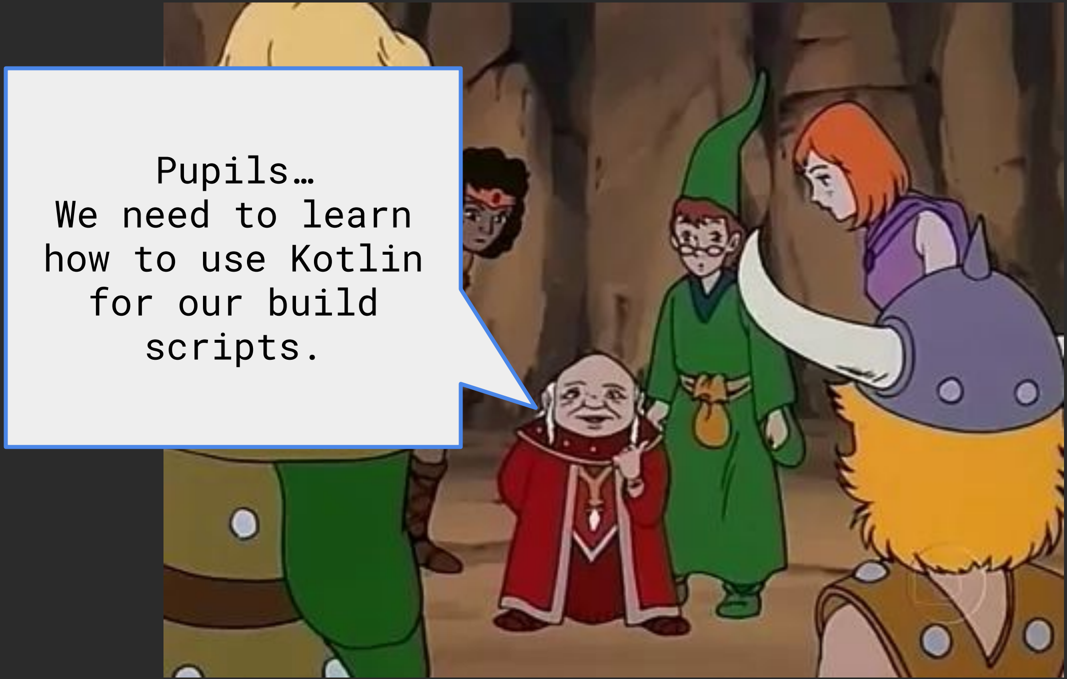
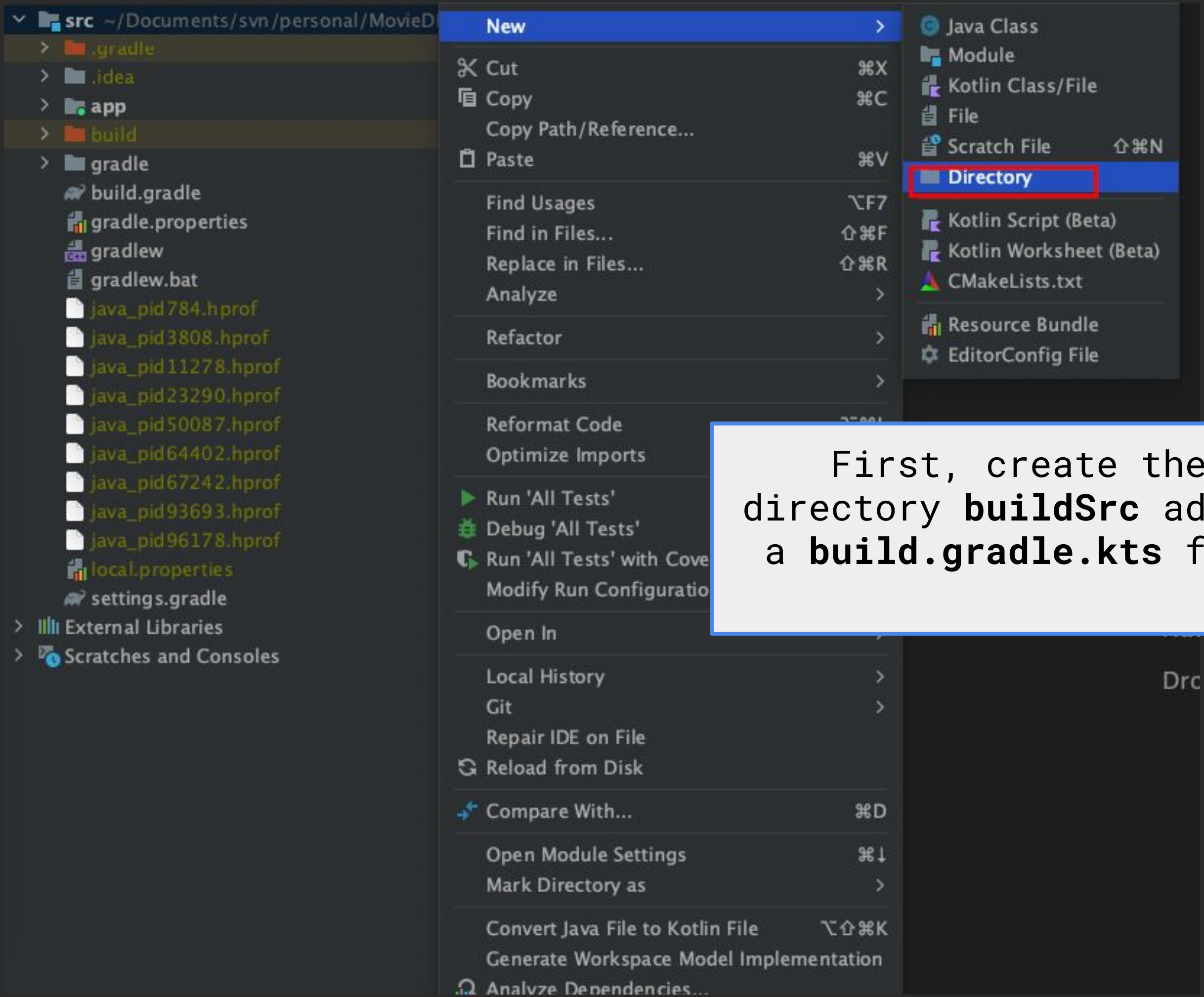
Build Configuration

