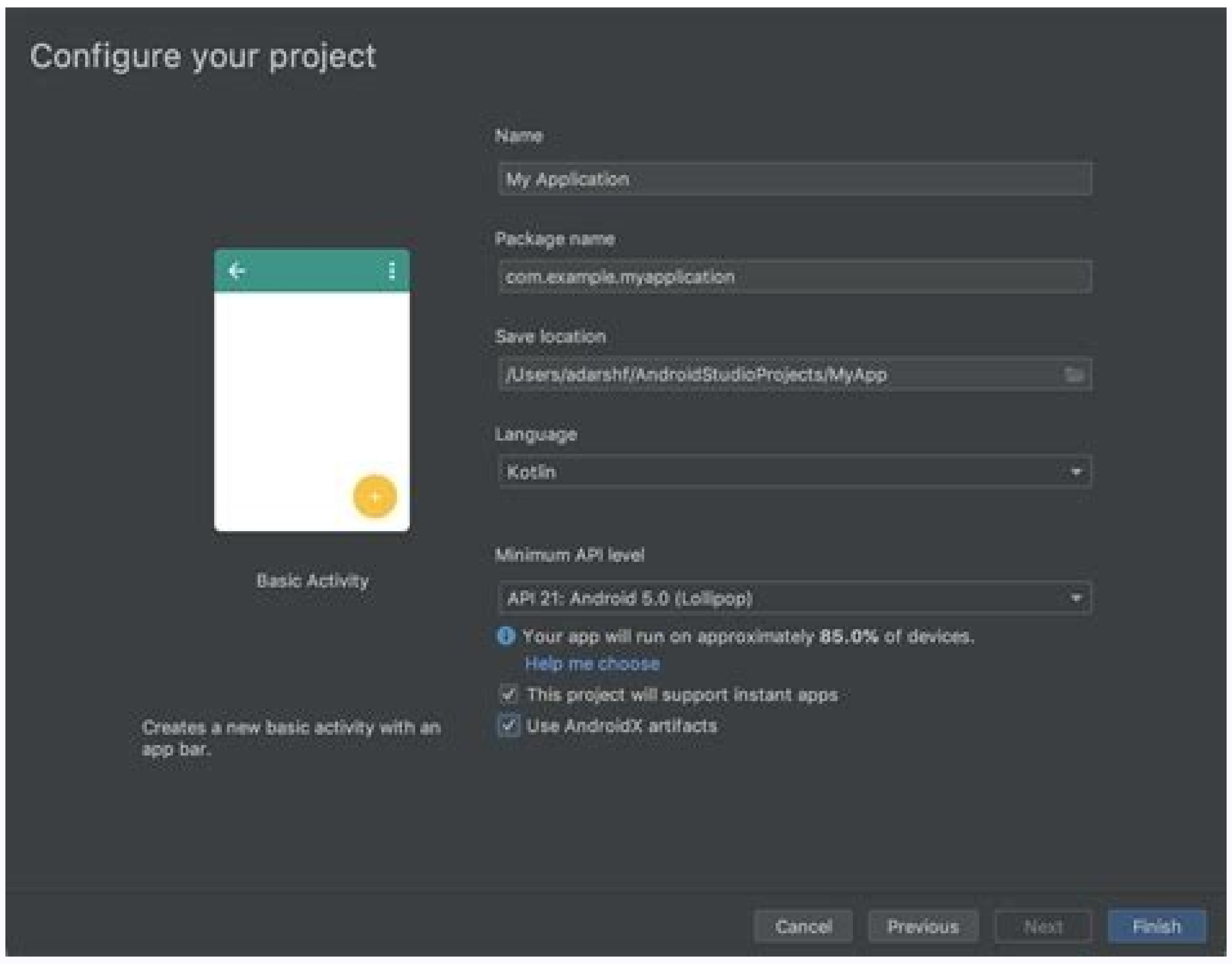
Change api in android studio

Continue



Configure your project

Name: My Application

Package name: com.example.myapplication

Save location: /Users/adarshf/AndroidStudioProjects/MyApp

Language: Kotlin

Minimum API level: API 21: Android 5.0 (Lollipop)

Your app will run on approximately **85.0%** of devices.

