

```
1 package com.example.youtube
2 import android.os.Bundle
3 import android.view.ViewGroup
4 import android.webkit.WebChromeClient
5 import android.webkit.WebSettings
6 import android.webkit.WebView
7 import androidx.activity.ComponentActivity
8 import androidx.activity.compose.setContent
9 import androidx.compose.foundation.layout.Column
10 import androidx.compose.foundation.layout.fillMaxSize
11 import androidx.compose.foundation.layout.height
12 import androidx.compose.foundation.layout.width
13 import androidx.compose.material3.MaterialTheme
14 import androidx.compose.material3.Surface
15 import androidx.compose.runtime.Composable
16 import androidx.compose.ui.Modifier
17 import androidx.compose.ui.unit.dp
18 import androidx.compose.ui.viewinterop.AndroidView
19 import com.example.youtube.ui.theme.YoutubeTheme
20
21 class MainActivity : ComponentActivity() {
22     private val VIDEO_ID_1 = "HDVyFsFUuVg"
23     private val VIDEO_ID_2 = "2hIY1xuImuQ&t=218s"
24     private val VIDEO_ID_3 = "dnhQzVGywQ"
25     override fun onCreate(savedInstanceState: Bundle
26         ?) {
27         super.onCreate(savedInstanceState)
28         setContent {
29             YoutubeTheme {
30                 Surface(
31                     modifier = Modifier.fillMaxSize
32                     (),
33                 ) {
34                     Column(modifier = Modifier.
35                         fillMaxSize()) {
36                         YouTubePlayerWebView(
37                             VIDEO_ID_1)
38                         YouTubePlayerWebView(
39                             VIDEO_ID_2)
40                         YouTubePlayerWebView(
41                             VIDEO_ID_3)
```

```
36
37         }
38     }
39 }
40 }
41 }
42 @Composable
43 private fun YouTubePlayerWebView(videoId: String
44 ) {
45     AndroidView(
46         factory = { context ->
47             WebView(context).apply {
48                 layoutParams = ViewGroup.
49                     LayoutParams(
50                         ViewGroup.LayoutParams.
51                             MATCH_PARENT,
52                             ViewGroup.LayoutParams.
53                                 MATCH_PARENT
54                             )
55                 settings.apply {
56                     javaScriptEnabled = true
57
58                     mediaPlaybackRequiresUserGesture = false
59                     allowFileAccess = false
60                     allowContentAccess = false
61                     allowFileAccessFromFileURLs
62                     = false
63                     allowUniversalAccessFromFileURLs = false
64                     mixedContentMode =
65                         WebSettings.
66                             MIXED_CONTENT_NEVER_ALLOW
67                 }
68                 webChromeClient = WebChromeClient
69                 loadVideo(videoId)
```

```
68
69         }
70     },
71     update = { webView ->
72         webView.loadVideo(videoId)
73     },
74     modifier = Modifier
75         .height(244.dp)
76         .width(380.dp)
77 )
78 }
79 private fun WebView.loadVideo(videoId: String) {
80     val videoUrl =
81         "https://www.youtube.com/embed/$videoId?
autoplay=1&fs=0&rel=0&showinfo=0&loop=1"
82     loadUrl(videoUrl)
83 }
```