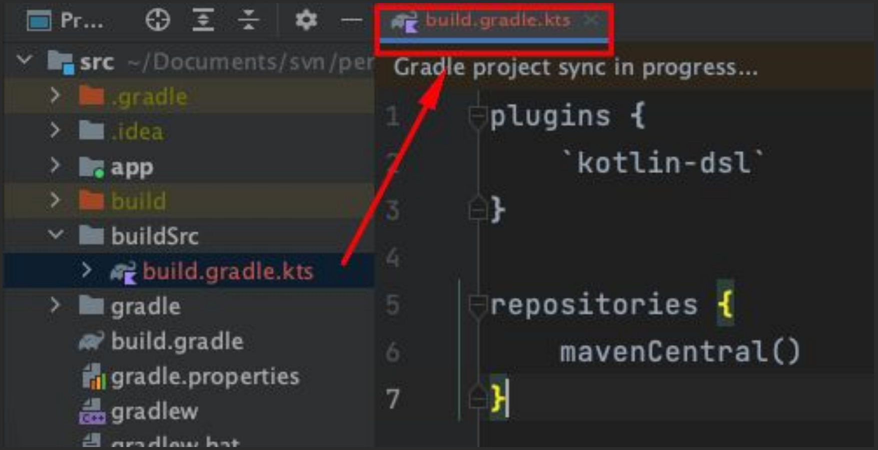
From Groovy to Kotlin DSL

By Moro


Pupils…
We need to learn how to use Kotlin for our build scripts.

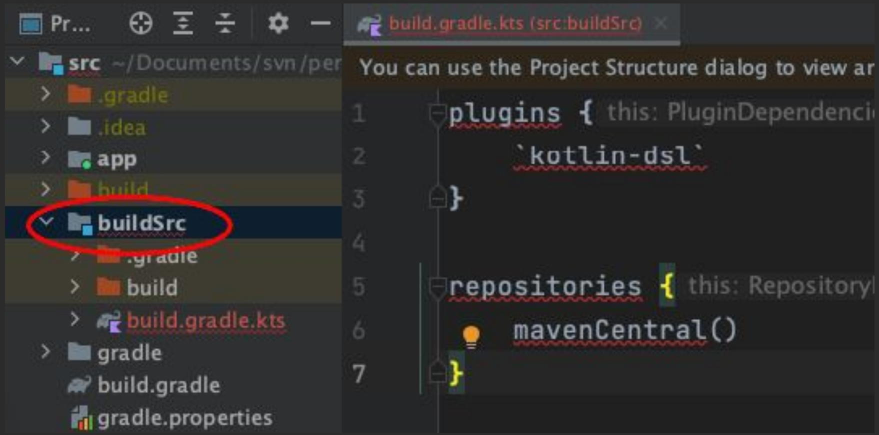


First, create the directory **buildSrc** adding a **build.gradle.kts** file



Gradle project sync in progress...

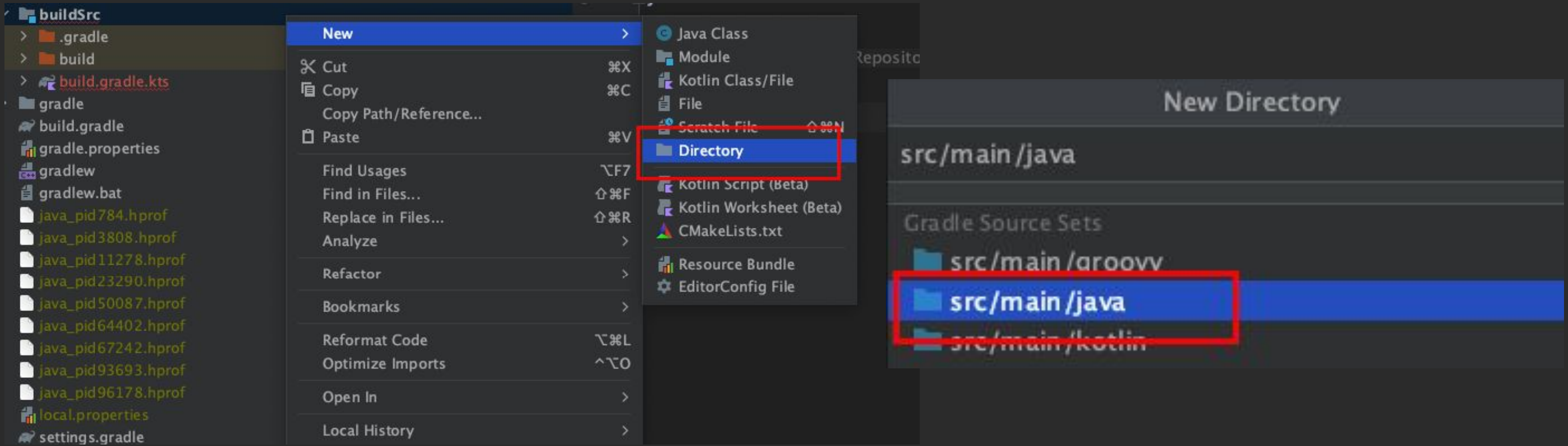
```
plugins {
    `kotlin-dsl`
}

repositories {
    mavenCentral()
}
```



