大丈夫。

そのため、前回のレッスンでは、タスクプロバイダーからタスクを取得し、タスク画面とタスクリストの両方でそれを読み取る方法について説明しました。

このレッスンでは、タスクプロバイダーに新しいタスクを追加する方法と、タスクの追加画面からそのタスクを取得する方法について説明します。

したがって、ユーザーがここに新しいタスクを入力し、追加ボタンを押すと、そのデータはツリーのこの部分であるこのウィジェット内の右側に届きます。そのデータはタスクオブジェクトに移動して更新する必要があるため、リストに新しいタスクを追加し、タスク画面内の両方でそのプロバイダーの変更をリッスンして、タスクの数と内部の両方を更新できるようにしますタスクリストを使用して、新しいアイテムをリストに実際に追加し、リストを再構築できます。

あなたがこれに取り組むことができると思うなら、今すぐビデオを一時停止し、それを試してみて、この追加タスク画面を機能させ、リスト内でそれを見るだけでなく、それを見て、ここで番号を更新するこの機能を完了しようとします

そして、一度行って戻ってきたら、私のソリューションを紹介します。私たちはそれで大丈夫です。

したがって、タスクを追加するために、もちろんタスク追加画面内でそれを行います。現時点での動作方法は、ユーザーが新しいタスクに対応する新しいテキストを入力するこのテキストフィールドがあり、それをnewTaskTitleというローカルプロパティ内に保存することです。

そして、ユーザーが最終的にテキストフィールドの変更または入力を完了したら、[追加]ボタンを押します。その瞬間に、その値を取得し、設定状態を呼び出すタスク画面内に戻すコールバックをトリガーします。新しいタスクをタスクのリストに追加します。

しかし、現在、タスクのリストは存在せず、タスク画面もステートフルではないため、セット状態も使用できません。

プロバイダーの方法でこれを行う方法を見てみましょう。

今、あなたはそれができると思うかもしれない1つの方法はこのようなものです。タスクの追加画面内でコールバックを削除して、モーダルボトムシートとして作成される単純なステートレスウィジェットになる場合、タスクの追加画面に移動して、次の呼び出しも削除します。そのメソッドをここで実行し、そのコールバックを渡すこれらのプロパティをすべて削除します。

そのため、onPressedが呼び出されたときに新しいタスクを作成し、そのタスクに新しいタスクタイトルと呼ばれるローカルプロパティから作成された名前を付けます。

それをそこに入れましょう。

そしてもちろん、model / task.dartをインポートしていないため、タスクについては知りません。

それでは先に進んでそれをしましょう。

そこで、ユーザーが入力した内容に基づいて新しいタスクを作成しました。provider.dartをタップするだけで、models / task\_data.dartをタップして、onPressedプロバイダー内に書き込むだけでよいと思うかもしれません。データ型はTaskDataデータ型になります。

これで、タスクデータオブジェクトにアクセスできるようになり、ここで.tasks.addと言うだけで、ここに新しいタスクを追加できます。そして最後に、ここでnavigator.popを呼び出して、この現在の画面をポップします。

今、この種の種類は理論上、特にこの部分で理にかなっていますが、実際には、先に進んでこれを実行すると、実際には機能しないことがわかります。

それでは、先に進んでお見せします。

新しいタスクを入力しましたが、

[追加]を押すと、その画面は正しく閉じられますが、何も追加されません。

ここで何が起こっているのでしょうか？

前述の例を見ていくと、単にプロパティデータをタップして、新しい値を設定して新しい値を与えるだけで文字列を変更することはできないことを思い出してください。

通知リスナーを呼び出す必要があるため、メソッドを介して行う必要があります。そして、それは、特定のプロパティの状態をリッスンするすべての場所を更新し、再描画および再構築できるようにする変更通知クラスに由来するメソッドです。同様に、単にこの新しいタスクをタスクリストに追加するだけで、その通知リスナーを呼び出さなければ、何も通知されず、何も起こりません。

代わりに、メソッドを作成する必要があり、addTaskを呼び出してnewTaskTitleの文字列を渡します。そして、このメソッド内でタスクをタップし、.addを呼び出して新しいタスクを追加します。したがって、ここで順番に実行するか、ここで作成することができます。

そのため、最終タスクは、新しいタスクタイトルの名前を持つ新しいタスクと等しくなり、それを渡してこの新しいタスクをタスクリストに追加します。

しかし最も重要なことは、完了したら通知リスナを呼び出して、リスニングしているタスクオブジェクトの最新の状態ですべてのリスナを更新することです。

そこで、タスクの追加画面で、これをすべて行う代わりに、最初にこれを削除してから、provider.ofタスクデータを呼び出し、今作成したメソッドを追加して、Taskを追加し、その新しいタスクを渡します。ここからテキストフィールドを介して入力されるタイトル。だから今、私が先に進み、保存してもう一度チェックアウトしようとすると、「ジムに行く」という新しいタスクを追加し、「追加」をクリックすると、リストに追加されていることがわかりますここで重要なのは、リスナメソッドが呼び出されることです。これにより、ここだけでなくリスト内のタスクをリッスンしているすべての人が、それに応じて更新されます。単にこのようなタスクをそのままにしておくと、プロバイダーについて学んだこの知識のすべてを忘れてしまうことがあるので、実際には非常に危険です。 .tasks.addと言って新しいタスクを追加するだけです。