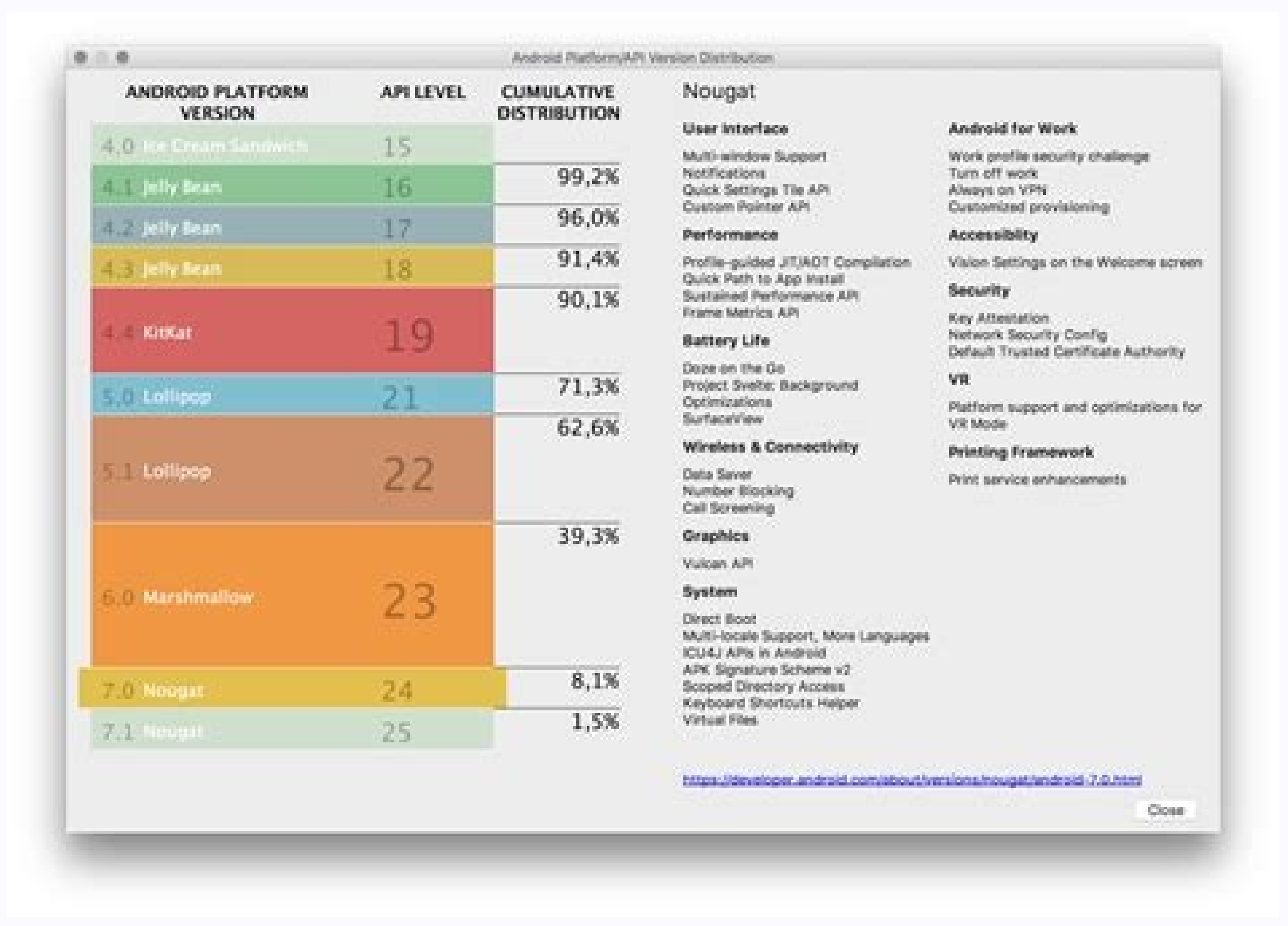
Help me choose

☑ This project will support instant apps

☑ Use AndroidX artifacts

Basic Activity

Creates a new basic activity with an app bar.