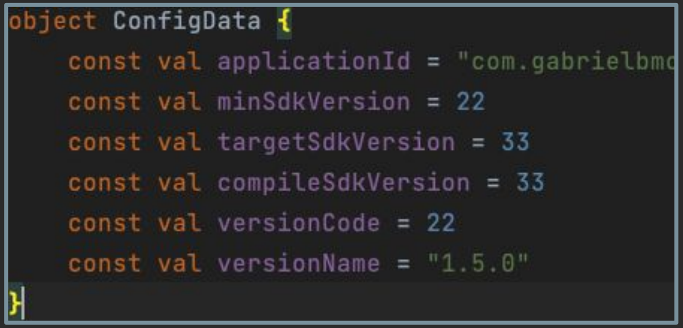
You can use the Project Structure dialog to view ar

```
plugins { this: PluginDependenci
    `kotlin-dsl`
}

repositories { this: Repository
    mavenCentral()
}
```

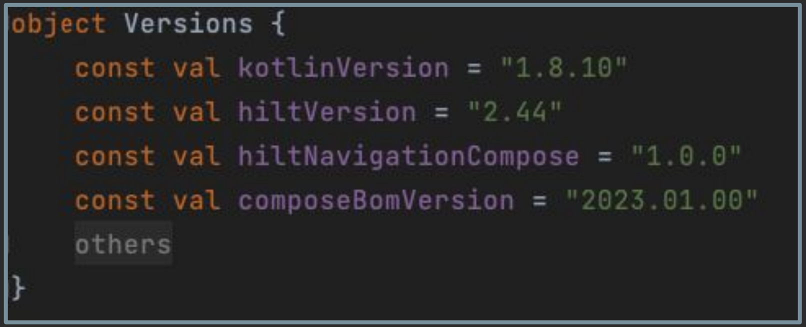
After you sync you will see an indicator in your **buildSrc**. Android Studio recognized.



After create the java directory, let's create two base objects…



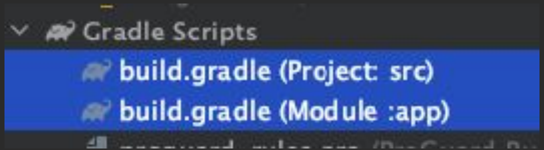
```kotlin
object ConfigData {
    const val applicationId = "com.gabrielbm
    const val minSdkVersion = 22
    const val targetSdkVersion = 33
    const val compileSdkVersion = 33
    const val versionCode = 22
    const val versionName = "1.5.0"
}
```



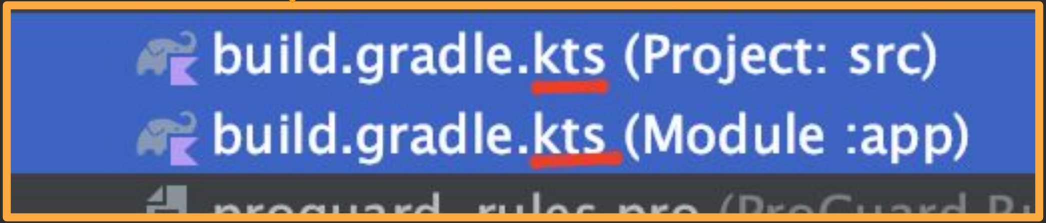
```kotlin
object Versions {
    const val kotlinVersion = "1.8.10"
    const val hiltVersion = "2.44"
    const val hiltNavigationCompose = "1.0.0"
    const val composeBomVersion = "2023.01.00"
    others
}
```

We will define our dependencies versions and our base data using this two files

Now, refactor the name of your build scripts to have the kts extension…



Gradle Scripts
build.gradle (Project: src)
build.gradle (Module :app)
proguard-rules.pro (ProGuard Ru



build.gradle.kts (Project: src)
build.gradle.kts (Module :app)
proguard-rules.pro (ProGuard Ru

Don’t forget pupils:
“*slowly you go far*”

Always the same thing.
Dungeon master says something and go away. Now we have a lot of errors in our Gradle files.

I hate when this little guy go away.

