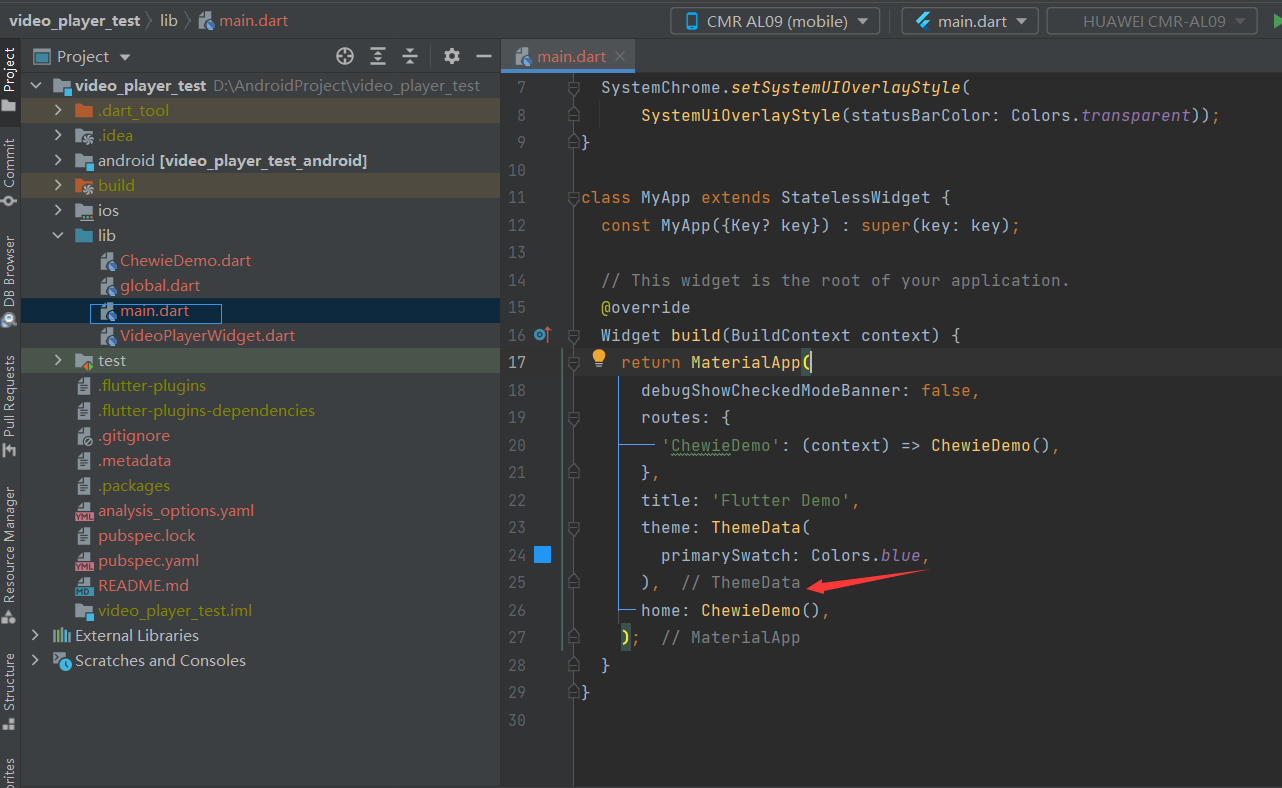
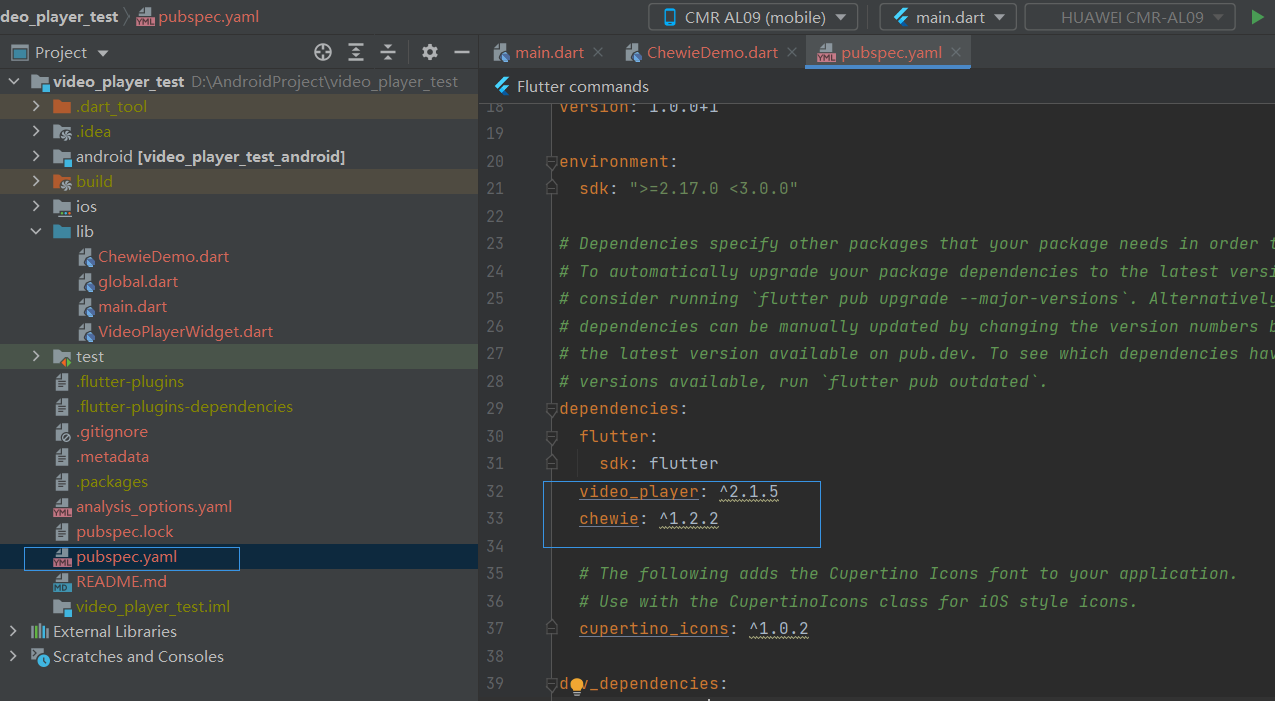
用了两个库

