大丈夫。

タスクを読み取ることができるようになったので、タスクを追加できます。

ここで、タスクをチェックオフする方法を理解します。

これで、実際のチェックオフはタスクタイトルで行われます。

そして、正しいタスクをチェックオフするために、それをタスクプロバイダー内に反映する必要があります。

そして、それをタスクリストに反映する必要があります。

そのため、タスクリストはその変更のリストになります。

この場合、タスクタイルを比較的単純に保ちたいため、それだけではありません。

したがって、ステートレスウィジェットとして保持したいものであり、データプロバイダーやプロバイダーパッケージのようなこれらすべてのことを知りたくないのです。

タスクリストから情報を取得し、必要に応じてタスクタイルをレンダリングする一種のダムウィジェットとして使用できるとしたらどうでしょうか。

それでは、タスクタイルを非常に単純なウィジェットに保ちながら、プロバイダーに関与せずにそのチェックオフ動作を取得するにはどうすればよいでしょうか。

したがって、まず、現在のタスクタイルがステートレスウィジェットであることに注意してください。

isCheckedブール値、タスクタイル文字列、チェックボックスが変更されたときに親に通知するコールバックが渡されます。そして、tasks\_list.dartである親の内部では、チェックボックスコールバック内に存在する設定状態を使用してタスクを更新していました。

この場合、ステートフルウィジェットの内部にいないため、状態を設定する必要はありません。

コンシューマーウィジェットへのアクセスが既にあるため、このタスクリストを状態を更新する場所に保つにはどうすればよいのでしょうか？タスクデータオブジェクトを既に変更でき、このウィジェットを複雑に巧妙に保ちながら、ここですべてを制限することができます指揮者のようなもので、タスクタイルをシンプルに保ち、指示されたものを実装するだけですか？

大企業と同様に、この場合はタスクリストであるミドルマネージャーがいる場合があります。それから、ミドルマネージャーが必要とせずに行うことを指示するだけの従業員がいる場合があります。 。

そのため、その仕事は非常にシンプルにしてください。

プロバイダーについて知っていることを使用して、どのようにそれを達成できますか？最初に行うことは、チェックボックスコールバック内でこのセット状態を削除することです。代わりに、提供されたタスクデータ内でそれを行います。

そのため、ここで新しいメソッドを作成します。そのupdateTaskを呼び出します。

そして、このメソッドでは、更新が必要なタスクを引き渡します。

そして、それを取得したら、task.toggle Doneを呼び出します。

そして、覚えているのはtask.dartであるため、特定のタスクの動作方法を定義したときに、そのtoggleDoneメソッドを使用してそれ自体の反対に切り替えることができるisDoneというプロパティを作成しました。そして、それがここで使用しているものです。

しかし、タスクプロパティisDoneの切り替えが完了したら、本当に重要なメソッドを呼び出します。

notifyListenersは、特定のタスクの状態をリッスンしているウィジェットを更新して、そのタスクの現在の完了状態またはチェックオフ状態を再構築して反映するようにします。

あとは、使用するだけです。先ほど言ったように、タスクタイルをシンプルに保ちます。

したがって、まったく同じ状態に保ちます。

まだコールバックがあります。この場合はタスクリストである上司に報告し、ユーザーがチェックボックスを変更するたびにトリガーするこのコールバックがあります。したがって、それをチェックオフまたはチェック解除するとき。そして、このコールバック内では、タスクデータをタップして、メソッド更新タスクを呼び出します。そして、私たちが渡そうとしているタスクはTaskData.tasksになり、現在のインデックスにあるタスクになります。

そのため、タスクタイルはリストビュービルダーから構築され、タスクタイルの各アイテムにインデックスが割り当てられます。

したがって、最上位にゼロがあり、次に1および2などになります。

したがって、更新する適切なタスクを取得するには、そのインデックスも使用する必要があります。

しかし、このtaskData.tasks [index]をどのように使用しているかに注目してください...現在3つの場所で。

したがって、タスクアイテムを作成するreturnステートメントのすぐ上に移動することで、これを実際に非常に簡単にリファクタリングできます。現在のインデックスでtaskData.tasksと等しくなるように設定します。

そして、そのタスクを保持したので、これらすべての場所でそれを使用できます。

したがって、task.name、task.isDoneになり、更新する必要があるものとしてタスクがここに入ります。

そのため、これによりコードが簡素化され、短く簡潔になります。

したがって、保存を押してアプリを再実行したら、Todoeyアプリに移動して、新しいタスクを追加し、それをチェックオフまたは他のタスクをチェックオフして、必要に応じてチェックを外すことができます。

それで、残っているのは、タスクを削除したい場合、タスクをチェックオフしたときでも完了したというだけなので、どうすればよいでしょうか？ そして、リストをチェックオフしてすべてが交差していることを確認する喜びをユーザーから奪いたくないので、チェックオフして削除したくありません。

実際、リストからチェックするために何かをするだけだと思うことがあります。

しかし、もし彼らがこれを行うためにリストをクリーンアップしたい場合はどうでしょうか？

さて、アイテムの1つをクリックして押したままにして、リストから削除することができたらどうでしょうか。

それが次のレッスンで取り組むことです。

それで、それ以上のために、あなたに会いましょう。