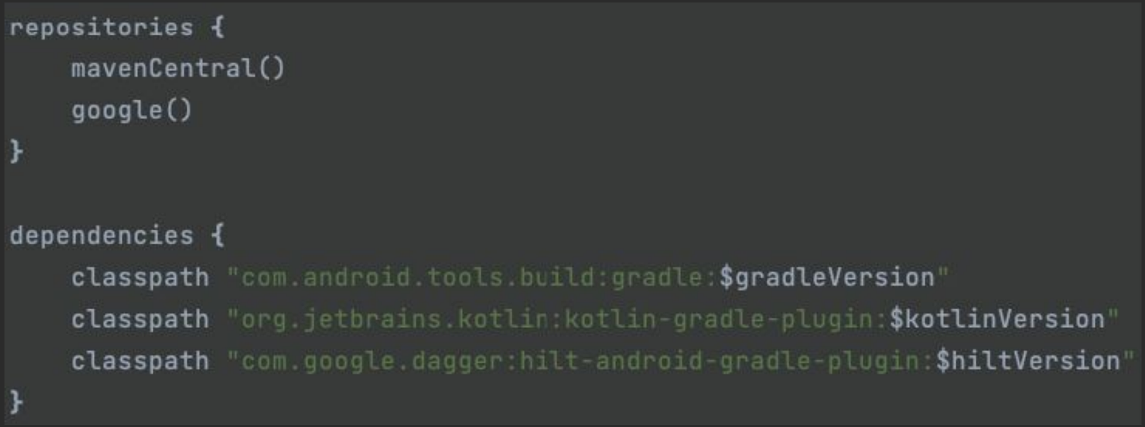
After 4 hours…

I saw you trying. We have a lot of errors isn’t?

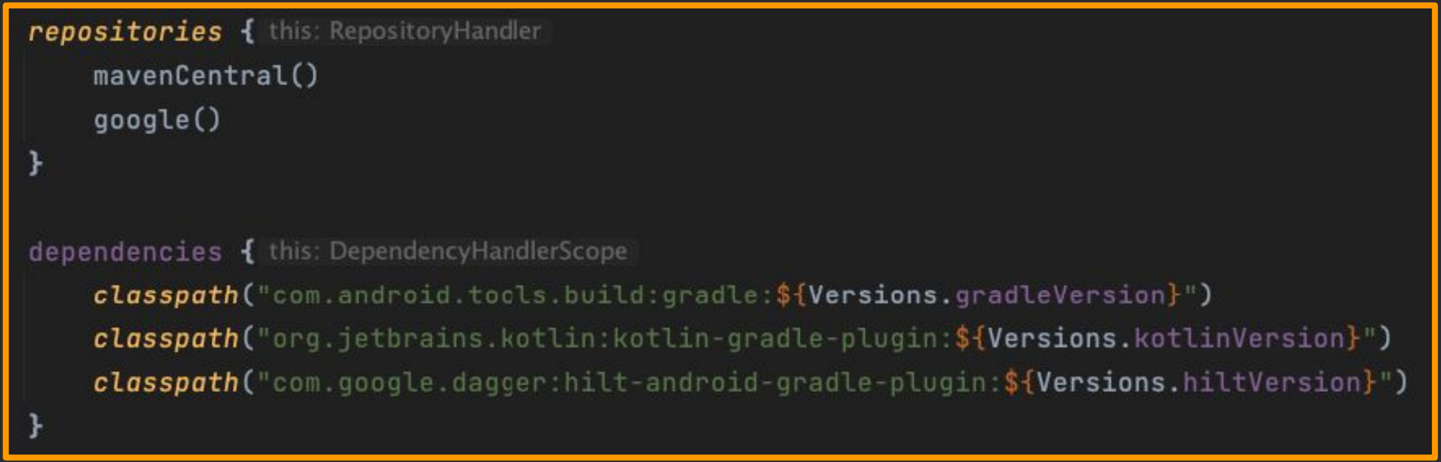
I was having lunch.

I will help you, just this time!

build.gradle -> Project level

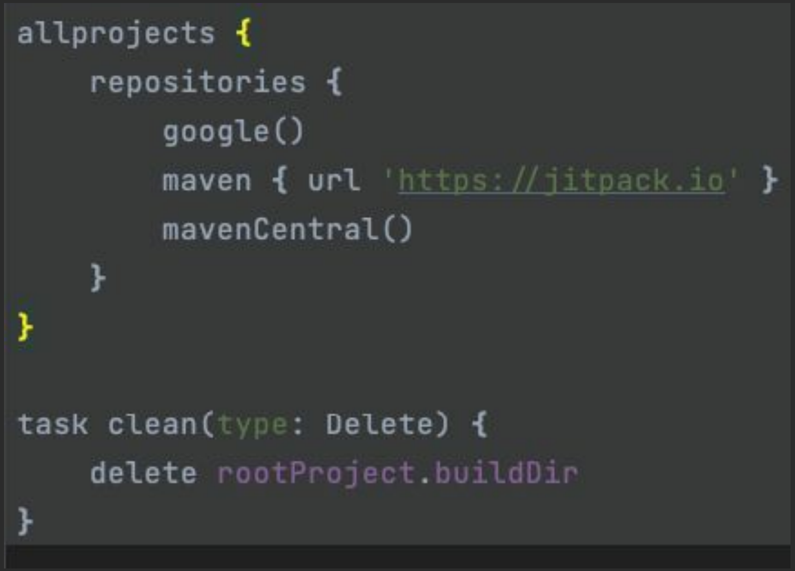


```
repositories {
    mavenCentral()
    google()
}

dependencies {
    classpath "com.android.tools.build:gradle:$gradleVersion"
    classpath "org.jetbrains.kotlin:kotlin-gradle-plugin:$kotlinVersion"
    classpath "com.google.dagger:hilt-android-gradle-plugin:$hiltVersion"
}
```

