## 3. Debug trực tiếp APK bằng Android Studio

**Harry**

Debug trực tiếp APK bằng Android Studio

## Xác định ứng dụng cần Debug

Chắc hẳn các bạn trong "nghề" ít nhất vài đôi lần phải tìm kiếm lỗ hổng trong ứng dụng Android vì đam mê hoặc vì lý do công việc. Nếu bạn là người phát triển ứng dụng, mã nguồn có trong tay và ý tưởng thì có trong đầu, dẫn tới việc đọc code tìm lỗi là điều quá đơn giản rồi nhưng với những người kiểm thử Blackbox thì sao? Trong tay thì có mỗi một file APK duy nhất!?

Họ phải áp dụng một vài phương pháp dịch ngược ứng dụng Android để có được mã nguồn, rồi thực hiện kiểm tra "bằng mắt" là một điều thực sự mệt mỏi, và đôi khi rất khó để hiểu các Work Flow của chương trình, Logic của một function hoặc ý đồ của lập trình viên.

Trong bài viết này, mình hướng dẫn các bạn sử dụng chính Android Studio và ApkStudio để thực hiện việc Debug trực tiếp ứng dụng APK. Thú thật dự định của bài viết ban đầu là không dùng Android Studio đâu, mình sử dụng một phương pháp khá nông dân. Nhưng sau khi tải lại Android Studio và làm vài vòng ngó nghiêng các tính năng của nó thì mình quyết định dẹp bỏ hết cái phương pháp cũ, và cách làm dưới đây thực sự là chân ái và nhẹ nhàng hơn rất nhiều.

OK, việc đầu tiên trong kiểm thử ứng dụng Android là bạn phải chạy được nó trong môi trường lý tưởng của bạn và bạn có thể kiểm soát được nó. Ví dụ trong bài viết này, mình sử dụng ứng dụng `FPT Techday 2019 - com.fpt.event.techday` để làm mẫu phân tích.

Nếu chưa có file APK bạn có thể thực hiện Download nó về thông qua chợ ứng dụng Play Store của Google, hoặc sử dụng các chợ không chính thống khác để download trực tiếp các file APK. Với chợ không chính thống, mình hay sử dụng `https://apkpure.com/`. Còn mình sẽ hướng dẫn các bạn lấy file APK từ việc cài đặt thông qua PlayStore.

À trước tiên có vài điều mà bạn cần phải làm trước khi đi tiếp nhé:

- Chế độ Developer được bật
- Trong `Developer Options` cần lưu ý các điều sau:
  - Stay Awake: Cái này nên bật để màn hình của bạn luôn sáng và không bị khóa trong quá trình Debug ứng dụng.

    - Android Debugging: Cái này phải bật để có thể giao tiếp được với máy tính thông qua `adb`
    - ADB Over Network: Cái này bật thì `adb-server` sẽ được khởi tạo và listen cổng `5555` qua TCP/IP. Cái này mình hay sử dụng trong trường hợp quên dây cáp USB ở nhà.
- Trong `Security` Settings cần lưu ý:
    - Unknow Sources: Cái này phải bật để bạn có thể cài file `apk` bằng ADB, một cách không chính thống.

Sau khi đảm bảo các điều trên, tiếp tục thực hiện vào PlayStore và tải và cài đặt một ứng dụng mà bạn thích để phân tích.

Trên máy tính mình thực hiện lệnh kết nối tới máy điện thoại thông qua `adb`

```
$ adb devices
```

hoặc kết nối thông qua `Over network`

```
$ adb connect 192.168.69.9"555
connected to 192.168.69.6:5555
```

Thực hiện tìm kiếm file apk của chương trình bạn vừa cài đặt

```
$ adb shell pm list packages -f| grep fpt
package:/data/app/com.fpt.event.techday-
1/base.apk=com.fpt.event.techday
package:/data/app/com.shopfptshop.apps-
1/base.apk=com.shopfptshop.apps
```

Ở đây chúng ta đã thấy ứng dụng `com.fpt.event.techday` được cài đặt, và đường dẫn file APK được tải về trên điện thoại là /data/app/com.fpt.event.techday-1/base.apk. Chúng ta tiếp tục sử dụng `adb pull` để download file này về máy tính, và sau đó đổi tên nó thành dạng `<tên-package>.apk` để tiện phân tích về sau.

```
$ adb pull /data/app/com.fpt.event.techday-1/base.apk
com.fpt.event.techday.apk
/data/app/com.fpt.event.techday-1/base.ap...ped. 0.8 MB/s
(10973287 bytes in 13.875s)
```