Android Platform/API Version Distribution

ANDROID PLATFORM VERSION	API LEVEL	CUMULATIVE DISTRIBUTION
4.0 Ice Cream Sandwich	15	
4.1 Jelly Bean	16	99,2%
4.2 Jelly Bean	17	96,0%
4.3 Jelly Bean	18	91,4%
4.4 KitKat	19	90,1%
5.0 Lollipop	21	71,3%
5.1 Lollipop	22	62,6%
6.0 Marshmallow	23	39,3%
7.0 Nougat	24	8,1%
7.1 Nougat	25	1,5%

Nougat

User Interface
Multi-window Support
Notifications
Quick Settings Tile API
Custom Pointer API

Performance
Profile-guided JIT/AOT Compilation
Quick Path to App Install
Sustained Performance API
Frame Metrics API

Battery Life
Doze on the Go
Project Svelte: Background Optimizations
SurfaceView

Wireless & Connectivity
Data Saver
Number Blocking
Call Screening

Graphics
Vulcan API

System
Direct Boot
Multi-locale Support, More Languages
ICU4J APIs in Android
APK Signature Scheme v2
Scoped Directory Access
Keyboard Shortcuts Helper
Virtual Files

Android for Work
Work profile security challenge
Turn off work
Always on VPN
Customized provisioning

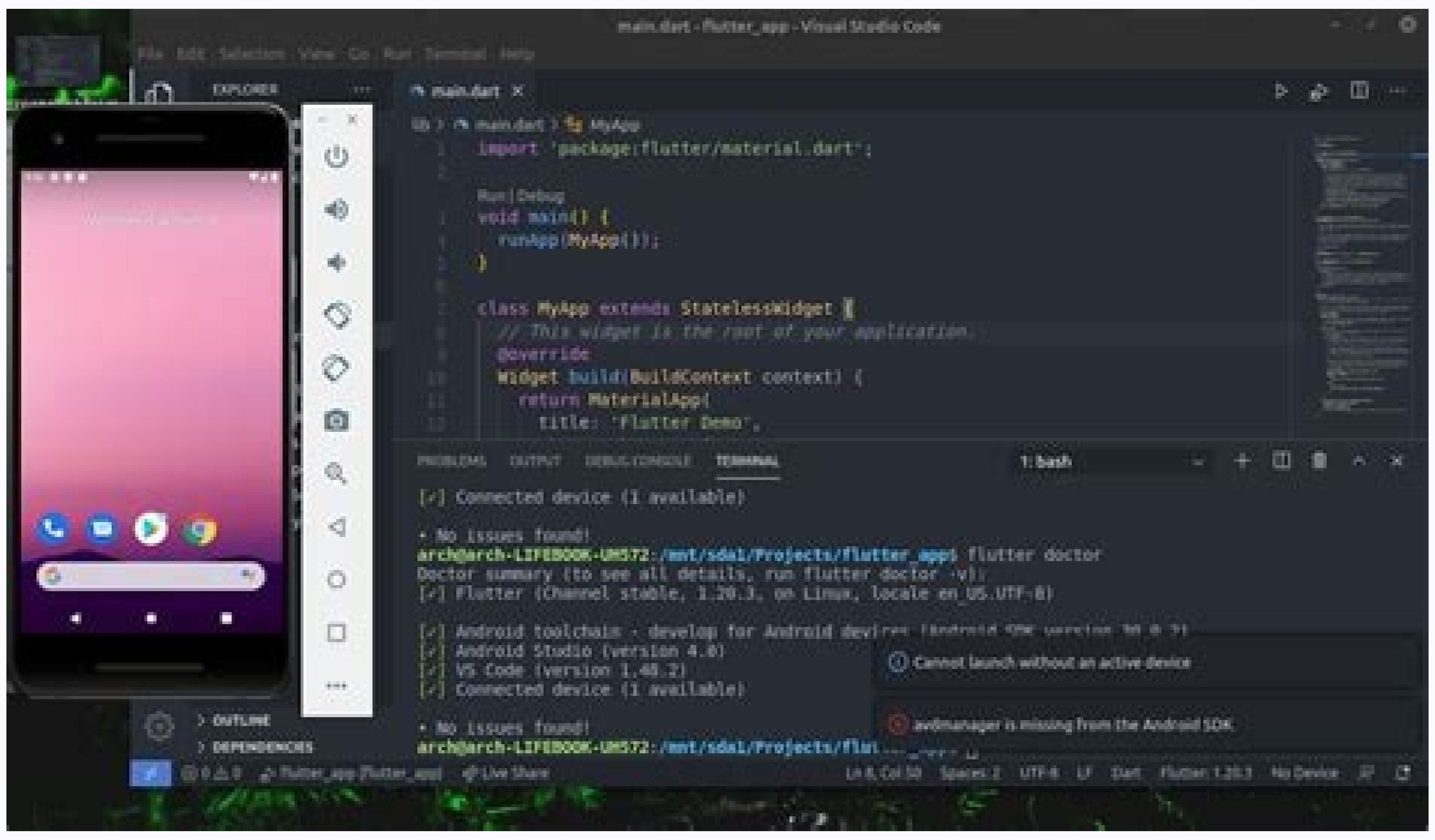
Accessibility
Vision Settings on the Welcome screen

