タスクをチェックオフし、タスクを追加し、タスクを読み取ることができるようになったので、実装する最後の作業に進みます。これは、ユーザーがアプリからタスクを削除できるようにする方法です。そして、ユーザーが特定のアイテムに触れて保持するだけで、それができるようになりたいと思っています。

そして、彼らがそれを十分に長い間（非常に長いタッチで）保持している限り、そのアイテムをリストから削除したい。そのため、タスクはウィジェットツリーの最上部で提供されていることがわかり、タスクタイル内でも相互作用が検出されていることがわかります。

タスクリストからそのタスクを削除できるようにするために、前のレッスンで確認した情報と使用した情報をどのように使用できますか？

この場合、その動作はタスクタイルのツリーの一番下で検出され、その情報を取得してタスクリストに渡して、タスクデータを更新できるようにする必要があります。ユーザーが選択したタスクを削除します。そして、その情報は、すべての関係者、つまりタスク画面とタスクリストのツリーをフィルター処理する必要があります。

これを試して、その機能を実装し、ロングトップが検出されたアイテムを削除することをお勧めします。

途中でスタックした場合は、Stack Overflowを使用するか、Googleで検索して、それらのハードルを克服してください。

そして最後には、あなたはそれにより強力な開発者になることを約束します。

また、完了したら、戻ってソリューションのバージョンを確認する場所を確認してください。今すぐビデオを一時停止し、試してみてください。

まず最初にやらなければならないのは、その長押しを実際に検出する方法をよく理解することです。

幸運なことに、リストタイルには、実際にはonLongPressというプロパティがあります。このプロパティは、長押しを検出するとコールバックをトリガーするので、数秒かかります。これで、チェックボックスコールバックで実行したものとまったく同じものを使用できます。

longPressCallbackと呼ばれる新しい最終関数を作成できます。また、コンストラクター内で初期化することもできます。そして、それをonLongPressに引き渡します。

したがって、リストタイルが長押しを検出すると、コールバックがトリガーされ、コールバックはタスクタイルの親から取得されます。これはもちろん、コンシューマが存在するタスクリストであり、提供されたタスクデータ。

したがって、チェックボックスコールバックに加えて、longPressCallbackも追加します。この場合、トリガーするのは、タスクデータ内のdeleteメソッドの呼び出しだけです。

そこで、deleteTaskと呼ばれる別のメソッドを作成し、削除する必要があるタスクを渡します。そして、タスクのリストをタップして、removeメソッドを呼び出します。そして、ユーザーが削除したいタスクになる特定のオブジェクトを削除します。ここ。

そしてもちろん、必要に応じて更新できるように、すべてのリスナーに通知する必要もあります。

ここで保存ボタンを押して、長押しコールバック内のタスクリストに移動すると、誰かがタスクタイルを長押ししたときに、実行したいのはタスクデータをタップして、deleteTaskメソッドを呼び出すことですタスクタイルの現在のタスクであるタスクを渡します。

それで、私がする必要があるのはそれだけです。

そして、保存ボタンを押してTodoeyアプリに移動し、そのボタンを押し続けると、my itemsが削除されることがわかります。そして、それが完了したかどうかは関係ありません。それがリストの上位にあるかどうかは関係ありません。すべて機能します。