

使用 Java 8 语言功能和 API 🔖

Android Gradle 插件 3.0.0 及更高版本支持所有 Java 7 语言功能，以及部分 Java 8 语言功能（具体因平台版本而异）。使用 Android Gradle 插件 4.0.0 及更高版本构建应用时，您可以使用多种 Java 8 语言 API，而无需为应用设置最低 API 级别。

本页介绍您可以使用的 Java 8 语言功能、如何正确配置项目以使用这些功能，以及您可能会遇到的任何已知问题。此外，请观看以下视频，对此有个大致了解。

Android Gradle 插件对使用某些 Java 8 语言功能以及利用这些功能的第三方库提供内置支持。如图 1 所示，默认工具链实现新语言功能的方法是在使用 D8/R8 将类文件编译成 dex 代码的过程中执行字节码转换，这种转换称为 `desugar`。

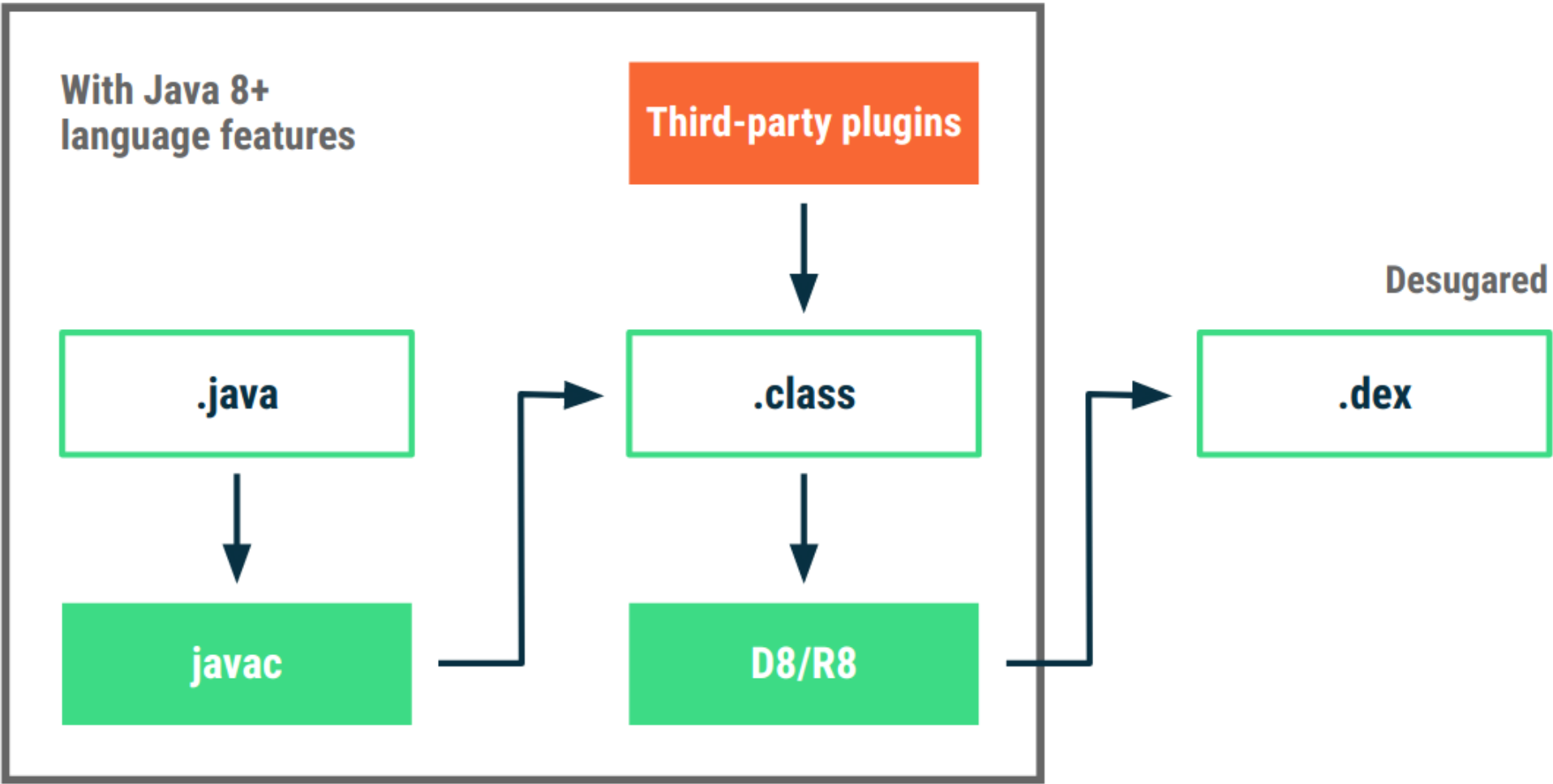


图 1. 通过 `desugar` 字节码转换实现 Java 8 语言功能支持。

Java 8 语言功能支持（Android Gradle 插件 3.0.0 及更高版本）

如需开始使用受支持的 Java 8 语言功能，请将 [Android 插件更新](#) (`/studio/releases/gradle-plugin#updating-plugin`) 为 `3.0.0`（或更高版本）。在此之后，针对使用（在源代码中使用或通过依赖项使用）Java 8 语言功能的每个模块，更新模块的 `build.gradle` 文件，如下所示：

```
GroovyKotlin (#kotlin)
(#groovy)

android {
    ...
    // Configure only for each module that uses Java 8
    // language features (either in its source code or
    // through dependencies).
    compileOptions {
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
    // For Kotlin projects
    kotlinOptions {
        jvmTarget = "1.8"
    }
}
```

使用 Android Gradle 插件 3.0.0 及更高版本构建应用时，插件并非支持所有 Java 8 语言功能。现在，以下语言功能在任何 API 级别上均可用：

Java 8 语言功能	备注
lambda 表达式 (https://docs.oracle.com/javase/tutorial/java/javaOO/lambdaexpressions.html)	请注意，Android 不支持 lambda 表达式的序列化。
