていることがわかります。これは本当にデザイナーの秘密の1つです。すべてが単一のポイントまたは単一の線に整列している限り、多くの労力をかけずに何でもかなり見栄えよくすることができます。

リストビューでも同様です。したがって、コンテナ内にいるため、もちろんパディングプロパティがあります。

そのため、左右にパディングを追加するだけです。

そのため、エッジインセットは対称になります。

したがって、水平を指定するだけで済みます。つまり、左右を指定します。

そして、前のものと一致するために、おそらく20ピクセルになるでしょう。

それで、今ではすっきりしていて、もっとデザインされていますよね？

リストビューを作成したので、タスク画面で少し面倒になりました。そして、すべてが同じ場所に座っていることは望ましくありません。

それでは、いくつかのウィジェットを抽出してみましょう。さて、最初に削除したいものは、おそらくリストビューになります。

Flutterアウトラインのリストビューをクリックして、ウィジェットを抽出してみましょう。

そこで、これをTasksListと呼び、リファクタリングを実行します。そして、ここで「タスクリスト」内に、リストタイルがまだありますか？

そこで、これらのいずれかを使用して、先に進んでそれを抽出します。

したがって、これはTaskTileになります。

そして今、私はそれをすべて屈折させて、物事を片付ける準備ができています。

いつものように、私は新しいキーワードを削除します。また、自動的に作成または抽出されたウィジェットのすべてのコンストラクターも削除します。本当に必要ないからです。

実際に構築するものは何も渡していません。

そして、ここに別の新しいキーワードがあり、コードを少しクリーンアップします。

タスクタイルを個別のウィジェットに抽出したので、ウィジェットのすべてのコードを書き出す必要はもうありません。

必要なときに別のタスクタイルを作成するだけです。

ここで別のタスクタイルを作成し、3つ目のタスクタイルをここに追加します。

これで、すべてそのチェックボックスとタイトルを使用した3つのタスクができました。

これでチェックポイントに達したので、保存をクリックして、これらの小さなアイテム、小さな抽出されたウィジェットのそれぞれを保持するための新しいフォルダーを作成しましょう。そして、このフォルダーをウィジェットフォルダーと呼びます。これらは、私たちが作成したすべてのカスタムウィジェットの一種だからです。

そして、widgetsフォルダー内でいくつかのDartファイルを作成しましょう。

これらの1つはtasks\_listになり、もう1つはtask\_titleになります。

これらの両方で、マテリアルパッケージをインポートする必要があります。

ですから、今すぐに行ってください。そして、関連するステートレスウィジェットを単純にコピーします。

タスクリストはここに入り、タスクタイルはここに行きます。そのため、Dart分析にいくつかのエラーがあります。

メソッドのタスクリストはこのクラスに対して定義されていないため、インポートしていないため、このタスクリストについては知りません。

だから、いたずら。

では、widgets / tasks\_list.dartを追加して、今何を話しているのかを正確に把握しましょう。

しかし、ここでは、タスクタイルが何であるかが分からないため、まだ3つのエラーがあります。

それでは、ウィジェットフォルダーとtask\_tile.dartをインポートしてみましょう。

これで、すべてのエラーをほぼ取り除くことができたので、赤で表示されたエラーと、現在の警告は単にいくつかの警告であるため、黄色で表示されたエラーのみです。そして、アプリ内では、上部領域、タイトルとアイコンの列、サブタイトルの列を取得するためのインターフェイス全体を作成し、スクロール可能な小さなリストをここに作成しました。 「チェックボックスとリスト項目があり、アクションボタンがあります。

これを試してみてもうまくいかなくても心配しないでください。チェックボックスはまだ機能しません。動作方法を指示するコードをまだ追加していないため、まもなく実行します。しかし、今ここでDart分析を見てみると、整理する必要があることがいくつかあります。まず、onPressedパラメーターが必要です。

そのため、ウィジェットに必要なプロパティまたはパラメーターを指定していない場合、ウィジェットはこの明るい黄色で強調表示されます。

カーソルを合わせると、何が欠けているかがわかります。この場合、フローティングアクションボタンにはこの素晴らしいレイアウトがすべて含まれていますが、実際には、押されたときに何をすべきかまだわかりません。

それでは、onPressedコールバックをここに追加して、何が起こるべきかをここに追加しましょう。

ここでこのプラスボタンをクリックすると、実際に新しいタスクを入力できるようにスライドさせます。モックアップを振り返って2ページ目に進むと、これがまさに私たちがやろうとしていることです。

スライドする新しいコンテナを取得し、素敵な丸いエッジも取得します。ここで新しいタスクを入力して、To Doリストに追加することができます。

それが次のレッスンで取り組むことです。

それで、それ以上のことについて、私はそこを見るでしょう。