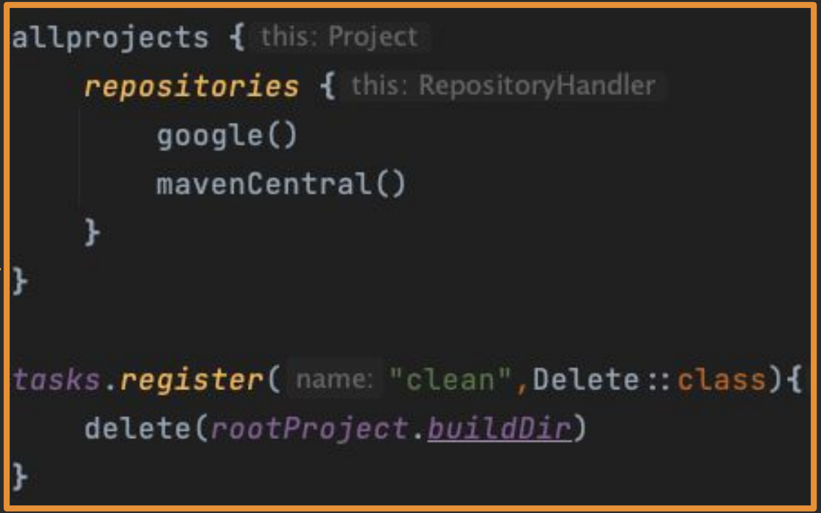


```
repositories { this: RepositoryHandler
    mavenCentral()
    google()
}

dependencies { this: DependencyHandlerScope
    classpath("com.android.tools.build:gradle:${Versions.gradleVersion}")
    classpath("org.jetbrains.kotlin:kotlin-gradle-plugin:${Versions.kotlinVersion}")
    classpath("com.google.dagger:hilt-android-gradle-plugin:${Versions.hiltVersion}")
}
```

build.gradle -> Project level

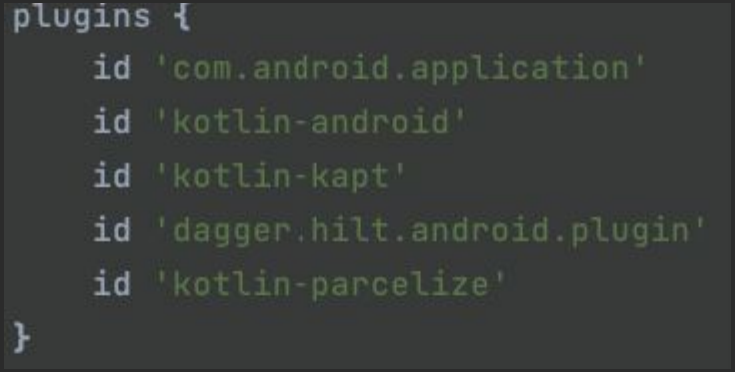


```
allprojects {
    repositories {
        google()
        maven { url 'https://jitpack.io' }
        mavenCentral()
    }
}

task clean(type: Delete) {
    delete rootProject.buildDir
}
```

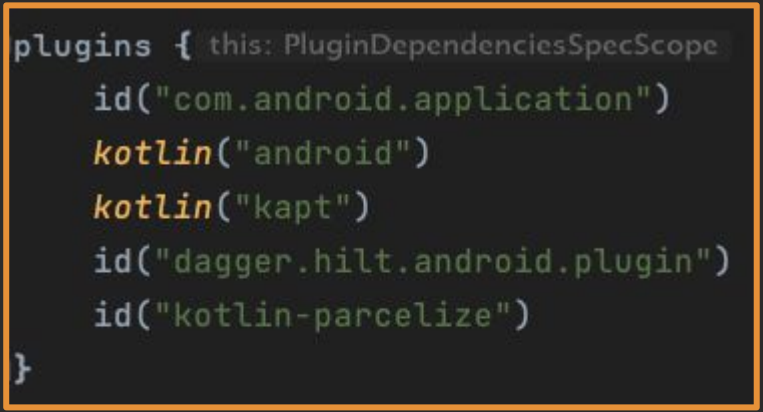


```
allprojects { this: Project
    repositories { this: RepositoryHandler
        google()
        mavenCentral()
    }
}

tasks.register( name: "clean",Delete::class){
    delete(rootProject.buildDir)
}
```

