# **Android混淆工具——Proguard实践**

### Proguard****介绍****

[官网](http://proguard.sourceforge.net" \t "_blank)的介绍是：ProGuard是一个免费的Java类文件缩小，优化，混淆和预验证的工具。它检测和删除未使用的类，字段，方法和属性；优化字节码并删除未使用的指令；它使用短的无意义的名称重命名剩余的类，字段和方法。所得到的应用程序和库更小，更快，并且更好地针对逆向工程进行优化。

而且Proguard已经集成在Android studio构建系统里了，可以通过简单的代码来实现构建apk的时候进行混淆打包。

### Proguard****使用****

首先，我们需要在项目里的build.gradle文件里配置Proguard。

buildTypes {

release {

shrinkResources true

minifyEnabled true

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

}

debug {

shrinkResources true

minifyEnabled true

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

}

}

可以看见，分别在release和debug两种构建类型配置了Proguard，在debug配置是因为你可以直接编译测试你混淆的效果，是否有影响正常功能的使用。

shrinkResources 是去除无效的资源文件，压缩资源。

minifyEnabled 是开启混淆。

这是默认的Proguard配置，proguard-rules.pro是需要在你的项目里创建的文件，层级跟build.gradle文件一样的，当然你可以随意改他的文件名，只不过需要在配置代码里面跟着修改，其实这个文件就是Proguard的自定义配置文件，**没有这个文件你构建是会报错的**。

proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'

### Proguard****基本语法****

-keep 不混淆类和成员或者被重命名

-keepnames 防止类和成员被重命名

-keepclassmembers 不混淆成员被移除或者被重命名

-keepnames 防止成员被重命名

-keepclasseswithmembers 不混淆拥有该成员的类和成员或者被重命名

-keepclasseswithmembernames 防止拥有该成员的类和成员被重命名

而且还支持通配符\*

例如

不混淆某个类

-keep public class com.shadow.example.abc { \*; }

不混淆某个包

-keep public class com.shadow.example.\*\* { \*; }

不混淆某个类的子类

-keep public class \* extends com.shadow.exampl.abc { \*; }

了解这些基本语法后，我们就来看一下自定义配置，没有自定义混淆配置的话会构建出错的。

#### Proguard****自定义配置****

接下来介绍一下自定义混淆配置

以下是我在项目里使用的配置，我会在注释里说明。

//--- 基础混淆配置 ---

-optimizationpasses 5 //指定代码的压缩级别

-allowaccessmodification //优化时允许访问并修改有修饰符的类和类的成员

-dontusemixedcaseclassnames //不使用大小写混合

-dontskipnonpubliclibraryclasses //指定不去忽略非公共库的类

-verbose //混淆时是否记录日志

-ignorewarnings //忽略警告，避免打包时某些警告出现，没有这个的话，构建报错

-optimizations !code/simplification/arithmetic,!code/simplification/cast,!field/\*,!class/merging/\* //混淆时所采用的算法

-keepattributes \*Annotation\* //不混淆注解相关

-keepclasseswithmembernames class \* { //保持 native 方法不被混淆

native <methods>;

}

-keepclassmembers enum \* { //保持枚举 enum 类不被混淆

public static \*\*[] values();

public static \*\* valueOf(java.lang.String);

}

//不混淆Parcelable

-keep class \* implements android.os.Parcelable {

public static final android.os.Parcelable$Creator \*;

}

//不混淆Serializable

-keep class \* implements java.io.Serializable {\*;}

-keepnames class \* implements java.io.Serializable

-keepclassmembers class \* implements java.io.Serializable {\*;}

-keepclassmembers class \*\*.R$\* { //不混淆R文件

public static <fields>;

