

Bugly (Android) SDK 快速接入向导

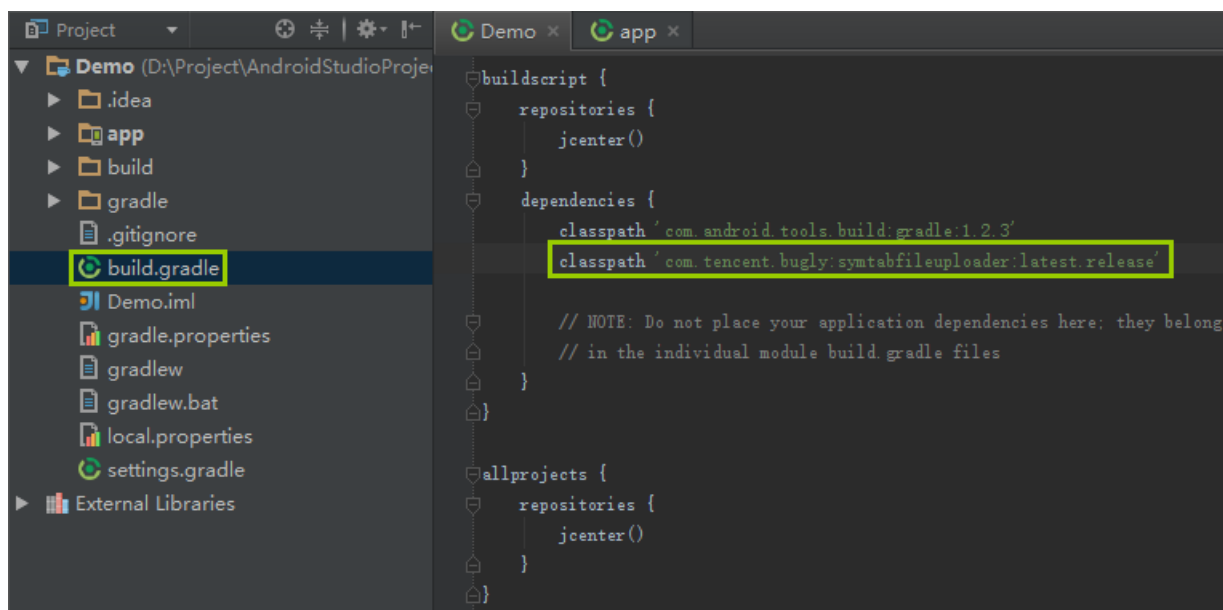
一、库文件导入

方式1. 自动导入 (推荐)

如果您使用Gradle编译Apk，我们强烈推荐您使用自动接入方式配置库文件Bugly插件 ([JCenter仓库](#))。

在项目的build.gradle文件的dependencies (buildscript部分) 中添加Bugly插件：

```
classpath 'com.tencent.bugly:symtabfileuploader:latest.release' //latest.release代表最新版本
```