Security
Key Attestation
Network Security Config
Default Trusted Certificate Authority

VR
Platform support and optimizations for VR Mode

Printing Framework
Print service enhancements

https://developer.android.com/about/versions/nougat/android-7.0.html



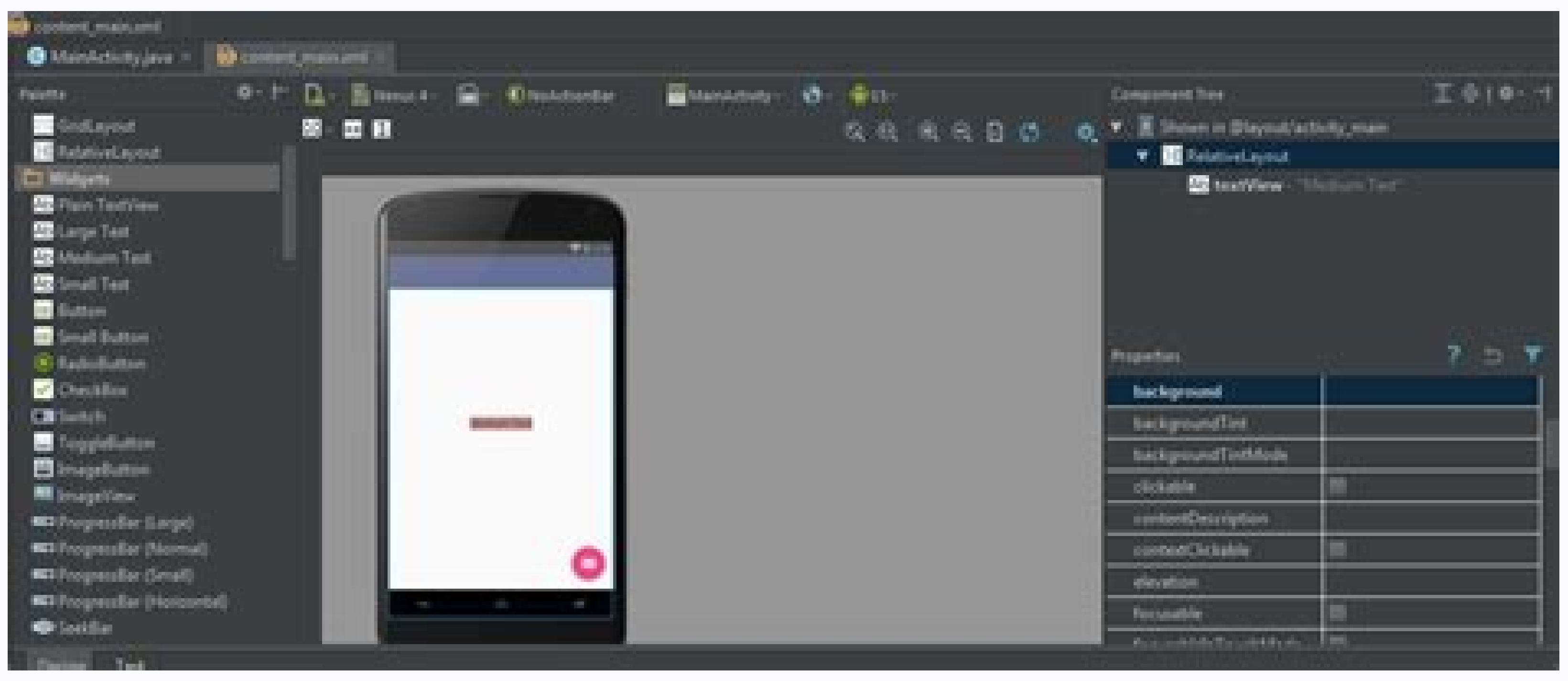
main.dart - flutter_app - Visual Studio Code

