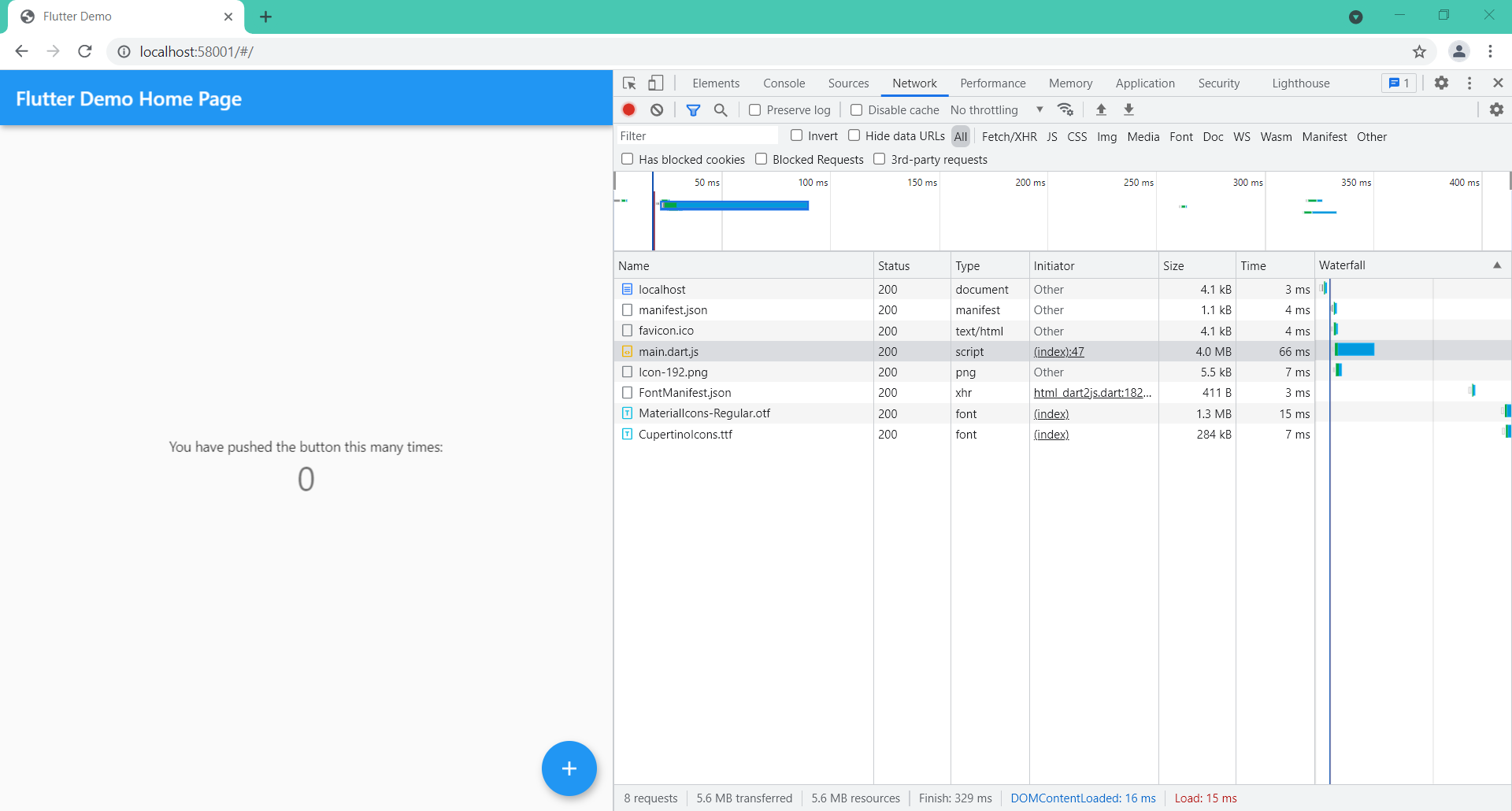


Canvaskit 125



36. Cooperation propriety with a native function. Cooperation with C++, etc.

Flutter mobile can use the [dart:ffi](https://api.dart.dev/dev/dart-ffi/dart-ffi-library.html) library to call native C, C++ APIs. But it didn’t work on flutter web. though we can use java script.

References:

Js: <https://pub.dev/packages/js>

C: <https://dart.dev/guides/libraries/c-interop>

35. limited package source. Some packages only work with android platform, not good with web for example: dio, path\_provider,..