方法引用 (https://docs.oracle.com/javase/tutorial/java/javaOO/methodreferences.html)	
类型注解 (https://docs.oracle.com/javase/tutorial/java/annotations/type_annotations.html)	类型注解信息仅在编译时可用，在运行时不可用。此外，在 API 级别 24 及更低级别中，平台支持 <code>ElementType#TYPE</code> (<code>/reference/java/lang/annotation/ElementType#TYPE</code>)，而不支持 <code>ElementType.TYPE_USE</code> 或 <code>ElementType.TYPE_PARAMETER</code> 。
默认和静态接口方法 (https://docs.oracle.com/javase/tutorial/java/landl/defaultmethods.html)	
重复注解 (https://docs.oracle.com/javase/tutorial/java/annotations/repeating.html)	

除了上述 Java 8 语言功能之外，插件版本 3.0.0 及更高版本还将对 `try-with-resources` (<https://docs.oracle.com/javase/tutorial/essential/exceptions/tryResourceClose.html>) 的支持扩展到所有 Android API 级别。

脱糖处理目前暂不支持 `MethodHandle.invoke` (<https://docs.oracle.com/javase/8/docs/api/java/lang/invoke/MethodHandle.html#invoke-java.lang.Object...->) 和 `MethodHandle.invokeExact` (<https://docs.oracle.com/javase/8/docs/api/java/lang/invoke/MethodHandle.html#invokeExact-java.lang.Object...->)。如果您的源代码或某个模块依赖项使用其中某种方法，则需要指定 `minSdkVersion 26` 或更高版本。否则，您会遇到以下错误：

```
Dex: Error converting bytecode to dex:
Cause: signature-polymorphic method called without --min-sdk-version >= 26
```