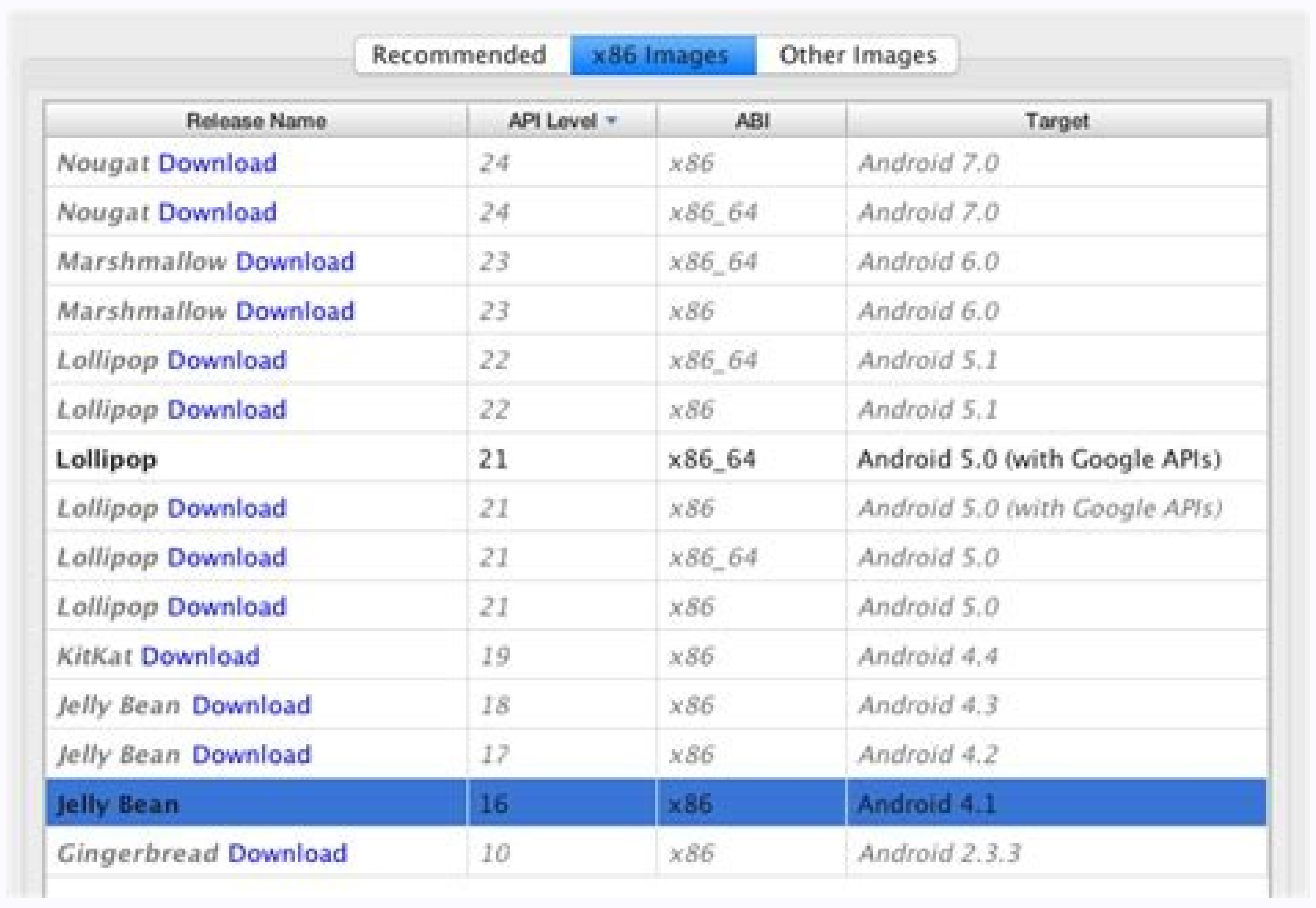
```
import 'package:flutter/material.dart';

Run | Debug
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
```

```
[✓] Connected device (1 available)

• No issues found!
arch@arch-LIFEBOOK-UH572:/mnt/sda1/Projects/flutter_app$ flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 1.20.3, on Linux, locale en_US.UTF-8)

[✓] Android toolchain - develop for Android devices (Android SDK version 30.0.2)
[✓] Android Studio (version 4.0)
[✓] VS Code (version 1.48.2)
[✓] Connected device (1 available)

• No issues found!
arch@arch-LIFEBOOK-UH572:/mnt/sda1/Projects/flut
```