Main

import 'package:flutter/material.dart';  
import 'package:flutter/services.dart';  
import 'package:video\_player\_test/ChewieDemo.dart';  
  
void main() {  
 runApp(const MyApp());  
 SystemChrome.*setSystemUIOverlayStyle*(  
 SystemUiOverlayStyle(statusBarColor: Colors.*transparent*));  
}  
  
class MyApp extends StatelessWidget {  
 const MyApp({Key? key}) : super(key: key);  
  
 // This widget is the root of your application.  
 @override  
 Widget build(BuildContext context) {  
 return MaterialApp(  
 debugShowCheckedModeBanner: false,  
 routes: {  
 'ChewieDemo': (context) => ChewieDemo(),  
 },  
 title: 'Flutter Demo',  
 theme: ThemeData(  
 primarySwatch: Colors.*blue*,  
 ),  
 home: ChewieDemo(),  
 );  
 }  
}

ChewieDemo

import 'package:chewie/chewie.dart';  
import 'package:flutter/material.dart';  
import 'package:video\_player/video\_player.dart';  
import 'package:video\_player\_test/global.dart';  
  
class ChewieDemo extends StatefulWidget {  
 @override  
 \_ChewieDemoState createState() => \_ChewieDemoState();  
}  
  
class \_ChewieDemoState extends State<ChewieDemo> {  
 late VideoPlayerController \_videoPlayerController;  
 late ChewieController \_chewieController;  
  
 @override  
 void initState() {  
 super.initState();  
  
 // 初始化VideoPlayerController  
 \_videoPlayerController =  
 VideoPlayerController.network('${Global.*wordData*['video']['url']}');  
 // VideoPlayerController.network(  
 // 'http://vfx.mtime.cn/Video/2021/07/10/mp4/210710171112971120.mp4');  
  
 // 初始化ChewieController  
 \_chewieController = ChewieController(  
 maxScale: 6.5,  
 videoPlayerController: \_videoPlayerController,  
 autoPlay: true,  
 looping: true,  
 // fullScreenByDefault: true,  
 subtitle: Subtitles(subtitle()),  
 // 其他配置选项...  
 subtitleBuilder: (context, subtitle) {  
 return Container(  
 padding: EdgeInsets.all(10.0),  
 child: Text(  
 subtitle,  
 style: TextStyle(color: Colors.*white*, fontSize: 25.0),  
 textAlign: TextAlign.center,  
 ),  
 );  
 },  
 );  
 }  
  
 List<Subtitle?> subtitle() {  
 List<Subtitle?> list = [];  
 List.generate(Global.*wordData*['subtitles'][0]['lines'].length, (index) {  
 var mapEnglish = Global.*wordData*['subtitles'][0]['lines'][index] as Map;  
  
 var mapChinese = Global.*wordData*['subtitles'][1]['lines'][index] as Map;  
  
 String startTimeStr = "${mapEnglish['startString']}";  
 String endString = "${mapEnglish['endString']}";  
 RegExp regExp = RegExp(r"(\d+):(\d+):(\d+),(\d+)");  
  
