# **[Android学习探索之Java 8 在Android 开发中的应用](http://www.cnblogs.com/whoislcj/p/6121832.html)**

### **前言：**

  Java 8推出已经将近2年多了，引入很多革命性变化，加入了函数式编程的特征，使基于行为的编程成为可能，同时减化了各种设计模式的实现方式，是Java有史以来最重要的更新。但是Android上，一直没有看到支持Java8的消息。Android到底会不会支持Java8呢？答案是肯定的，Android N已经开始支持Java 8 了。

### **关于Java 8**

#### **（1.）使用Lambda表达式**

   Java 8的一大亮点是引入Lambda表达式，使用它设计的代码会更加简洁。当开发者在编写Lambda表达式时，也会随之被编译成一个函数式接口。

#### **（2.）引入函数式接口**

   Java 8 引入的一个核心概念是函数式接口。如果一个接口定义个唯一一个抽象方法，那么这个接口就成为函数式接口。

#### **（3.）泛型接口改进**

  这是一个以前不能做到的，对编译器判定泛型能力的努力改进。在以前版本的Java中有许多情形编译器不能给某个方法计算出泛型，当方法处于嵌套的或串联方法调用这样的上下文的时候，即使有时候对程序员来说它看起来“很明显”。

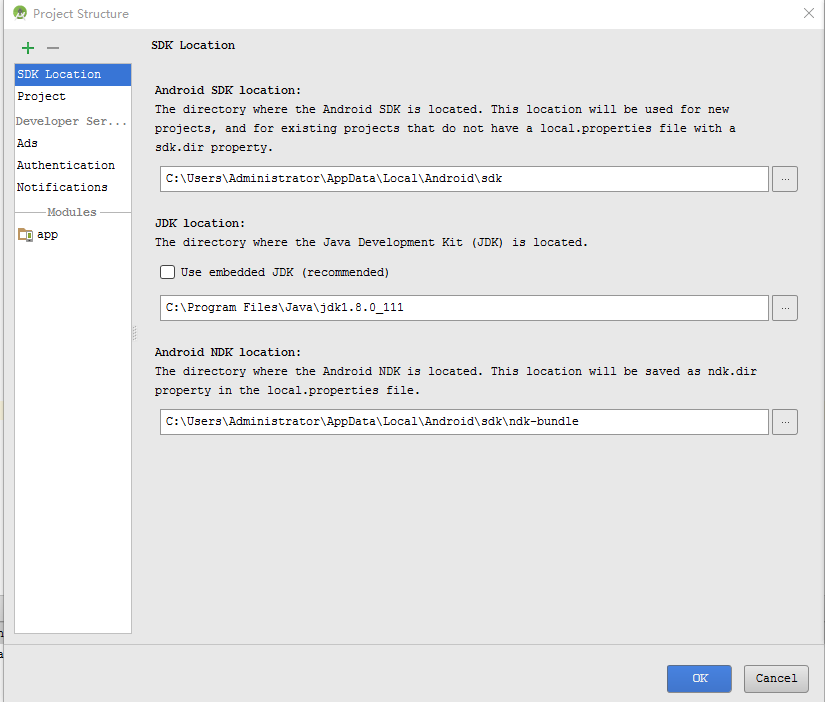
#### **（4.）Streams编程**

  集合（Collections）的改进也是Java 8的一大亮点，而让集合越来越好的核心组件则是“Stream”。它与java.io包里的InputStream和OutputStream是完全不同的概念，它是一个全新的概念，大家不要混淆。此外，Stream的出现也并不是要取代ArrayLists或其他集合，它提供了一种操作大数据接口，让数据操作更容易和更快。

解说：不过上面的几大改变仅仅只能在Android N及之后的版本才能使用，这也让Android 23及以下还在使用Java 7开发的小伙伴们望尘莫及，不过Java 8的lambda表达式还是可以支持的，接下来我们看下如何使用Java 8.

### **如何在Android 开发中使用Java 8？**

#### **1.）第一步，安装Java8，并指定项目使用Java8，minSdkVersion <24 只能使用Java8 的 Lambda特性,minSdkVersion>= 24可以使用Java8特性，由此可见全部普及Java8还是需要一定时间的**



#### **2.）第二步在module的build.gradle中添加相关配置**

android {

...

defaultConfig {

...

jackOptions {

enabled true

}

}

compileOptions {

sourceCompatibility JavaVersion.VERSION\_1\_8

targetCompatibility JavaVersion.VERSION\_1\_8

}

}

我这里使用的Android studio 2.2.2版本 Android版本为24，如果是低于24版本的话是无法使用Java 8特性的。如果仅仅只想使用lambda表达式的话完全够用了，我们只需把minSdkVersion设为14即可。

以下是我自己的设置范例

android {

compileSdkVersion 24

buildToolsVersion '24.0.2'

defaultConfig {

applicationId "com.whoislcj.jsbridge"

minSdkVersion 14

targetSdkVersion 24

versionCode 1

versionName "1.0"

jackOptions {

enabled true

}

}

buildTypes {

release {

minifyEnabled false

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

}

}

dexOptions {

incremental true

}

compileOptions {

sourceCompatibility JavaVersion.VERSION\_1\_8

targetCompatibility JavaVersion.VERSION\_1\_8

}

productFlavors {

}

}

### **关于Lambda**

   “Lambda 表达式”(lambda expression)是一个匿名函数，Lambda表达式基于数学中的λ演算得名，直接对应于其中的lambda抽象(lambda abstraction)，是一个匿名函数，即没有函数名的函数。Lambda表达式可以表示闭包（注意和数学传统意义上的不同）。

### **如何使用Java 8 Lambda表达式？**

#### **1.）使用Lambda表达式示例**

这里以常见的Java匿名内部类为例

（1.）未使用lambda表达式之前

1 findViewById(R.id.to\_web).setOnClickListener(new View.OnClickListener() {2 @Override3 public void onClick(View v) {4 Toast.makeText(MainActivity.this, "终于等到你", Toast.LENGTH\_LONG).show();5 }6 });

（2.）使用lambda表达式

1 findViewById(R.id.to\_web).setOnClickListener((View v) -> {2 Toast.makeText(this, "终于等到你", Toast.LENGTH\_LONG).show();3 });

（3.）进一步简化

1 findViewById(R.id.to\_web).setOnClickListener(v -> Toast.makeText(this, "终于等到你", Toast.LENGTH\_LONG).show());

#### **2.）扩展知识**

  在Android studio 2.1 版本要想实现Lambda表达式的话可以采用第三方AS retrolambda插件来实现，github地址：https://github.com/evant/gradle-retrolambda，同样那种方式仅仅支持lambda表达式而不支持Java8特性。

### **目前遇见的问题**

 1.）中文乱码问题

这个问题我一直以为是编码问题，后来才发现是jackOptions {enabled true}引起的中文乱码，解决办法在project的gradle.properties中添加如下代码

org.gradle.jvmargs=-Dfile.encoding=UTF-8

 2.）Instant Run 目前不能用于 Jack，目前建议暂时关闭Instant Run使用。

### **总结：**

   本文仅仅是学习了如何使用在Android 开发中使用Java8的lambda表达式来使代码变得简洁，借来会逐步对Java的一些特性进行学习。