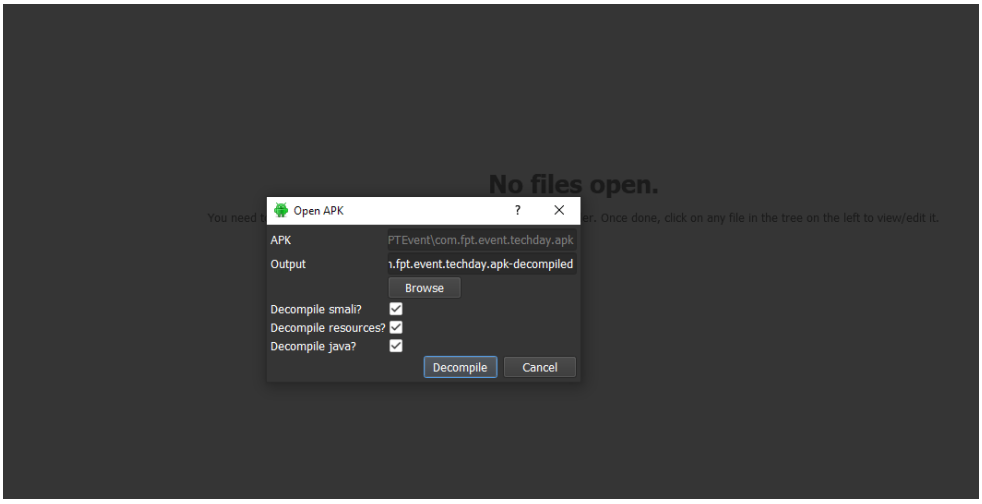
### Bật tính năng Debug trong AndroidManifest.xml

Để Debug được ứng dụng thì có một điều quan trọng chúng ta cần kiểm tra trong file `AndroidManifest.xml` có được thiết lập tham số `android:debuggable="true"` hay không, nếu không thì bạn cần phải Patch lại file này, sau đó sign và build lại một file APK của riêng ta. Nghe phức tạp phải không nào? Nhưng đừng bỏ cuộc nhé có một phần mềm `ApkStudio` nó đã làm hết điều này cho chúng ta. Một vài lưu ý:

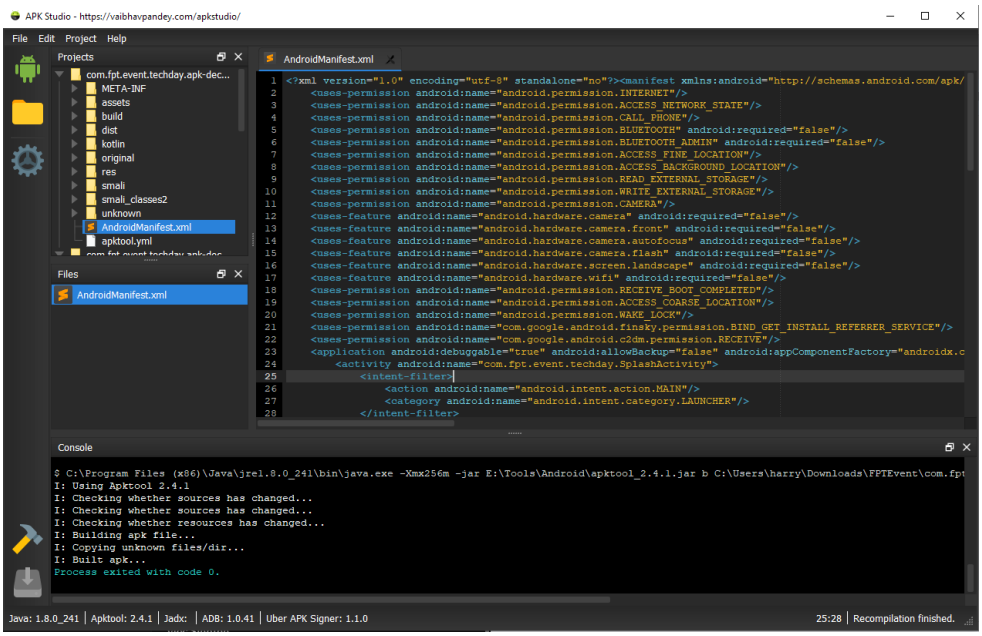
- Link tải https://github.com/vaibhavpandeyvpz/apkstudio
- APKStudio sẽ cần thêm một vài thư viện để có thể hoạt động được, bạn cần có:
  - Java - Tất nhiên rồi (JRE 8 nhé)
  - APKTool, Dùng để phân tích file APK, build APK https://github.com/iBotPeaches/Apktool/releases
  - Jadx, Dùng để chuyển Dex sang Java https://github.com/skylot/jadx/releases
  - ADB - Tất nhiên lúc này máy bạn đã được cài đặt rồi
  - Uber APK Sign, Dùng để Sign lại file APK đã patch https://github.com/patrickfav/uber-apk-signer/releases