}

//不做预校验，preverify是proguard的四个步骤之一，Android不需要preverify，去掉这一步能够加快混淆速度。

-dontpreverify

-keepattributes Signature //过滤泛型 出现类型转换错误时，启用这个

//--- 不能被混淆的基类 ---

-keep public class \* extends android.app.Activity

-keep public class \* extends android.app.Fragment

-keep public class \* extends android.app.Application

-keep public class \* extends android.app.Service

-keep public class \* extends android.content.BroadcastReceiver

-keep public class \* extends android.content.ContentProvider

-keep public class \* extends android.app.backup.BackupAgentHelper

-keep public class \* extends android.preference.Preference

-keep public class \* extends android.view.View

-keep class org.xmlpull.v1.\*\* { \*; }

//--- 不混淆android-support-v4包 ---

-dontwarn android.support.v4.\*\*

-keep class android.support.v4.\*\* { \*; }

-keep interface android.support.v4.app.\*\* { \*; }

-keep class \* extends android.support.v4.\*\* { \*; }

-keep public class \* extends android.support.v4.\*\*

-keep public class \* extends android.support.v4.widget

-keep class \* extends android.support.v4.app.\*\* {\*;}

-keep class \* extends android.support.v4.view.\*\* {\*;}

-keep public class \* extends android.support.v4.app.Fragment

//不混淆继承的support类

-keep public class \* extends android.support.v4.\*\*

-keep public class \* extends android.support.v7.\*\*

-keep public class \* extends android.support.annotation.\*\*

//不混淆log

-assumenosideeffects class android.util.Log {

public static boolean isLoggable(java.lang.String, int);

public static int v(...);

public static int i(...);

public static int w(...);

public static int d(...);

public static int e(...);

}

//保持Activity中参数类型为View的所有方法

-keepclassmembers class \* extends android.app.Activity {

public void \*(android.view.View);

}

//--- 不混淆第三方库 这个可以去相关的第三方库官网找寻混淆代码 如果被混淆了会无法使用 ---

//Gson

-keepattributes \*Annotation\*

-keep class sun.misc.Unsafe { \*; }

-keep class com.idea.fifaalarmclock.entity.\*\*\*

-keep class com.google.gson.stream.\*\* { \*; }

-keep class com.你的bean.\*\* { \*; }

//OkHttp3

-dontwarn okhttp3.logging.\*\*

-keep class okhttp3.internal.\*\*{\*;}

-dontwarn okio.\*\*

//Retrofit

-dontwarn retrofit2.\*\*

-keep class retrofit2.\*\* { \*; }

-keepattributes Signature

-keepattributes Exceptions

//RxJava RxAndroid

-dontwarn sun.misc.\*\*

-keepclassmembers class rx.internal.util.unsafe.\*ArrayQueue\*Field\* {

long producerIndex;

long consumerIndex;

}

-keepclassmembers class rx.internal.util.unsafe.BaseLinkedQueueProducerNodeRef {

rx.internal.util.atomic.LinkedQueueNode producerNode;

}

-keepclassmembers class rx.internal.util.unsafe.BaseLinkedQueueConsumerNodeRef {

rx.internal.util.atomic.LinkedQueueNode consumerNode;

}

//微信

-keep class com.tencent.mm.\*\* {\*;}

//Glide图片库

-keep class com.bumptech.glide.\*\*{\*;}

//友盟

-keepclassmembers class \* {

public <init> (org.json.JSONObject);

}

-keep class com.umeng.onlineconfig.OnlineConfigAgent {

public <fields>;

public <methods>;

}

-keep class com.umeng.onlineconfig.OnlineConfigLog {

public <fields>;

public <methods>;

}

-keep interface com.umeng.onlineconfig.UmengOnlineConfigureListener {

public <methods>;

}

//Testin

-dontwarn com.testin.agent.\*\*

-keep class com.testin.agent.\*\* {\*;}

//--- 一些特殊的混淆配置 ---

//有用到WEBView的JS调用接口不被混淆

-keepclassmembers class fqcn.of.javascript.interface.for.webview {

public \*;

}

//对于带有回调函数的onXXEvent、\*\*On\*Listener的，不能被混淆

-keepclassmembers class \* {

void \*(\*\*On\*Event);

void \*(\*\*On\*Listener);

}

//抛出异常时保留代码行号 方便测试

-keepattributes SourceFile,LineNumberTable

//不混淆我们自定义控件（继承自View）

-keep public class \* extends android.view.View{

\*\*\* get\*();

void set\*(\*\*\*);

public <init>(android.content.Context);

public <init>(android.content.Context, android.util.AttributeSet);

public <init>(android.content.Context, android.util.AttributeSet, int);

}