Cannot launch without an active device

avdmanager is missing from the Android SDK





Release Name	API Level	ABI	Target
Nougat Download	24	x86	Android 7.0
Nougat Download	24	x86_64	Android 7.0
Marshmallow Download	23	x86_64	Android 6.0
Marshmallow Download	23	x86	Android 6.0
Lollipop Download	22	x86_64	Android 5.1
Lollipop Download	22	x86	Android 5.1
Lollipop	21	x86_64	Android 5.0 (with Google APIs)
Lollipop Download	21	x86	Android 5.0 (with Google APIs)
Lollipop Download	21	x86_64	Android 5.0
Lollipop Download	21	x86	Android 5.0
KitKat Download	19	x86	Android 4.4
Jelly Bean Download	18	x86	Android 4.3
Jelly Bean Download	17	x86	Android 4.2
Jelly Bean	16	x86	Android 4.1
Gingerbread Download	10	x86	Android 2.3.3

How to change api sdk level in android studio. How to use api in android studio. How to use news api in android studio. How to change api level in android studio. How to change minimum api level in android studio. Api in android studio example. What is api in android studio.

I have several Genymotion emulators for different API levels. The problem is this: I'm running the app in an emulator with the "Same device for future run" option enabled. I open another emulator. I restarted the app. After that the app only works on the first emulator and I can't figure out how to bring up the dialog box (select target) again to run the app on both emulators. 1 As per this answer, just don't include minsdkversion in the manifest.xml file and the build system will use the values from the build.gradle file and include the information in the final apk file. Since the build system needs this information anyway, this makes sense. These values should not be entered twice. After modifying the build.gradle file, all you need to do is sync the project, but Android Studio 0.5.2 will show a yellow status bar at the top of the build.gradle editor window to help you. Also note that there are at least two build.gradle files: one for main and one for app/module. What needs to be changed is in the app/module that already has the minSdkVersion property in the newly generated project. Improve article Save article as article In general, the API level refers to the Android version. This determines which version developers target their app and the minimum Android version level their app will run on. To set the minimum and maximum values, Android Studio offers two terminologies. minSdkVersion: This means the minimum version of the Android OS that the application supports and targetSdkVersion: This means the version that developers are actually developing their application for. The app will be compatible with all Android versions between the minimum SDK level and the target SDK level. Sometimes you need to change Android Studio's API layer during development. So we have two methods to change the API level. In this article, we will discuss both methods. This method is very simple and very directYou have to be very careful when making changes here. Step 1: Open the project in android mode, then go to Gradle Scripts > build.gradle (module: app) as shown in the image below. Step 2: See the picture below and here you need to change minSdkVersion and targetSdkVersion as needed. Once you've made your changes as required, click the Sync Now button and you're done. Method 2 Step 1: Open Android Studio and choose File > Project Structure as shown in the image below. Step 2: A pop-up window will appear as shown below. Now select Modules > Default Configuration and scroll down and you will see two sections as shown in the image below in the Default Configuration section. Here you will change the SDK version according to your requirement then click OK button below. And you did it. Note. If you choose the second approach, you don't need to make any changes to Gradle. Automatically updates the degree. Update your module's gradle script. In Gradle Scripts the file appears as build.gradle (module: app). No build.gradle(Project: nameofproject), this is an example plugin to use gradle file: "com.android.application" android { compileSdkVersion 22 buildToolsVersion "25.0.0" defaultConfig { applicationId "com.project" minSdkVersion 18 targetSdkversionCode 1 versionName "1.0 " } buildTypes { release { minifyEnabled false proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro' } } } change minSdkVersion to required. sync gradient and clean your design. I need to change the base url on the fly. I have a login button and when I click on the login button it tells me my login API as below: successful response from the first API, I