**一、java.lang.NoSuchMethodError基本介绍**

|  |  |
| --- | --- |
| 全名 | java.lang.NoSuchMethodError |
| 官方解释 | Thrown if an application tries to call a specified method of a class (either static or instance), and that class no longer has a definition of that method.  意思就是程序找不到你要用的那个方法！ |
| 影响力排名 | 出错量排名第4 |
| 精神哥点评 | 抛出这异常，只能怪这个喜新厌旧、鱼龙混杂的社会！虽然是社会的错，但你不想办法解决它，就只能等屎了！ |

java.lang.NoSuchMethodError问题的根本原因是**开发编译时所依赖的环境，跟实际App运行的环境不匹配。**

**二、错误场景分析**

**1、你就喜欢搞新API，这么喜新厌旧！你家妞知道不？**

|  |  |
| --- | --- |
| 错误代码 | //检查该acitivity是否已经销毁  public static boolean isActivityClose( Activity activity){  if(activity != null){  returnactivity.isDestroyed();  }  return false ;  } |
| 编译环境 | [**Android**](http://lib.csdn.net/base/15) 4.2 SDK（ API LEVEL17） |
| 运行环境 | Android 4.0 设备（ API LEVEL14） |
| 期望结果 | 正确返回activity是否已经销毁。 |
| 实际结果 | Crash！！  java.lang.NoSuchMethodError：com.tencent.bugly.crashreport.test.MainActivity.isDestroyed |
| 原因分析 | Activity.isDestroyed()方法是Android 4.2增加的，在这之前的系统肯定找不到这个方法，所以在4.2之前的设备上都会Crash！ |
| 解决方法 | 调用时对当前系统API LEVEL做判断，如果小于17就放弃调用 |
| 修复代码 | public Boolean isActivityClose( Activity activity){  if(Build.VERSION.SDK\_INT < 17)  return null;  if(activity != null){  returnactivity.isDestroyed();  }  return false ;  } |

**2、@Deprecated的API厂家最爱删，你竟然敢用？**

|  |  |
| --- | --- |
| API标志 | @Deprecated标注的API |
| 标志描述 | Annotation type used to mark program elements that should no longer be used by programmers.  这个API后面随时会被干掉！ |
| 现状描述 | 谷歌还是比较小心的，精神哥发现Android的SDK及Android的亲儿子设备上，这些@Deprecated的API基本上都保留了，谷歌最多就是置空实现或直接抛出一个非法访问的异常，所以开发过程中并没有感觉到API过期不能用（当然有警告啦）  但谷歌亲儿子设备只占Android设备的冰山一角，很多厂家可没管那么多，为了尽可能的节省资源，大刀阔斧的对API进行裁剪，而这个@Deprecated的API就是最有理由被干掉的，所以出现大量的java.lang.NoSuchMethodError的Crash！ |

**3、@Hide的API是人家谷歌私有的，删改都不会跟你商量的，你还敢用吗？**

|  |  |
| --- | --- |
| API标志 | @Hide标注的API |
| 标志描述 | When applied to a package, class, method or field, @hide removes that node and all of its children from the documentation.  这个API压根不想让你看到，更别说让你使用了。 |
| 现状描述 | 由于Android的开源，加上Java的反射机制的便利，开发者总是喜欢研究源码，用注入或反射的方式获取到官方未正式开放的能力。这很有效，但也很危险，因为谷歌随时会调整，会导致你的App出现各种诡异的java.lang.NoSuchMethodError！  很多同学认为public的私有API，谷歌不会乱改，可以大胆的用。  精神哥，随手给你挑一个反例来证明你的天真：  android.content.pm.PackageParser（该类用于apk安装包内容解析，很受欢迎）这个私有类的public构造函数在5.0发生变化了，而之前一直没变过。  //5.0以前有String作为参数，用于传入apk路径  public PackageParser(String archiveSourcePath) {  mArchiveSourcePath = archiveSourcePath;  }  //5.0优化成无参实现了，一个对象可以多次解析了  public PackageParser() {  mMetrics = new DisplayMetrics();  mMetrics.setToDefaults();  }  想看PackageParser.java源码又懒得找的同学，公众号里@精神哥，我给你发哈！ |