

在 Android 11 及更高版本系统中处理可空性



谷歌开发者

已认证的官方帐号

已关注

10 人赞同了该文章

作者 / David Winer, Kotlin 产品经理



在去年 5 月的 I/O 开发者大会上，我们正式宣布 Kotlin 优先 (Kotlin First) 的这一重要理念，Kotlin 将成为 Android 开发者的首选语言。目前，在排名前 1,000 位的 Android 应用中，已有超过 60% 正在使用 Kotlin 进行开发。为什么 Kotlin 受到这么多开发者的喜爱呢？这里就不得不提 Kotlin 在可空性方面的优势了。Kotlin 将可空性直接融合到了类型系统中，这意味着开发者在声明一个参数时，需要提前说明该参数能否接纳 null 值。本文将带您了解 Android 11 SDK 引入了哪些变更，以便在 API 中显示更多的可空性信息。此外，我们还将介绍一些实用方法与技巧，帮助您做好准备，顺利应对 Kotlin 中的可空性问题。

浅谈 Kotlin 中的可空性

使用 Kotlin 编写代码时，您可以使用问号操作符来指明可空性：

KOTLIN

```
var x: Int = 1
x = null // compilation error

var y: Int? = 1
y = null // okay
```

Kotlin 的这个特性能够让您的代码更安全。即使您随后调用一个方法或试图访问 x 等非空变量属性，也不会面临空指针异常的风险。许多开发者向我们反馈表示，自从有了该特性后，他们便可以

如果我使用的 (Android) API 不是用 Kotlin 编写的, 该怎么办? 不用担心, Kotlin 编译器能够识别 Java 代码的注释, 从而判定方法返回的结果是否为可空值, 例如:

JAVA

```
public @Nullable String getCurrentName() {  
    return currentName;  
}
```

添加 @Nullable 注释后, 当您在 Kotlin 文件中使用 getCurrentName 的结果时, 您必须先进行 null 值检查, 才能对其解析引用, 否则会触发 Android Studio 报错, 而且 Kotlin 编译器也会在构建时抛出错误。@NonNull 注释则恰好与之相反, 如果一个方法标有 @NonNull 注释, Kotlin 编译器会将该方法返回的结果视为非空类型, 并禁止您在之后的代码中把该结果设置为 null。

此外, Kotlin 编译器还可以识别另外两个类似的注释: @RecentlyNullable 和 @RecentlyNonNull。这两个注释与 @Nullable 和 @NonNull 完全相同, 唯一的区别在于它们会生成警告而非错误*。

- 鉴于 Kotlin 注释处理的相关规则, 目前在少数情况下, 编译器仅对 @Nullable 引用报错, 而对 @RecentlyNullable 引用不报错。更多详情: youtrack.jetbrains.com/...

Android 11 中的可空性

我们在近期发布了 [Android 11 开发者预览版](#), 邀请开发者们试用最新的 Android 11 SDK。我们升级了 SDK 中的部分注释, 将 @RecentlyNullable 和 @RecentlyNonNull 分别更改为 @Nullable 和 @NonNull (可空性违规行为从触发警告变为导致错误)。此外, 我们也继续为 SDK 中不包含可空性信息的方法增加 @RecentlyNullable 和 @RecentlyNonNull 注释。

下一步

如果您正在使用 Kotlin 编写代码, 当您从 Android 10 升级至 Android 11 SDK 后, 您可能会遇到一些新的编译器警告, 另外, 之前的警告也可能以错误形式出现。这是我们特意引入的变更, 也是 Kotlin 编译器的特性之一。这些警告会提醒您当前代码可能会导致应用崩溃 (如果您编写的不是 Kotlin 代码, 您可能会完全忽略这个风险)。您可以通过在代码中添加[空值检查](#)来解决这些警告或报错。

我们将继续遵循以下原则为 Android SDK 添加注释: 先在较低版本 (例如 Android 10) 中采用 @RecentlyNullable 和 @RecentlyNonNull, 然后再在新版本 (例如 Android 11) 中将其升级为 @Nullable 和 @NonNull。此举的目的是为了给您预留至少一个版本发布周期的时间更新 Kotlin 代码, 确保其质量更高、更加健壮。

发布于 04-16

[Android](#) [Android 11](#) [Kotlin](#)

推荐阅读



作为一个程序员怎么通过 indroid开发赚钱

室内设计二... 发表于室内设计

一种绕过Android P上非SDK接口限制的简单方法

众所周知，Android P 引入了 针对非 SDK 接口（俗称为隐藏API）的使用限制。这是继 Android N上 针对 NDK 中私有库的链接限制 之后的又一次重大调整。从今以后，不论是native层的NDK还是 Ja...

weishu



除了Android开发Kotlin 还能做什么？六款优质Kotlin项目分享

Gitee 发表于Gitee...

还没有评论

写下你的评论...