get the client server url to change the baseUrl. companyUrlConfigEntity companyUrlConfigEntity = response.body(); as shown below: String clientUrl = companyUrlConfigEntity.clientUrl = Basically the client or company in this project. So they have their own server. Each company uses more than 20 APIs. So I need to change the base url. I will also check to change the base url from the following link: and modified code like this public static void changeApiBaseUrl(String newApiBaseUrl) { API_BASE_URL = newApiBaseUrl; builder = new Retrofit.Builder() .baseUrl(API_BASE_URL) .addConverterFactory(new NullOnEmptyConverterFactory()) .addConverterFactory(ScalarsConverterFactory.create()) .addConvertersonNewcrectory))); } When I debugged and checked my baseUrl it looked like API_BASE_URL = But when I call the client API it turns out that my first call to the base url which happened to be url didn't change. Expected Customers API: Reality Customers API: I also verified from the following link: retrofit-2 -how-to-use-dynamic-urls-for-requests which works but each API must pass the fullPath URL with each API as below: @GET public Call profilePicture(@Url String url); But when using this method, each API call site must add the full path to the URL. Are there other options? Help me please. ServiceGenerator.class public class ServiceGenerator { public static string API_BASE_URL = " ; private static update; private static OkHttpClient.Builder httpClient = new OkHttpClient.Builder(); private static Retrofit.Builder builder = new Retrofit.Builder() .baseUrl(API_BASE_URL) .addConverterFactory(new NullOnEmptyConverterFactory()) .addConverterFactory(ScalarsConverterFactory.create)FreadConverterFactory(ConversonGactory()); private ServiceGenerator() { } public static void changeApiBaseUrl (String newApiBaseUrl) { API_BASE_URL = newApiBaseUrl;= new Retrofit.Builder() .baseUrl(API_BASE_URL) .addConverterFactory(new NullOnEmptyConverterFactory()) .addConverterFactory(ScalarsConverterFactory.create()) .addConverterFactory ()) .addConverterFactory ()) . } public static S createService(Class serviceClass) { return createService(serviceClass, null, null); } public static S createService(Class serviceClass, final String authToken, final ProgressListener progressListener) { if (authToken != null) { httpClient.addInterceptor(new Interceptor() { @Override public Response intercept(Chain chain) throws IOException { Request original = chain.request(); Final String headerValue = AUTHORIZATION_TYPE + authToken; Request request = original.newBuilder() .header(AUTHORIZATION_HEADER_KEY, headerValue).method(original.body()()), original. . build(); return chain.proceed(request); } }); } addResponseProgressListener(progressListener); if (BuildConfig.DEBUG) { HttpLoggingInterceptor httpLoggingInterceptor = new HttpLoggingInterceptor(); httpLoggingInterceptor.setLevel(HttpLoggingInterceptor.Level.BODY); httpClient.addInterceptor(httpLoggingInterceptor); } if (authToken != null) { if (picasso == null) { setUpPicasso(authToken); } } OkHttpClient client = httpClient.build(); httpClient.connectTimeout(15, TimeUnit.SECONDS); httpClient.readTimeout(2, TimeUnit.MINUTES); httpClient.writeTimeout(2, TimeUnit.MINUTES); upgrade = builder.client(client).build(); return retrograde.create(serviceClass); } } LoginFragment.java @OnClick(R.id.bt_login) void onLogin() { checkValidityOfUser(); } private void checkValidityOfUser() { final login login = getLoginCredentials(); Call callCheckValidity = dataProcessController. getApiClient(). checkValidityOfUsers(login.getUsername()); callCheckValidity.enqueue(new Callback() { @Override public void onResponse(Call call,response) { if (response.code() == 200) { companyUrlConfigEntity companyUrlConfigEntity = response.body(); boolean status = firmUrlConfigEntity.isValidUser(); if (status) { String baseUrls = companyUrlConfigEntity . getBaseUrl(); baseUrls = baseUrls + "/api/"; ServiceGenerator.changeApiBaseUrl(baseUrls); Log in(); } else { ToastHelper.show("contact admin"); } } else { ToastHelper.show("" + response.code() + response.message()); } } @Override public void onFailure(Call call, Throwable t) { ToastHelper.show("Contact Admin"); } }); } private cannot login() { login = getLoginCredentials(); Call callLogin = DataProcessController. getApiClient(). login (login); callLogin.enqueue(new Callback() { @Override public void onResponse(Call call, Response response) { if (response.code() == 200) { } else if (response.code () == 401) { } } @Override public void onFailure(Call call, Throwable t) { } }); } }