34. Is state management performed?

we can use provider package to manage state. Following are the 3 basic components to management:

* Provider: provide an object that we can use anywhere in the widget tree covered by the Provider.
* ChangeNotifierProvider : listen for the change of the object it provides.
* Consumer: widgets wrapped by it will be rebuilt

import 'package:flutter/material.dart';

import 'package:provider/provider.dart';

void main() {

  runApp(const MyApp());

}

class MyApp extends StatelessWidget {

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

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      title: 'Flutter Demo',

      theme: ThemeData(

        primarySwatch: Colors.blue,

      ),

      home: MyHomePage(),

    );

  }

}

class MyHomePage extends StatelessWidget {

  @override

  Widget build(BuildContext context) {

    return ChangeNotifierProvider<MyModel>(

      create: (context) => MyModel(),

      child: MaterialApp(

        home: Scaffold(

          appBar: AppBar(title: Text('My App')),

          backgroundColor: Colors.grey,

          body: Row(

            mainAxisAlignment: MainAxisAlignment.center,

            crossAxisAlignment: CrossAxisAlignment.center,

            children: <Widget>[

              Container(

                padding: const EdgeInsets.all(20),

                color: Colors.green[200],

                child: Consumer<MyModel>(

                  builder: (context, mymodel, child) {

                    return RaisedButton(

                      child: Text('Do something'),

                      onPressed: () {

                        mymodel.doSomething();

                      },

                    );

                  },

                ),

              ),

              Container(

                padding: const EdgeInsets.all(35),

                color: Colors.blue[200],

                child: Consumer<MyModel>(

                  builder: (context, mymodel, child) {

                    return Text(mymodel.text);

                  },

                ),

              ),

            ],

          ),

        ),

      ),

    );

  }

}

class MyModel with ChangeNotifier {

  String text = "Hello";

  void doSomething() {

    text = "World";

    notifyListeners();

  }

}

32. Can file transmission and reception be performed?

Flutter web has support for rendering ui processing.

Download: We cam use Firebase, html, http. Dio does not support web platform.

<pre>

import 'dart:html' as html;

void downloadFile(String url){

html.AnchorElement anchorElement = new html.AnchorElement(href: url);

anchorElement.download = url;

anchorElement.click();

}

</pre>

Upload: Firebase, Dio,http.

Sample code: Upload file to Firebase.

<pre>

lass UplaodPage extends StatefulWidget {

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

  @override

  \_UplaodPageState createState() => \_UplaodPageState();

}

class \_UplaodPageState extends State<UplaodPage> {

  double progress = 0.0;

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      body: Container(

        child: Center(

          child: Column(

            mainAxisAlignment: MainAxisAlignment.center,

            children: [

              TextButton(

                onPressed: () async {

                  FilePickerResult? result =

                      await FilePicker.platform.pickFiles();

                  if (result != null) {

                    Uint8List? file = result.files.first.bytes;

                    String fileName = result.files.first.name;

                    UploadTask task = FirebaseStorage.instance

                        .ref()

                        .child("files/$fileName")

                        .putData(file!);

                    task.snapshotEvents.listen((event) {

                      setState(() {

                        progress = ((event.bytesTransferred.toDouble() /

                                    event.totalBytes.toDouble()) \*

                                100)

                            .roundToDouble();

                        print(progress);

                      });

                    });

                  }

                },

                child: Text("Upload"),

              ),

              SizedBox(

                height: 50.0,

              ),

              Container(

                height: 200.0,

                width: 200.0,

                child: LiquidCircularProgressIndicator(

                  value: progress / 100,

                  valueColor: AlwaysStoppedAnimation(Colors.pinkAccent),

                  backgroundColor: Colors.white,

                  direction: Axis.vertical,

                  center: Text(

                    "$progress%",

                    style: GoogleFonts.poppins(

                        color: Colors.black87, fontSize: 25.0),

                  ),

                ),

              )

