五步搞定

1

打开项目中的proguard-rules.pro，将下面的代码复制进去。我的思路是将混淆分为2个主要部分

定制化区域。这里边的内容是我们主要需要补充的部分，大致分为4个小部分，我们接下来的步骤主要是补充这4个部分的东西。

基本不用动区域。顾名思义，基本不用动。这块的内容我已经写好了，除非有特殊的需求，基本不用改动。。。请叫我雷锋。。。。。

#-------------------------------------------定制化区域----------------------------------------------

#---------------------------------1.实体类---------------------------------

#-------------------------------------------------------------------------

#---------------------------------2.第三方包-------------------------------

#-------------------------------------------------------------------------

#---------------------------------3.与js互相调用的类------------------------

#-------------------------------------------------------------------------

#---------------------------------4.反射相关的类和方法-----------------------

#----------------------------------------------------------------------------

#---------------------------------------------------------------------------------------------------

#-------------------------------------------基本不用动区域--------------------------------------------

#---------------------------------基本指令区----------------------------------

-optimizationpasses 5

-dontusemixedcaseclassnames

-dontskipnonpubliclibraryclasses

-dontskipnonpubliclibraryclassmembers

-dontpreverify

-verbose

-printmapping proguardMapping.txt

-optimizations !code/simplification/cast,!field/\*,!class/merging/\*

-keepattributes \*Annotation\*,InnerClasses

-keepattributes Signature

-keepattributes SourceFile,LineNumberTable

#----------------------------------------------------------------------------

#---------------------------------默认保留区---------------------------------

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

-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 public class com.android.vending.licensing.ILicensingService

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

-keepclasseswithmembernames class \* {

native <methods>;

}

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

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

}

-keepclassmembers enum \* {

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

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

}

-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);

}

-keepclasseswithmembers class \* {

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

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

}

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

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

}

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

static final long serialVersionUID;

private static final java.io.ObjectStreamField[] serialPersistentFields;

private void writeObject(java.io.ObjectOutputStream);

private void readObject(java.io.ObjectInputStream);

java.lang.Object writeReplace();

java.lang.Object readResolve();

}

-keep class \*\*.R$\* {

\*;

}

-keepclassmembers class \* {

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

}

#----------------------------------------------------------------------------

#---------------------------------webview------------------------------------

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

public \*;

}

-keepclassmembers class \* extends android.webkit.WebViewClient {

public void \*(android.webkit.WebView, java.lang.String, android.graphics.Bitmap);

public boolean \*(android.webkit.WebView, java.lang.String);

}

-keepclassmembers class \* extends android.webkit.WebViewClient {

public void \*(android.webkit.WebView, jav.lang.String);

}

#----------------------------------------------------------------------------

#---------------------------------------------------------------------------------------------------

2

理解完战略级思想后，我们开始第一部分补充-实体类。实体类由于涉及到与服务端的交互，各种gson的交互如此等等，是要保留的。将你项目中实体类都拎出来，用以下语法进行保留。

-keep class 你的实体类所在的包.\*\* { \*; }

如我的项目下类User的完整路径为：com.demo.login.bean.User, 那我的混淆如下

#---------------------------------1.实体类---------------------------------

-keep class com.demo.login.bean.\*\* { \*; }

#-------------------------------------------------------------------------

当然你的实体类肯定不止这一个，请用上边的方式一一添加，如果你的实体类都在一个包下，那你就幸福了。

3

第2部分是第三方包。打开你的build.gradle文件，查看你用了哪些第三方的包。

dependencies {

compile 'com.github.bumptech.glide:glide:3.7.0'

compile 'org.greenrobot:eventbus:3.0.0'

}

我这里用了glide，eventbus。我去他们的官网把已经写好的混淆copy下来。

#---------------------------------2.第三方包-------------------------------

#eventBus

-keepattributes \*Annotation\*

-keepclassmembers class \*\* {

@org.greenrobot.eventbus.Subscribe <methods>;

}

-keep enum org.greenrobot.eventbus.ThreadMode { \*; }

-keepclassmembers class \* extends org.greenrobot.eventbus.util.ThrowableFailureEvent {

<init>(java.lang.Throwable);

}

#glide

-keep public class \* implements com.bumptech.glide.module.GlideModule

-keep public enum com.bumptech.glide.load.resource.bitmap.ImageHeaderParser$\*\* {

\*\*[] $VALUES;

public \*;

}

#-------------------------------------------------------------------------

一般官网都是有混淆的，没有的话就google，也没有的话自己按照上面的写法自己写，还不会的话。。。。。只能换个包。。。。。如果你是直接包含的jar包的话，你这样写

#---------------------------------2.第三方包-------------------------------

#log4j

-libraryjars log4j-1.2.17.jar

-dontwarn org.apache.log4j.\*\*

-keep class org.apache.log4j.\*\* { \*;}

#-------------------------------------------------------------------------

大致意思就是不混淆，不报warn。如果gradle报错的话，可以考虑注释掉-libraryjars log4j-1.2.17.jar这句。

4

第三部分与js互调的类，工程中没有直接跳过。一般你可以这样写

-keep class 你的类所在的包.\*\* { \*; }

如果是内部类的话，你可以这样

-keepclasseswithmembers class 你的类所在的包.父类$子类 { <methods>; }

例如

-keepclasseswithmembers class com.demo.login.bean.ui.MainActivity$JSInterface {

<methods>;

}

5

第四部分与反射有关的类，工程中没有直接跳过。类的话直接这样

-keep class 你的类所在的包.\*\* { \*; }

熟练后不用五分钟就能搞定。大家可以把自己写完混淆所用的时间写在评论里（0，0）

深入了解

先逐条介绍下基本指令区指令的含义

-optimizationpasses 5

代码混淆的压缩比例，值在0-7之间

-dontusemixedcaseclassnames

混淆后类名都为小写

-dontskipnonpubliclibraryclasses

指定不去忽略非公共的库的类

-dontskipnonpubliclibraryclassmembers

指定不去忽略非公共的库的类的成员

-dontpreverify

不做预校验的操作

-verbose

-printmapping proguardMapping.txt

生成原类名和混淆后的类名的映射文件

-optimizations !code/simplification/cast,!field/\*,!class/merging/\*

指定混淆是采用的算法

-keepattributes \*Annotation\*,InnerClasses

不混淆Annotation

-keepattributes Signature

不混淆泛型

-keepattributes SourceFile,LineNumberTable

抛出异常时保留代码行号

基本指令区基本介绍完了，说下2条用的最多的指令

-keep class XXXX

保留类名不变，也就是类名不混淆，而类中的成员名不保证。当然也可以是继承XXX类的所有类名不混淆，具体代码不贴了，重在理解。

-keepclasseswithmembers class XXXX

保留类名和成员名。当然也可以是类中特定方法，代码不贴了，理由同上。

作者：wolearn

链接：http://www.jianshu.com/p/f3455ecaa56e

來源：简书

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。