build.gradle -> **app module** level

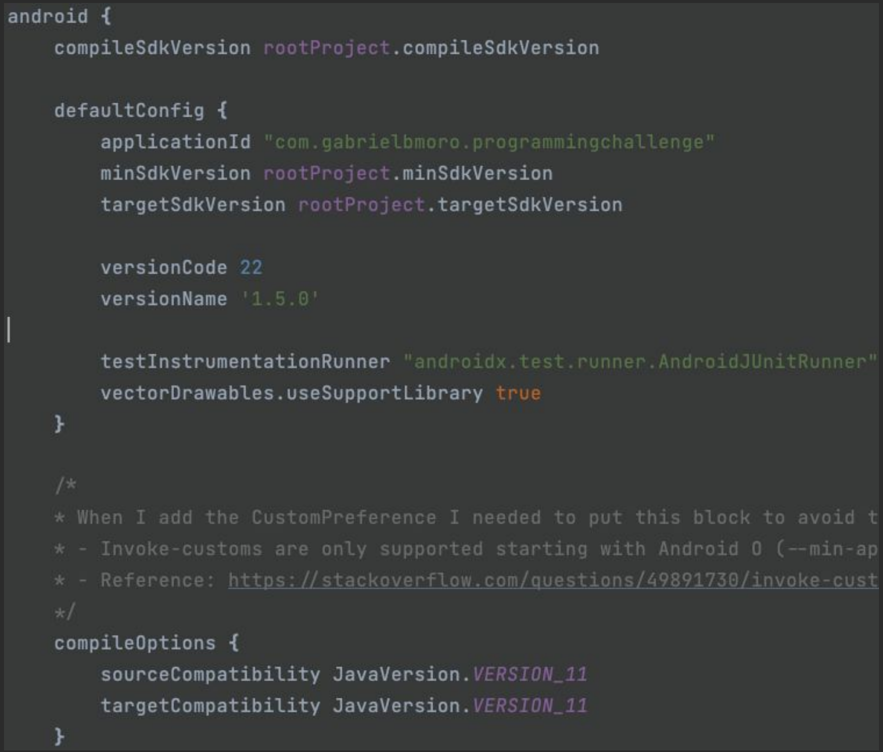


```
plugins {
    id 'com.android.application'
    id 'kotlin-android'
    id 'kotlin-kapt'
    id 'dagger.hilt.android.plugin'
    id 'kotlin-parcelize'
}
```

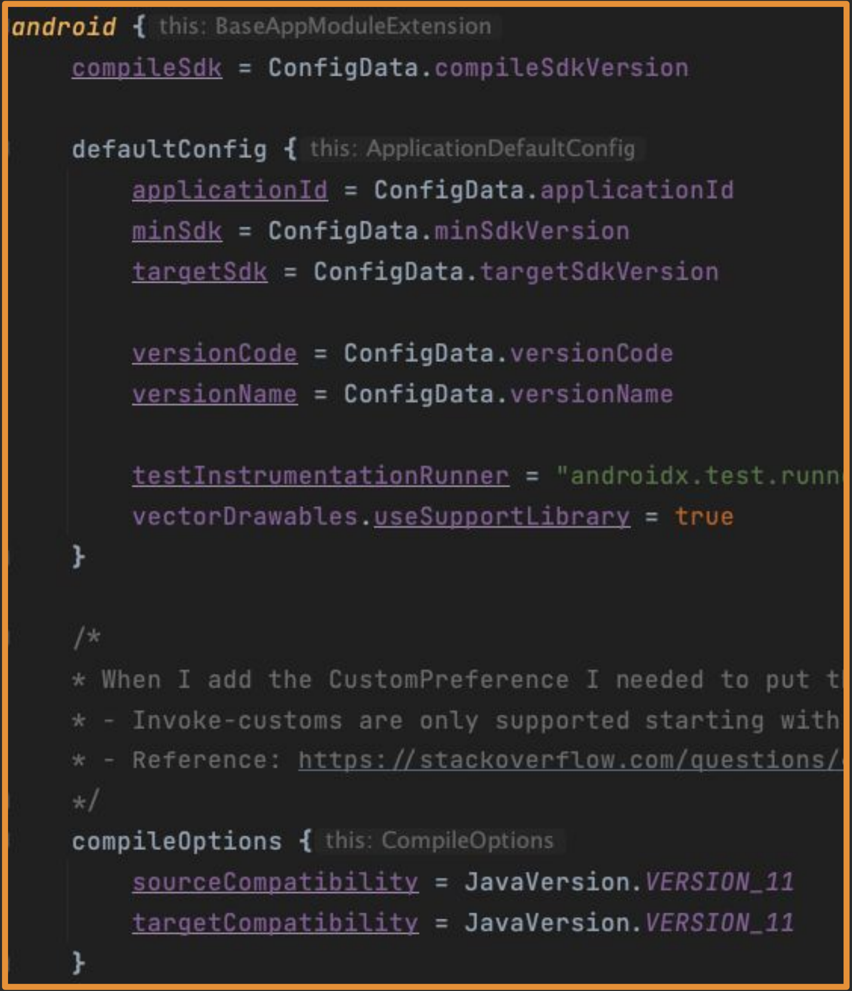


```
plugins { this: PluginDependenciesSpecScope
    id("com.android.application")
    kotlin("android")
    kotlin("kapt")
    id("dagger.hilt.android.plugin")
    id("kotlin-parcelize")
}
```