            ],

          ),

        ),

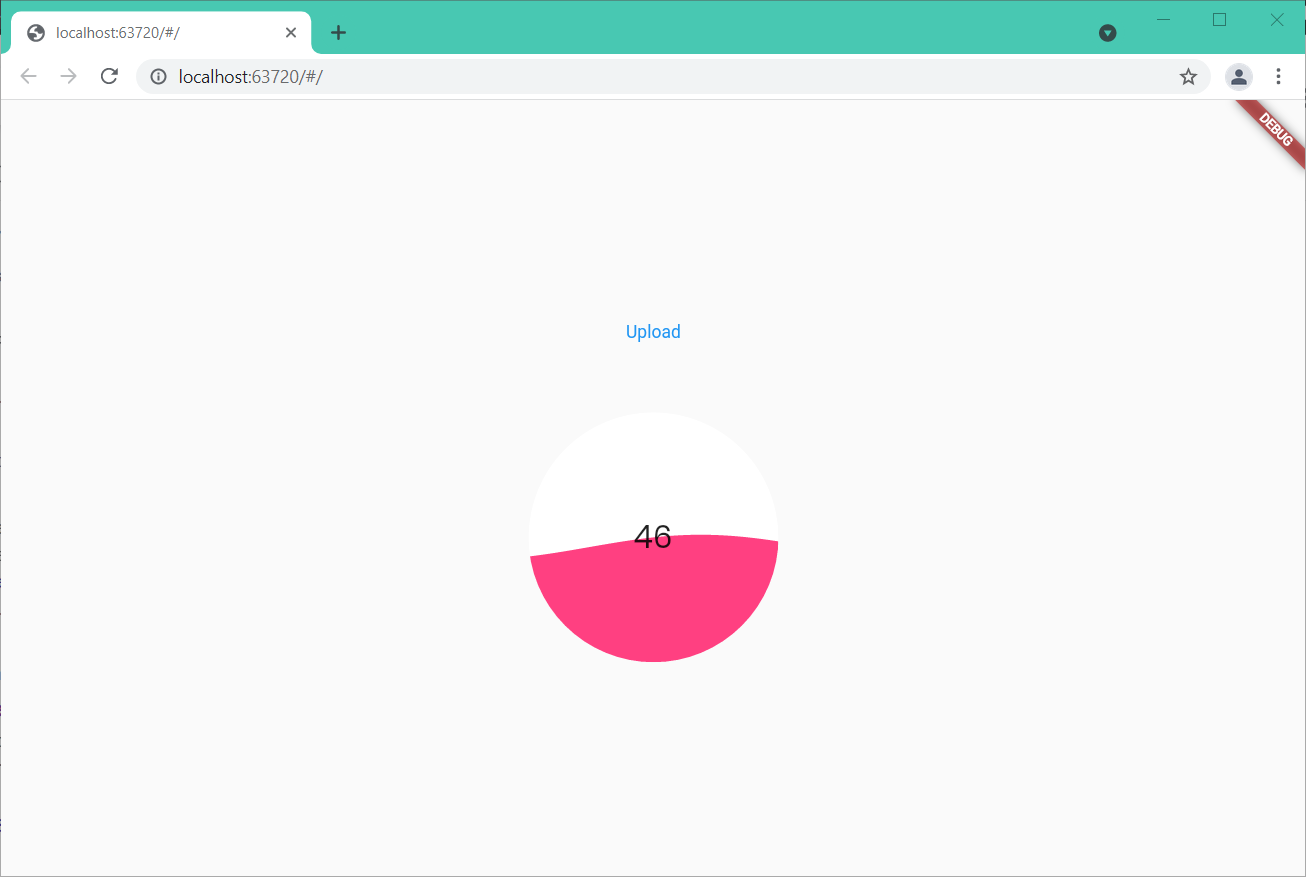
      ),

    );

  }

}

</pre>



Refrences: <https://pub.dev/packages/firebase_core>

30.

we can build crossplatform android,ios, web using material.dart, or we can optionally build function according to the platform of running device.

<pre>

import 'package:flutter/cupertino.dart';

import 'package:flutter/material.dart';

import 'widgets.dart';

class SongDetailTab extends StatelessWidget {

  const SongDetailTab({

    required this.id,

    required this.song,

    required this.color,

    Key? key,

  }) : super(key: key);

  final int id;

  final String song;

  final Color color;

  Widget \_buildBody() {

    return SafeArea(

      bottom: false,

      left: false,

      right: false,

      child: Text('My Widget: ...'),

    );

  }

  Widget \_buildAndroid(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text(song)),

      body: \_buildBody(),

    );

  }

  Widget \_buildWindow(BuildContext context) {

    return Scaffold(

      appBar: AppBar(title: Text(song)),

      body: \_buildBody(),

    );

  }

  Widget \_buildIos(BuildContext context) {

    return CupertinoPageScaffold(

      navigationBar: CupertinoNavigationBar(

        middle: Text(song),

        previousPageTitle: 'Songs',

      ),

      child: \_buildBody(),

    );

  }

  @override

  Widget build(context) {

    return PlatformWidget(

      androidBuilder: \_buildAndroid,

      iosBuilder: \_buildIos,

      windowBuilder: \_buildWindow,

    );

  }

}

</pre>

<pre>

class PlatformWidget extends StatelessWidget {

  const PlatformWidget({

    Key? key,

    required this.androidBuilder,

    required this.iosBuilder,

    required this.windowBuilder,

  }) : super(key: key);

  final WidgetBuilder androidBuilder;

  final WidgetBuilder iosBuilder;

  final WidgetBuilder windowBuilder;

  @override

  Widget build(context) {

    switch (defaultTargetPlatform) {

      case TargetPlatform.android:

        return androidBuilder(context);

      case TargetPlatform.iOS:

        return iosBuilder(context);

      default:

        return windowBuilder(context);

    }

  }

}

</pre>