 RegExpMatch? startMatch = regExp.firstMatch(startTimeStr);  
 RegExpMatch? endMatch = regExp.firstMatch(endString);  
  
 if (startMatch != null && endMatch != null) {  
 String start\_hour = startMatch.group(1)!;  
 String start\_minute = startMatch.group(2)!;  
 String start\_second = startMatch.group(3)!;  
 String start\_millisecond = startMatch.group(4)!;  
  
 String end\_hour = endMatch.group(1)!;  
 String end\_minute = endMatch.group(2)!;  
 String end\_second = endMatch.group(3)!;  
 String end\_millisecond = endMatch.group(4)!;  
  
 String sentence = "${mapChinese['words']}";  
 String keyword = "sample";  
 String chineseWords = '';  
  
 List<String> splitSentence = sentence.split(keyword);  
  
 if (splitSentence.length > 1) {  
 String firstPart = splitSentence[0].trim();  
 String secondPart = splitSentence[1].trim();  
  
 print("First part: $firstPart");  
 print("Second part: $secondPart");  
 }  
 if (splitSentence.length > 2) {  
 String thirdPart = splitSentence[2].trim();  
 print("Third part: $thirdPart");  
 }  
  
 list.add(Subtitle(  
 index: index,  
 text: "${mapEnglish['words']}\n${mapChinese['words']}",  
 start: Duration(  
 hours: int.*parse*(start\_hour),  
 minutes: int.*parse*(start\_minute),  
 seconds: int.*parse*(start\_second),  
 milliseconds: int.*parse*(start\_millisecond)),  
 end: Duration(  
 hours: int.*parse*(end\_hour),  
 minutes: int.*parse*(end\_minute),  
 seconds: int.*parse*(end\_second),  
 milliseconds: int.*parse*(end\_millisecond)),  
 ));  
 }  
 });  
 return list;  
 }  
  
 @override  
 void dispose() {  
 // 释放资源  
 \_videoPlayerController.dispose();  
 \_chewieController.dispose();  
 super.dispose();  
 }  
  
 @override  
 Widget build(BuildContext context) {  
 return Scaffold(  
 body: Container(  
 color: Colors.*blueGrey*,  
 child: Stack(  
 children: [  
 Positioned(  
 child: Chewie(  
 controller: \_chewieController,  
 ),  
 ),  
 Positioned(  
 left: 0,  
 right: 0,  
 top: 25,  
 child: Row(  
 mainAxisAlignment: MainAxisAlignment.center,  
 children: [  
 Spacer(),  
 Text(  
 '首页',  
 style: TextStyle(color: Color(0xffB5C273), fontSize: 25),  
 ),  
 SizedBox(  
 width: 20,  
 ),  
 Text(  
 '词库',  
 style: TextStyle(color: Color(0xff3F3F3F), fontSize: 25),  
 ),  
 Spacer(),  
 ClipRRect(  
 borderRadius: BorderRadius.circular(100),  
 child: Image.network(  
 '${Global.*wordData*['user']['avatar']}',  
 width: 30,  
 height: 30,  
 fit: BoxFit.fill,  
 ),  
 ),  
 SizedBox(  
 width: 10,  
 ),  
 ],  
 ),  
 ),  
 Positioned(  
 left: 0,  
 right: 0,  
 top: 70,  
 child: Column(  
 children: [  
 Text(  
 '${Global.*wordData*['words'][0]['wordsStr']}',  
 textAlign: TextAlign.center,  
 style: TextStyle(color: Color(0xff8151FA), fontSize: 35),  
 ),  
 SizedBox(  
 height: 8,  
 width: 1000,  
 ),  
 Text(  
 '[${Global.*wordData*['words'][0]['phonetic']}]',  
 textAlign: TextAlign.center,  
 style: TextStyle(color: Colors.*white*, fontSize: 25),  
 ),  
 SizedBox(  
 height: 8,  
 ),  
 Text(  
 '${Global.*wordData*['words'][0]['chineseExplain']}',  
 style: TextStyle(color: Colors.*white*, fontSize: 20),  
 maxLines: 1,  
 ),  
 ],  
 ),  
 ),  
 Positioned(  
 right: 0,  
 top: 0,  
 bottom: 0,  
 child: Center(  
 child: ClipRRect(  
 borderRadius: BorderRadius.circular(100),  
 child: Image.network(  
 '${Global.*wordData*['user']['avatar']}',  
 width: 50,  
 height: 50,  
 fit: BoxFit.fill,  
 ),  
 ),  
 ),  
 ),  
 ],  
 ),  
 ),  
 );  
 }  
}