build.gradle -> **app module** level

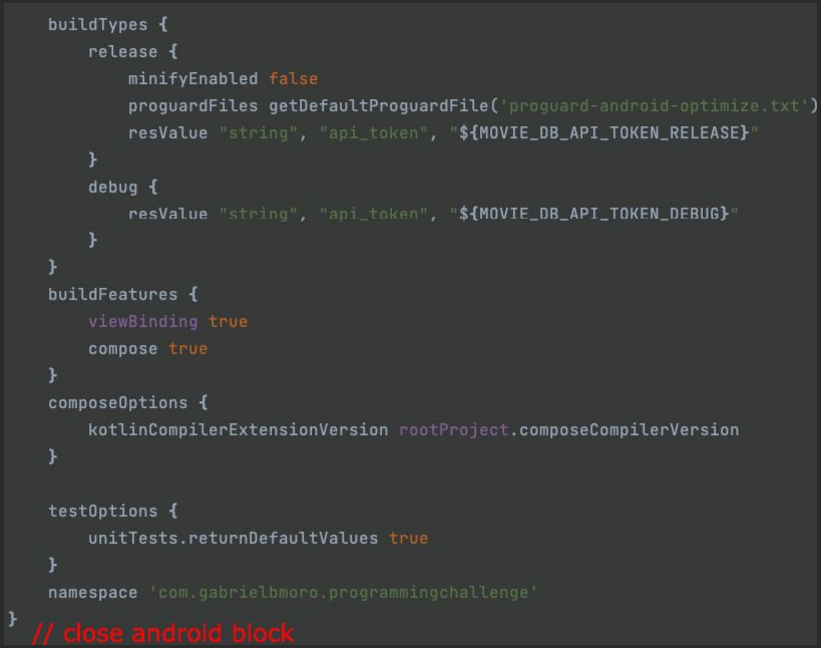


```
android {
    compileSdkVersion rootProject.compileSdkVersion

    defaultConfig {
        applicationId "com.gabrielbmoro.programmingchallenge"
        minSdkVersion rootProject.minSdkVersion
        targetSdkVersion rootProject.targetSdkVersion

        versionCode 22
        versionName '1.5.0'

        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
        vectorDrawables.useSupportLibrary true
    }

    /*
    * When I add the CustomPreference I needed to put this block to avoid t
    * - Invoke-customs are only supported starting with Android O (--min-ap
    * - Reference: https://stackoverflow.com/questions/49891730/invoke-cust
    */
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_11
        targetCompatibility JavaVersion.VERSION_11
    }
```

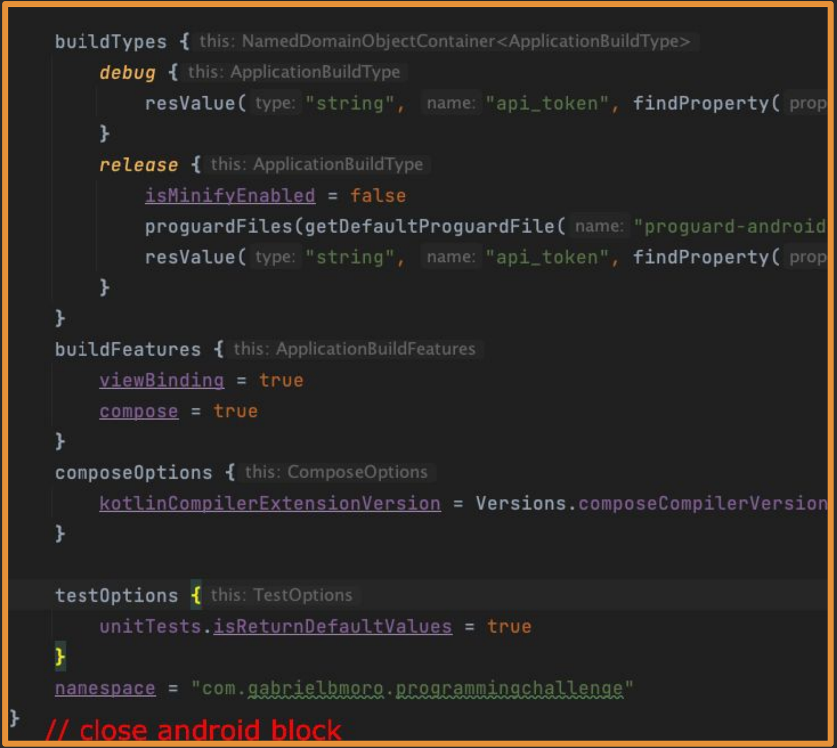


```
android { this: BaseAppModuleExtension
    compileSdk = ConfigData.compileSdkVersion

    defaultConfig { this: ApplicationDefaultConfig
        applicationId = ConfigData.applicationId
        minSdk = ConfigData.minSdkVersion
        targetSdk = ConfigData.targetSdkVersion

        versionCode = ConfigData.versionCode
        versionName = ConfigData.versionName

        testInstrumentationRunner = "androidx.test.runn
        vectorDrawables.useSupportLibrary = true
    }

    /*
    * When I add the CustomPreference I needed to put t
    * - Invoke-customs are only supported starting with
    * - Reference: https://stackoverflow.com/questions/
    */
    compileOptions { this: CompileOptions
        sourceCompatibility = JavaVersion.VERSION_11
        targetCompatibility = JavaVersion.VERSION_11
    }
```

