

快速了解 Kotlin

一：环境搭建

Android Studio 开发工具

方法/步骤

1. 打开 Android Studio, 在 Android Studio 左上角点击 File 选项, 选择 New, 接着点击 New Project 创建一个 Android 项目。

2. 一路 next 后, 我们把定制 Activity 这步中的两个选项都勾上, 最后点击右下角 Finish 创建好一个普通 Android 项目。

3. 然后在创建的项目中找到 MainActivity 这个文件, 打开这个文件, 点击 Android Studio 上面的 Code 选项, 在展开的面板中, 我们选择最后一项将 java 文件转换成 Kotlin 文件。IDE 提供的这个功能使得我们很容易将 java 文件转成 Kotlin 文件。

4. 完成转换后, 接下来, 在 IDE 右上角会出现下面的提示框 (如果没有出现请关闭重新打开一下就有了), 我们点击 configure。接着点击 Android with Gradle。

5. 接着会有一个相关配置的弹出框, 我们在弹出框中可以选择 kotlin 作用的范围已及使用 kotlin 的版本, 这个自己根据情况选择就好。然后点 Ok 就可以了。

6. 最后, 来到 build.gradle 文件, 我们会发现依赖中多了一项有关 kotlin 的配置。至此, kotlin 的环境已经搭建好了。可以愉快的进行 Kotlin 开发了。

二：引入开发 APP 需要的框架

1. 首先这里引用的是 AnKo 提供的 kotlin 框架，AnKo 提供了 App 所需的框架。

Anko Commons: a lightweight library full of helpers for intents, dialogs, logging and so on;

Anko Layouts: a fast and type-safe way to write dynamic Android layouts;

Anko SQLite: a query DSL and parser collection for Android SQLite;

Anko Coroutines: utilities based on the [kotlinx.coroutines](#) library.

图解：

Gradle-based project

Anko has a meta-dependency which plugs in all available features (including Commons, Layouts, SQLite) into your project at once:

```
dependencies {  
    compile "org.jetbrains.anko:anko:$anko_version"  
}
```

Make sure that you have the `$anko_version` settled in your gradle file at the project level:

```
ext.anko_version='0.10.3'
```

If you only need some of the features, you can reference any of Anko's parts:

```
dependencies {  
    // Anko Commons  
    compile "org.jetbrains.anko:anko-commons:$anko_version"  
  
    // Anko Layouts  
    compile "org.jetbrains.anko:anko-sdk25:$anko_version" // sdk15, sdk19, sdk21, sdk23 are also available  
    compile "org.jetbrains.anko:anko-appcompat-v7:$anko_version"  
  
    // Coroutine listeners for Anko Layouts  
    compile "org.jetbrains.anko:anko-sdk25-coroutines:$anko_version"  
    compile "org.jetbrains.anko:anko-appcompat-v7-coroutines:$anko_version"  
  
    // Anko SQLite  
    compile "org.jetbrains.anko:anko-sqlite:$anko_version"  
}
```

There are also a number of artifacts for the Android support libraries:

```
dependencies {
    // Appcompat-v7 (only Anko Commons)
    compile "org.jetbrains.anko:anko-appcompat-v7-commons:$anko_version"

    // Appcompat-v7 (Anko Layouts)
    compile "org.jetbrains.anko:anko-appcompat-v7:$anko_version"
    compile "org.jetbrains.anko:anko-coroutines:$anko_version"

    // CardView-v7
    compile "org.jetbrains.anko:anko-cardview-v7:$anko_version"

    // Design
    compile "org.jetbrains.anko:anko-design:$anko_version"
    compile "org.jetbrains.anko:anko-design-coroutines:$anko_version"

    // GridLayout-v7
    compile "org.jetbrains.anko:anko-gridlayout-v7:$anko_version"

    // Percent
    compile "org.jetbrains.anko:anko-percent:$anko_version"

    // RecyclerView-v7
    compile "org.jetbrains.anko:anko-recyclerview-v7:$anko_version"
    compile "org.jetbrains.anko:anko-recyclerview-v7-coroutines:$anko_version"

    // Support-v4 (only Anko Commons)
    compile "org.jetbrains.anko:anko-support-v4-commons:$anko_version"

    // Support-v4 (Anko Layouts)
    compile "org.jetbrains.anko:anko-support-v4:$anko_version"
}
```

此框架引入，可以实现纯代码布局，不习惯的同学们还是可以结合 android 布局方式来实现功能，这里只是推荐，可以根据个人的喜好。

2 Kotlin & JAVA

kotlin & java 混合开发实现功能

```
compile 'org.jetbrains.kotlin:kotlin-reflect:1.1.0'
// ...
apply plugin: 'kotlin-android-extensions'
import com.it.kotlins.master.utils.Utils
import kotlinx.android.synthetic.main.activity_main.*
class MainActivity : AppCompatActivity() {
```