在Module的build.gradle文件中添加依赖和属性配置：

```

apply plugin: 'bugly'    //添加Bugly符号表插件

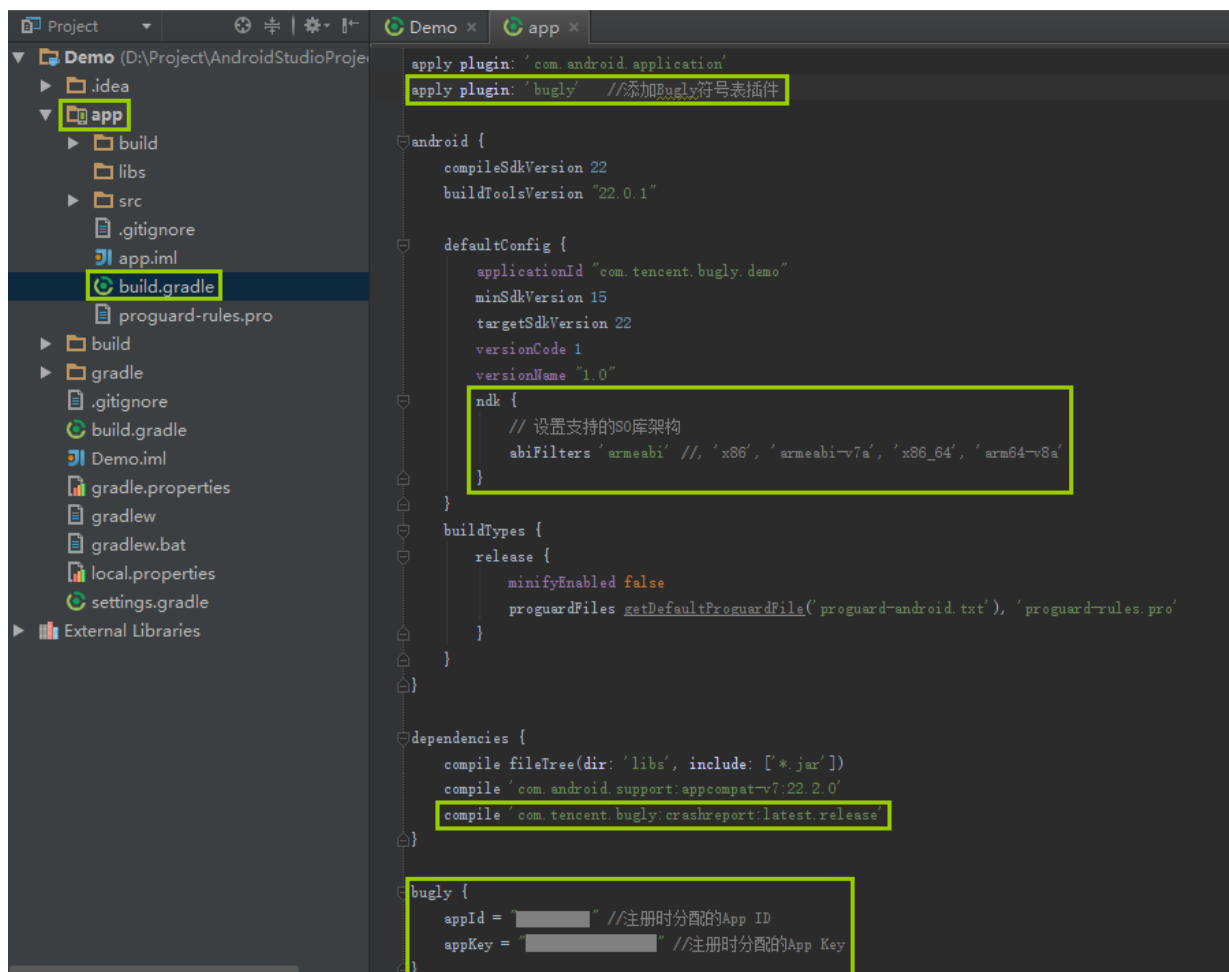
android {
    defaultConfig {
        ndk {
            // 设置支持的SO库架构
            abiFilters 'armeabi' //, 'x86', 'armeabi-v7a', 'x86_64', 'arm64-v8a'
        }
    }
}

dependencies {
    compile 'com.tencent.bugly:crashreport:latest.release'
}

bugly {
    appId = "your AppId"    //注册时分配的App ID
    appKey = "your AppKey"  //注册时分配的App Key
}