Ok, tiến hành tìm hiểu thôi. Trong giao diện của ứng dụng APKStudio, chọn tới file APK của các bạn `File -> Open -> APK`, hoặc tổ hợp phím `Ctrl + N`. Sau đó tick chọn:

- Decompile smali
- Decompile resource



Công cụ này sẽ tiến hành Decompile toàn bộ các file resource, extract các mã smali của ứng dụng. Và dưới đây là kết quả của chúng ta, màn hình hiển thị nội dung của tệp tin AndroidManifest.xml




Chúng ta sẽ thấy nội dung trong cấu hình `<application>` không có tham số `android:debuggable="true"`, tức là mặc định khi release APK thì ứng dụng này đã tắt tính năng debug. Chúng ta cần sửa đổi lại nội dung file `AndroidManifest` và thêm tham số Debug vào để ứng dụng sau khi Build lại có thể thuận tiện cho việc Debug. Nội dung sửa đổi như sau

```
<application android:debuggable="true" android:allowBackup="false"
android:appComponentFactory="androidx.core.app.CoreComponentFactor
y" android:icon="@mipmap/ic_techday"
android:label="@string/app_name"
android:name="com.fpt.event.techday.AppTechDay"
android:roundIcon="@mipmap/ic_techday" android:supportsRtl="true"
android:theme="@style/AppTheme"
android:usesCleartextTraffic="true">
```

Sau khi Patch AndroidManifest, bạn cần thực hiện build lại APK này bằng cách `Project -> Build` hoặc click vào Icon hình cái búa. File APK được build sẽ nằm trong thư mục `dist/`

```
$ C:\Program Files (x86)\Java\jre1.8.0_241\bin\java.exe -Xmx256m -
jar E:\Tools\Android\apktool_2.4.1.jar b
C:/Users/harry/Downloads/FPTEvent/com.fpt.event.techday.apk-
```

```
decompiled
I: Using Apktool 2.4.1
I: Checking whether sources has changed...
I: Checking whether sources has changed...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...
Process exited with code 0.
```

Vẫn ở trong công cụ `ApkStudio`, bạn thực hiện việc Sign lại App. Bằng cách vào `Setting -> Signing -> Keystore Password` bạn nhập một mật khẩu bất kì để phục vụ cho việc Signing. Nếu không Signing, khi bạn cài đặt file APK nó sẽ báo lỗi như sau

```
adb: failed to install com.fpt.event.techday.apk-
decompiled/dist/com.fpt.event.techday.apk: Failure
[INSTALL_PARSE_FAILED_NO_CERTIFICATES: Failed to collect
certificates from /data/app/vmdl1617245310.tmp/base.apk: Attempt
to get length of null array]
```

Để signing file apk vừa build, bạn vào `Project -> Sign/ Export -> Sign`

```
$ C:\Program Files (x86)\Java\jre1.8.0_241\bin\java.exe -Xmx256m -
jar E:\Tools\Android\uber-apk-signer-1.1.0.jar -a
C:/Users/harry/Downloads/FPTEvent/com.fpt.event.techday.apk-
decompiled/dist/com.fpt.event.techday.apk --allowResign --
overwrite
source:

C:\Users\harry\Downloads\FPTEvent\com.fpt.event.techday.apk-
decompiled\dist
zipalign location: BUILT_IN
        C:\Users\harry\AppData\Local\Temp\uapksigner-
2219023966756182198\win-zipalign_29_0_2.exe7674529056544419623.tmp
keystore:
        [0] 686a9e6e C:\Users\harry\.android\debug.keystore
(DEBUG_ANDROID_FOLDER)
01. com.fpt.event.techday.apk
        SIGN
        file:
C:\Users\harry\Downloads\FPTEvent\com.fpt.event.techday.apk-
decompiled\dist\com.fpt.event.techday.apk (10.57 MiB)
```

```
        checksum:
7e6124f335399e56d801fa5f70b49a55596478bf0e45383cbc6172ef44021ec5
(sha256)
        - zipalign success
        - sign success
        VERIFY
        file:
C:\Users\harry\Downloads\FPTEvent\com.fpt.event.techday.apk-
decompiled\dist\com.fpt.event.techday.apk (10.66 MiB)
        checksum:
6ceca0b1a6e72bd388491122159e4f4912930952f99f60c568d9883daf34c087
(sha256)
        - zipalign verified
        - signature verified [v1, v2, v3]
                1 warnings
                Subject: C=US, O=Android, CN=Android Debug
                SHA256:
9621852afa2a065cf0c87f43624e874b23813cc8ddf4a640fcb4ace641924ca2 /
SHA1withRSA
                Expires: Tue Apr 05 14:42:25 ICT 2050
[Tue Apr 14 01:57:13 ICT 2020][v1.1.0]
Successfully processed 1 APKs and 0 errors in 2.99 seconds.
Process exited with code 0.
```