在某些情况下，即使 `invoke` 或 `invokeExact` 方法包含在库依赖项中，您的模块也可能不使用这些方法。因此，如需在指定了 `minSdkVersion 25` 或更低版本的情况下继续使用该库，请启用代码缩减 (`/studio/build/shrink-code#shrink-code`) 来移除未使用的方法。如果这样做不起作用，可考虑使用一个替代库，让其不使用不受支持的方法。

可通过 Android Gradle 插件 3.0.0 实现的 Java 8 及更高版本语言功能脱糖不会使任何额外的类和 API（如 `java.util.stream.*`）可在较低的 Android 版本中使用。对部分 Java API 脱糖的支持可通过 Android Gradle 插件 4.0.0 或更高版本实现，下一部分对此进行了介绍。

Java 8 及更高版本 API 脱糖支持（Android Gradle 插件 4.0.0 及更高版本）

如果您使用 Android Gradle 插件 4.0.0 或更高版本构建应用，插件扩展了对使用多种 Java 8 语言 API 的支持，而无需为应用设置最低 API 级别。

之所以能够实现对较低平台版本的这种额外支持，是因为脱糖引擎经过插件 4.0.0 及更高版本扩展后，也能使 Java 语言 API 脱糖。因此，您可以在支持较低 Android 版本的应用中添加过去仅在最新 Android 版本中可用的标准语言 API（如 `java.util.streams`）。

使用 Android Gradle 插件 4.0.0 或更高版本构建应用时，支持下面一组 API：

- 顺序流 (`java.util.stream`)
- `java.time` 的子集
- `java.util.function`
- `java.util.{Map,Collection,Comparator}` 的最近新增内容
- 可选内容 (`java.util.Optional`、`java.util.OptionalInt` 和 `java.util.OptionalDouble`) 以及对上述 API 很有用的一些其他新类

- `java.util.concurrent.atomic` 的一些新增内容 (`AtomicInteger`、`AtomicLong` 和 `AtomicReference` 的新方法)
- `ConcurrentHashMap` (包含 Android 5.0 的问题修复)

如需查看受支持的 API 的完整列表，请参阅[通过脱糖获得 Java 8 及更高版本 API](/studio/write/java8-support-table) (/studio/write/java8-support-table)。

为了支持这些语言 API，插件编译了一个单独的 DEX 文件（其中包含缺失 API 的实现），并将其添加到您的应用中。脱糖过程会重新编写应用的代码，以便在运行时改用此库。

如需在任意版本的 Android 平台上启用对这些语言 API 的支持，请将 [Android 插件更新](/studio/releases/gradle-plugin#updating-plugin) (/studio/releases/gradle-plugin#updating-plugin) 为 **4.0.0**（或更高版本），并在**应用模块**的 `build.gradle` 文件中添加以下代码：

```
GroovyKotlin (#kotlin)
(#groovy)

android {
    defaultConfig {
        // Required when setting minSdkVersion to 20 or lower
        multiDexEnabled true
    }

    compileOptions {
        // Flag to enable support for the new language APIs
        coreLibraryDesugaringEnabled true
        // Sets Java compatibility to Java 8
        sourceCompatibility JavaVersion.VERSION_1_8
        targetCompatibility JavaVersion.VERSION_1_8
    }
}

dependencies {
    coreLibraryDesugaring 'com.android.tools:desugar_jdk_libs:1.1.5'
}
```

请注意，在以下情况下，您可能还需要在相应**库模块**的 `build.gradle` 文件中添加以上代码段：

- 库模块的插桩测试会使用这些语言 API（直接使用，或者通过库模块或其依赖项使用）。这是为了向您的插桩测试 APK 提供缺失的 API。
- 您想单独在该库模块上运行 lint。这是为了帮助 lint 识别出对语言 API 的有效使用，并避免错误地发出警告。

另请注意，API 脱糖可以与缩减功能结合使用，但仅限在使用 R8 缩减器时这样处理。

Content and code samples on this page are subject to the licenses described in the [Content License](/license) (/license). Java is a registered trademark of Oracle and/or its affiliates.

Last updated 2021-06-14 UTC.