build.gradle -> **app module** level

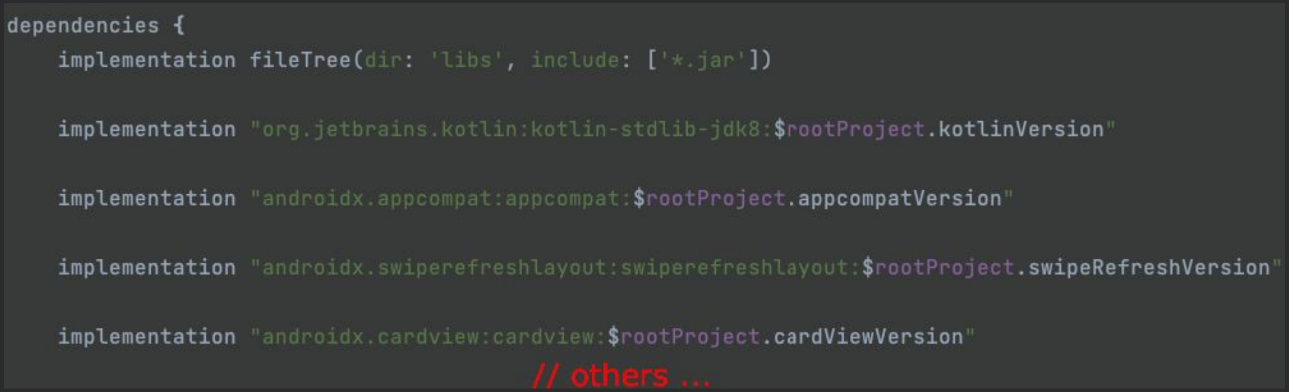


```
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.txt')
            resValue "string", "api_token", "${MOVIE_DB_API_TOKEN_RELEASE}"
        }
        debug {
            resValue "string", "api_token", "${MOVIE_DB_API_TOKEN_DEBUG}"
        }
    }
    buildFeatures {
        viewBinding true
        compose true
    }
    composeOptions {
        kotlinCompilerExtensionVersion rootProject.composeCompilerVersion
    }

    testOptions {
        unitTests.returnDefaultValues true
    }
    namespace 'com.gabrielbmoro.programmingchallenge'
} // close android block
```

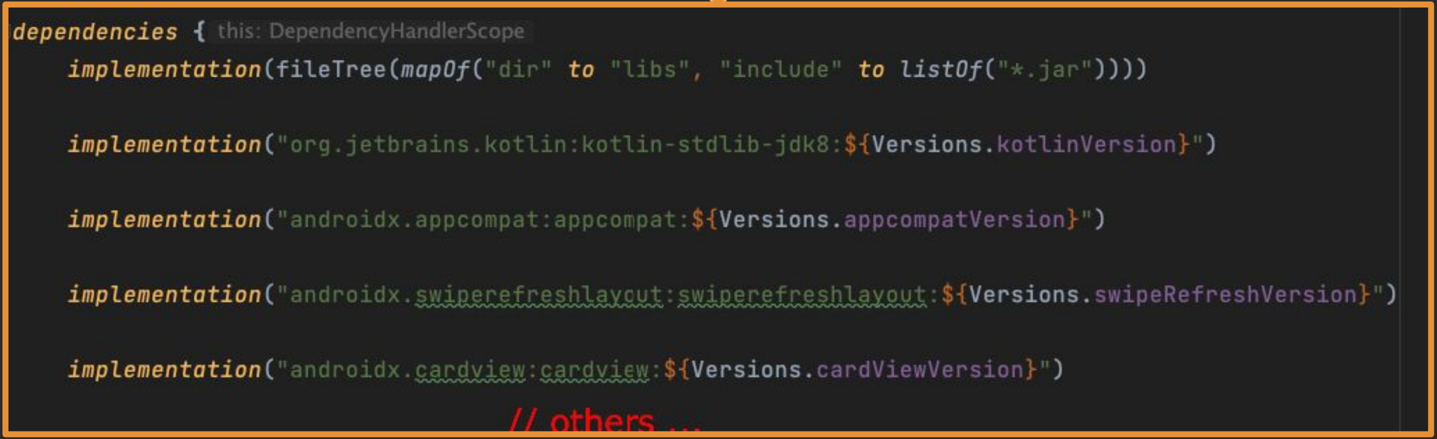


```
    buildTypes { this: NamedDomainObjectContainer<ApplicationBuildType>
        debug { this: ApplicationBuildType
            resValue( type: "string",  name: "api_token", findProperty( prop
        }
        release { this: ApplicationBuildType
            isMinifyEnabled = false
            proguardFiles(getDefaultProguardFile( name: "proguard-android
            resValue( type: "string",  name: "api_token", findProperty( prop
        }
    }
    buildFeatures { this: ApplicationBuildFeatures
        viewBinding = true
        compose = true
    }
    composeOptions { this: ComposeOptions
        kotlinCompilerExtensionVersion = Versions.composeCompilerVersion
    }

    testOptions { this: TestOptions
        unitTests.isReturnDefaultValues = true
    }
    namespace = "com.gabrielbmoro.programmingchallenge"
} // close android block
```

build.gradle -> **app module** level



```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])

    implementation "org.jetbrains.kotlin:kotlin-stdlib-jdk8:$rootProject.kotlinVersion"

    implementation "androidx.appcompat:appcompat:$rootProject.appcompatVersion"

    implementation "androidx.swiperefreshlayout:swiperefreshlayout:$rootProject.swipeRefreshVersion"

    implementation "androidx.cardview:cardview:$rootProject.cardViewVersion"
                                        // others ...
```



```
dependencies { this: DependencyHandlerScope
    implementation(fileTree(mapOf("dir" to "libs", "include" to listOf("*.jar"))))

    implementation("org.jetbrains.kotlin:kotlin-stdlib-jdk8:${Versions.kotlinVersion}")

    implementation("androidx.appcompat:appcompat:${Versions.appcompatVersion}")

    implementation("androidx.swiperefreshlayout:swiperefreshlayout:${Versions.swipeRefreshVersion}")

    implementation("androidx.cardview:cardview:${Versions.cardViewVersion}")

                              // others ...
```

By the way, I'm going to leave you without a shield for a few days. Just for the "*little guy*".

By Moro