**Widgetbook**

A powerful tool to test flutter widgets in different devices and themes. All of the widgets can be shared with designers, developers, product owners, clients.

# **Navigation Structure**

Widgetbook mainly contains different use cases of widgets and these can be categories in following ways.

* WidgetbookCategory
* WidgetbookFolder
* WidgetbookComponent
* WidgetbookUseCase

A WidgetbookCategory can contain multiple folders and components. Likewise, a component can contain multiple Use Cases. These are contained in the directory property of Widgetbook material app.

# **Knobs**

Knobs in Widgetbook helps to change the parameters of a use case. It contains a set of properties like label, description, initial value to display a use case differently. The different types of knobs are given below.

1. Boolean
2. Boolean?
3. String
4. String?
5. double
6. double?
7. num
8. num?
9. list

# **Addons**

Widgetbook material app also have addons which contains theme data addons for testing in different themes, device frame addon for testing in different resolutions, scalable text addon for testing in large and small text, and localization addon for testing in different languages.

# **App Builder**

Widgetbook material app have an app builder property which allows developers to customize the appearance of the use cases.

# **Implementation Ways**

It can be used in two different ways. The code generation method helps to save development time compared other widget isolation packages in flutter.

1. Manual Approach
2. Using Code Generator

# **Complete Example with Manual Approach**

import 'package:flutter/material.dart';

import 'package:widgetbook/widgetbook.dart';

void main() {

  runApp(const KnobsExample());

}

class KnobsExample extends StatelessWidget {

  const KnobsExample({Key? key}) : super(key: key);

  @override

  Widget build(BuildContext context) {

    return Widgetbook.material(

      addons: [

        MaterialThemeAddon(

          themes: [

            WidgetbookTheme(

              name: "light",

              data: ThemeData.light(),

            ),

            WidgetbookTheme(

              name: "dark",

              data: ThemeData.dark(),

            ),

          ],

        ),

        TextScaleAddon(

          scales: [

            1.0,

            1.25,

            1.5,

          ],

        ),

        DeviceFrameAddon(

          devices: [

            Devices.ios.iPhone13,

            Devices.android.samsungGalaxyNote20Ultra,

            Devices.ios.iPad12InchesGen4,

            Devices.windows.laptop,

          ],

          initialDevice: Devices.ios.iPhone13,

        ),

      ],

      directories: [

        WidgetbookCategory(

          name: 'Pages',

          children: [

            WidgetbookComponent(

              name: 'Onboarding',

              useCases: [

                WidgetbookUseCase(

                  name: 'Home Page',

                  builder: (context) => MyHomePage(

                    title: context.knobs

                        .string(label: 'Title', initialValue: 'Title'),

                    incrementBy: context.knobs.doubleOrNull

                            .slider(

                                label: 'Increment By',

                                min: 0,

                                initialValue: 5,

                                max: 10,

                                divisions: 10)

                            ?.toInt() ??

                        0,

                    countLabel: context.knobs.stringOrNull(

                      label: 'Count Label',

                      initialValue: 'Current Count',

                      description:

                          'This is the text that appears above the current count of increments',

                    ),

                    iconData: context.knobs.list(

                      label: 'Icon',

                      options: [

                        Icons.add,

                        Icons.crop\_square\_sharp,

                        Icons.circle

                      ],

                    ),

                    showToolTip: context.knobs.boolean(

                      label: 'Show Increment Tool Tip',

                      description:

                          'This is the tooltip that is displayed when hovering over the increment button',

                      initialValue: true,

                    ),

                  ),

                ),

              ],

            ),

            WidgetbookFolder(

              name: "Button Folder",

              children: [

                WidgetbookComponent(

                  name: 'button',

                  useCases: [

                    WidgetbookUseCase(

                      name: 'Elevated Button',

                      builder: (\_) => Center(

                        child: ElevatedButton(

                          onPressed: () {},

                          child: const Text("Button"),

                        ),

                      ),

                    ),

                    WidgetbookUseCase(

                      name: 'Text Button',

                      builder: (\_) => Center(

                        child: TextButton(

                          onPressed: () {},

                          child: const Text("Button"),

                        ),

                      ),

                    ),

                    WidgetbookUseCase(

                      name: 'Outlined Button',

                      builder: (\_) => Center(

                        child: OutlinedButton(

                          onPressed: () {},

                          child: const Text("Button"),

                        ),

                      ),

                    ),

                  ],

                ),

              ],

            ),

          ],

        ),

      ],

      appBuilder: (context, child) {

        return MaterialApp(

          debugShowCheckedModeBanner: false,

          home: child,

        );

      },

    );

  }

}

class MyHomePage extends StatefulWidget {

  const MyHomePage({

    Key? key,

    required this.title,

    this.incrementBy = 1,

    this.countLabel,

    this.iconData,

    this.showToolTip = true,

  }) : super(key: key);

  final String title;

  final int incrementBy;

  final String? countLabel;

  final IconData? iconData;

  final bool showToolTip;

  @override

  State<MyHomePage> createState() => \_MyHomePageState();

}

class \_MyHomePageState extends State<MyHomePage> {

  int \_counter = 0;

  void \_incrementCounter() {

    setState(() {

      \_counter += widget.incrementBy;

    });

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        title: Text(widget.title),

      ),

      body: Center(

        child: Column(

          mainAxisAlignment: MainAxisAlignment.center,

          children: <Widget>[

            Text(

              widget.countLabel ??

                  'You have pushed the button this many times:',

            ),

            Text(

              '$\_counter',

              style: Theme.of(context).textTheme.headlineMedium,

            ),

          ],

        ),

      ),

      floatingActionButton: FloatingActionButton(

        onPressed: \_incrementCounter,

        tooltip: widget.showToolTip ? 'Increment' : null,

        child: Icon(widget.iconData ?? Icons.add),

      ),

    );

  }

}