ですから、将来この間違いを犯さないようにするために、あなたは自分の未来に親切にならなければならないことを忘れないでください。

プログラマーとしてだけでなく、人生においても、あなたは自分の将来のバージョンのために人生を楽にしなければなりません。

それでは、将来この間違いを防ぐ方法を考えてみましょう。積極的に覚えておく必要はありません。

このリストをタップして、.addまたは.removeを使用することはできません。

実際に、リスナーに通知するメソッドを介して行う必要があります。

したがって、このタスクをどこにでも公開してアクセスできるようにする代わりに、その前にアンダースコアを追加してプライベートプロパティにします。

そのため、ここでは中括弧の2つの壁内でのみアクセスでき、メソッド内およびここのゲッター内でそれを利用できます。

そのため、タスクカウントを返すときとタスクを追加するときの両方で、このプライベートバージョンのタスクを利用しています。

そして今、私が外部からタスクを利用しようとした場合、私はもはやそれを行うことができません。

外部からはアクセスできなくなります。

そのため、新しいタスクのタイトルを新しいタスクの文字列として追加するために、タスクの追加メソッドを使用する必要があります。

これにより、エラーが最小限に抑えられます。しかし、今、ある種の問題がありますか？私はそのタスクを他の場所で使用しており、それらの場所は、たとえば、そのタスクリストから読み込もうとしているときにここにあるからです。そのタスクリストがプライベートになった場合、どうやってここで読むのですか？

そして、それが私がこれらすべてのエラーを受け取っている理由です。

それでは、このタスクのゲッターを作成しましょう。

そこで、ここでタスクタイプの新しいリストを作成し、それをパブリックにします。

したがって、アンダースコアなしでタスクと呼ばれます。

そして、ここでゲッターを作成します。プロパティ名の前にgetキーワードを追加し、中括弧のセットを開き、次に計算する値を返します。タスクのプライベートバージョンになります。

ここで問題がなく、タスクにアクセスできるため、これでうまくいくと思うかもしれません。しかし、今では同じ場所に戻ってきましたよね？

タスクを再度タップして、

私はそれに値を追加しようとすることができます

そして今、私は以前と同じ問題に直面しています。

それでは、どうしてこれを行わないことを確認し、代わりにaddメソッドを使用してこれを安全性チェックすることを確認できますか？計算されたプロパティで沼地の標準リストを使用する代わりに、ゲッターを持つタスクに変更不可のリストビューと呼ばれるものを使用します。これはDartコレクションライブラリからのものなので、これをインポートします。そして、ここでそのタイプをUnmodifiableListViewに変更します。

さて、これはあなたがそうするのであれば一種の誤称です。

ListViewがあるウィジェットのように、リストビューのことを考えさせるため、このクラスの名前の付け方が好きではありません。しかし、実際にはリストビューウィジェットではありません。

実際には、リストデータ型の別のバージョンですが、別のリストのビューのようなものです。

そのため、彼らはそれを変更不可能なリストビューと呼んでいます。

リストのバージョンを見ているようなものですが、ウィンドウペインからリストを見ているようなものです。

あなたはそれに触れることはできません、それを変更することはできません。そして実際、あなたが言うことを試みたなら、私は知りません、追加タスクの中で、例えば私たちのタスクをタップしてみてください

modifiableListViewで、.addを呼び出して、ランダムな名前の新しいタスクオブジェクトを追加しようとしました。

ここで私が望んでいたエラーはここでは表示されませんが、実際に実行するとエラーが表示されます。コマンドを押したままこ​​の追加関数をクリックして実装方法を確認すると、エラーをスローするだけです。

このメソッドは変更できないリストではサポートされていないため、変更できないリストに追加できないことを示しています。

それがポイントですよね？このリストを変更できるようにするつもりはありません。

そのため、変更不可能なリストビューを作成し、タスクの変更不可能なバージョンを返すことができます。

したがって、理論的には、外部からこのタスクに直接アクセスして変更しようとすると、エラーが発生します。

したがって、このタスクデータクラスを使用することで少し安全になり、addTaskメソッドを介して通知リスナーを呼び出すバージョンを常に使用するようになりました。これで、コードの処理が完了し、コードの安全性が向上しました。次は、アプリを保存してチェックアウトする準備が整いました。

それでは、「ジムに行く」という新しいアイテムを追加しましょう。

私は本当にジムに行く必要があります。

それをリストに追加して、コンシューマー経由でタスクをリッスンしている場所で更新を確認できるようにします。また、タスクデータのprovider.ofを使用してリッスンしている場所でも更新を確認できます。

それで、あなたはそれを機能させることができ、そのnotify andlistenersメソッドの呼び出しによってつまずいたのですか？

繰り返しますが、これがわかりにくい場合は、よりシンプルなバージョンに移行する価値があります。そのため、状態管理のサンプルアプリに戻り、すべてが非常にシンプルで、非常に簡単に理解できるようにします。自分でコードを書きます。

次のレッスンでは、設定状態やステートフルウィジェットを必要とせずに、タスクリストの項目をチェックオフする次の部分を見ていきます。

それで、それ以上については、次のレッスンで説明します。