```

注意：自动集成包含了Bugly SO库，建议在Module的build.gradle文件中使用NDK的abiFilter配置，设置支持的SO库架构。



至此，Bugly SDK和插件已经集成完成了。后续更新Bugly时，只需修改配置脚本中的版本号即可。

方式2. 手动导入

如果您不采用上述自动导入方式，也可以手动接入Bugly。

下载Bugly

请先下载Bugly的**SDK包**。

如果您的工程有Native代码（C/C++），请继续下载**NDK动态库**和**符号表工具**（纯Java工程可跳过）。

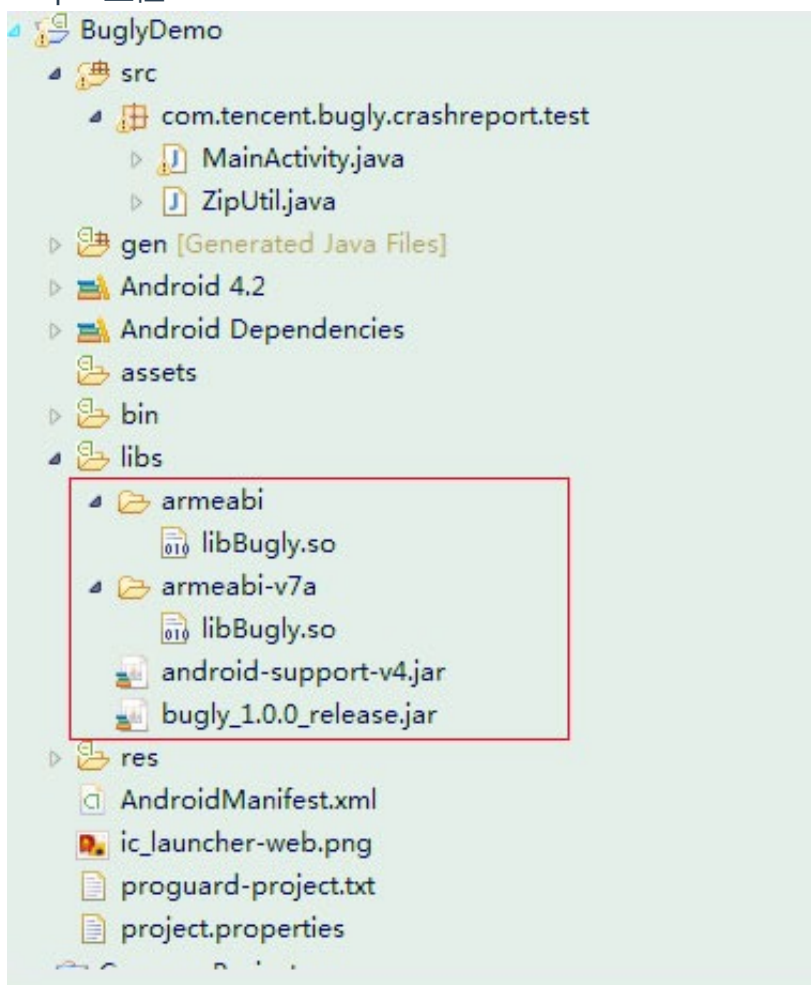
下载地址：<http://bugly.qq.com/whitebook>

导入文件

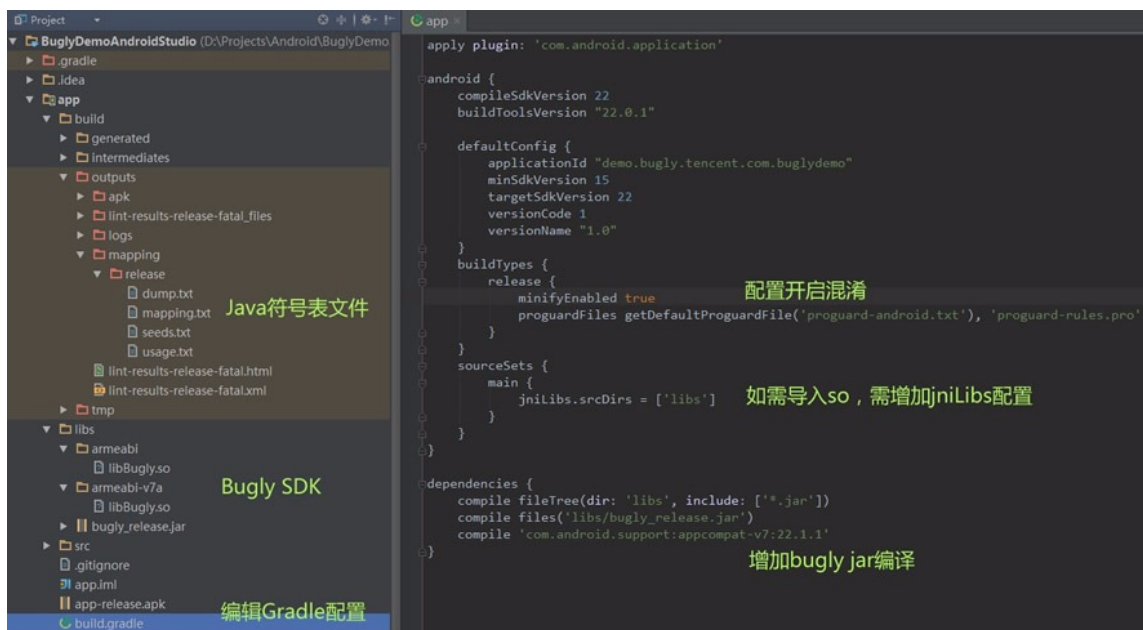
将文件复制到工程的libs目录下，并添加到依赖库中。

如果您的Android工程是纯Java代码，只需要导入Jar文件。如果含有Native代码，需一并导入so文件。

- Eclipse工程



- AndroidStudio工程



为了使您的堆栈可读性更高，对于纯Java代码项目，只需在版本管理（设置）的对应版本中上传混淆后生成的Mapping文件即可；对于含有Native代码的工程，除了Mapping文件，还需要通过符号表工具手动提Debug SO的Symbol符号表文件。具体方法请参考：《符号表配置（Android）》

二、参数配置

在AndroidManifest.xml中添加权限：

```
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_LOGS" />
```

避免混淆Bugly，请在Proguard混淆文件中配置：

```
-keep public class com.tencent.bugly.**{*;} 
```

三、代码调用

Bugly会为您检测使用环境并自动完成配置。请将以下CrashReport代码复制到项目Application类onCreate()中：

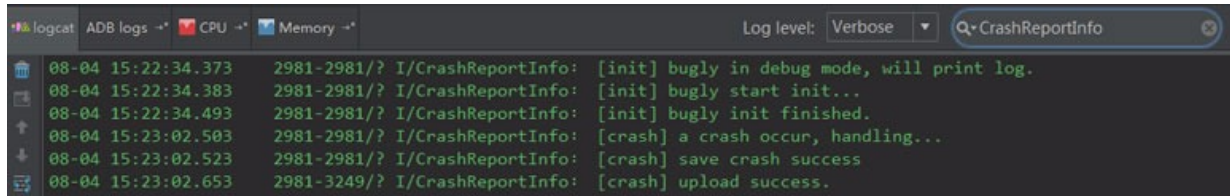
```
CrashReport.initCrashReport(appContext, "注册时申请的APPID", false);
```

四、测试

现在您可以制造一个Crash，来体验Bugly的能力了。在Bugly的初始化代码后，调用

```
CrashReport.testJavaCrash();
```

执行到这段代码时会发生一个Crash，此时Logcat的TAG=CrashReportInfo中输出为：



现在您已经可以在网页上看到刚才的异常详情了（延迟一般在10s以内）。