Flutter Course

Value Notifier

Task 6

Fatma Abdelghany Mohammed

Value Notifier

ValueNotifier is a special type of class that extends a ChangeNotifier. ValueNotifier is native to Flutter itself, it provides individual element reactivity. Flutter also has a widget by default, using which we can listen to the ValueNotifier called ValueListenableBuilder. It can also be listened using AnimationBuilder since it is Listenable.

Flutter

Value Notifier

Think of ValueNotifier as a stream of data which holds to a value. We provide it a value and every listener receives a notification of the value change.
We can create a ValueNotifier of any type int, bool, list, or any custom data type. You can create a ValueNotifier object like this:

```
ValueNotifier<int> counter = ValueNotifier<int>(0);
```

ValueListenableBuilder

There are many types of builders in Flutter like StreamBuilder, AnimatedBuilder, FutureBuilder, etc. Their first name suggests what type of object they consume. The ValueListenableBuilder consumes the ValueNotifier object. The builder method is executed every time we receive a value update. The ValueListenableBuilder automatically removes the listener internally when we route to another page.

Flutter

ValueListenableBuilder

This is the constructor of ValueListenableBuilder. Here, valueListenable is the ValueNotifier to listen. The builder function receives 3 parameters (BuildContext context, dynamic value, Widget child). The value is the data received from the valueNotifier provided. It is possible to use the child parameter. We use it for optimization if the child is expensive to build and doesn't depend upon the value from the notifier.

```
const ValueListenableBuilder({
    @required this.valueListenable,
    @required this.builder,
    this.child,
  })
```

For Example

Counter App using Value Notifier :

The counter app using ValueNotifer and ValueListenableBuilder looks like this. Note how no setState is being used, and we rebuilt only the Text section in case of the value changes.

Flutter

Counter App

```
import 'package:flutter/material.dart';

void main() {
  runApp(const App());
}

class App extends StatelessWidget {
  const App({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return const MaterialApp(
      home: HomePage(),
    );
  }
}

class HomePage extends StatefulWidget {
  const HomePage({Key? key}) : super(key: key);

  @override
  State<HomePage> createState() => _HomePageState();
}
```

Flutter

Counter App

```
class _HomePageState extends State<HomePage> {
  final _counterNotifier = ValueNotifier<int>(0);

  @override
  Widget build(BuildContext context) {
    print('HOMEPAGE BUILT');
    return Scaffold(
      appBar: AppBar(title: const Text('Counter App')),
      body: Center(
        child: ValueListenableBuilder(
          valueListenable: _counterNotifier,
          builder: (context, value, _) {
            return Text('Count: $value');
          },
        ),
      ),
      floatingActionButton: FloatingActionButton(
        onPressed: () {
          _counterNotifier.value++;
        },
        child: const Icon(Icons.add),
      ),
    );
  }

  @override
  void dispose() {
    _counterNotifier.dispose();
    super.dispose();
  }
}
```

Flutter

ValueNotifier

Conclusion:

ValueNotifier is a versatile class in Flutter that can be used for a variety of purposes, including managing application state, form validation, and updating the UI. By using ValueNotifier, developers can create simple and efficient solutions for managing state in their applications.

For more Examples follow this link:

https://medium.com/@thekavak/flutter-valuenotifier-with-examples-66b3933d7036