Tới đây bạn có thể cài đặt lại ứng dụng này vào máy nhưng trước hết cần gỡ ứng dụng cũ ra đã

```
$ adb uninstall com.fpt.event.techday
Success

$ adb install dist/com.fpt.event.techday.apk
Performing Streamed Install
Success
```

### Decompile Smali

Tiếp đó, bạn cần thực hiện decompile các `dex` file (mã chương trình đã được biên dịch) trong tệp tin APK thành các mã bytecode `jar`. Ở đây mình sử dụng `dex2jar` là một công cụ quá tiện lợi cho việc này. Tải `dex2jar` ở đường dẫn sau đây https://github.com/pxb1988/dex2jar

```
$ d2j-dex2jar.sh com.fpt.event.techday.apk
```

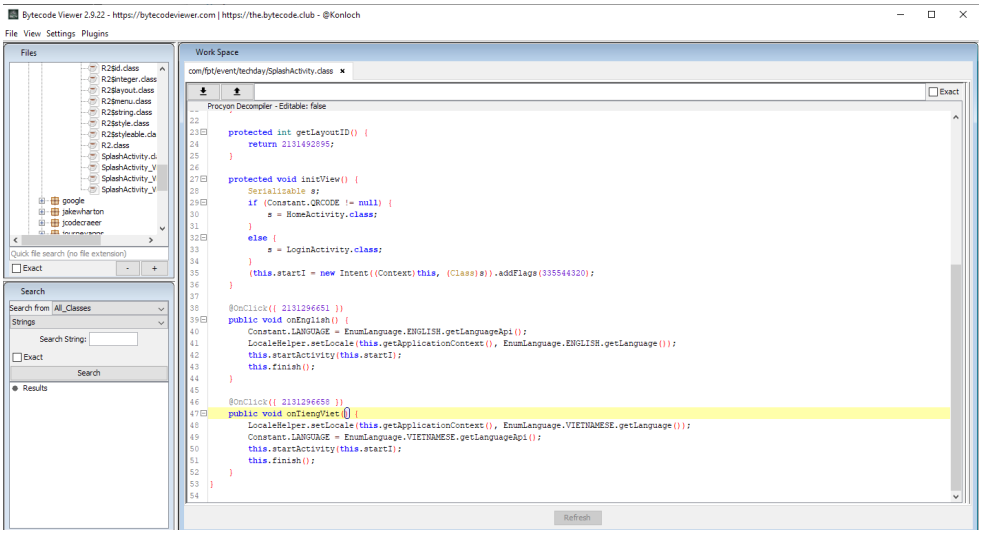
File bytecode `com.fpt.event.techday.jar` sau khi decompiled sẽ được mở bằng bytecode-viewer, tải ở đường dẫn này
https://github.com/Konloch/bytecode-viewer

Sau đó chúng ta sẽ export toàn bộ source code java/kotlin của ứng dụng Android này ra thông qua `ByteCode-Viewer`, bằng cách `File -> Decompile & Save All Classes`, sau đó lựa chọn vị trí lưu và tên của file zip chứa mã nguồn. Mình đặt tên là `source.zip`. Tiếp tục lựa chọn Decompiler cho việc chuyển đổi Bytecode sang mã Java là `Procyon`.

Bytecode-Viewer đang support những Decompiler sau:

- Procyon - fast and and well-supported decompiler for modern Java
- CFR - very good and well-supported decompiler for modern Java
- Krakatau
- Fernflower

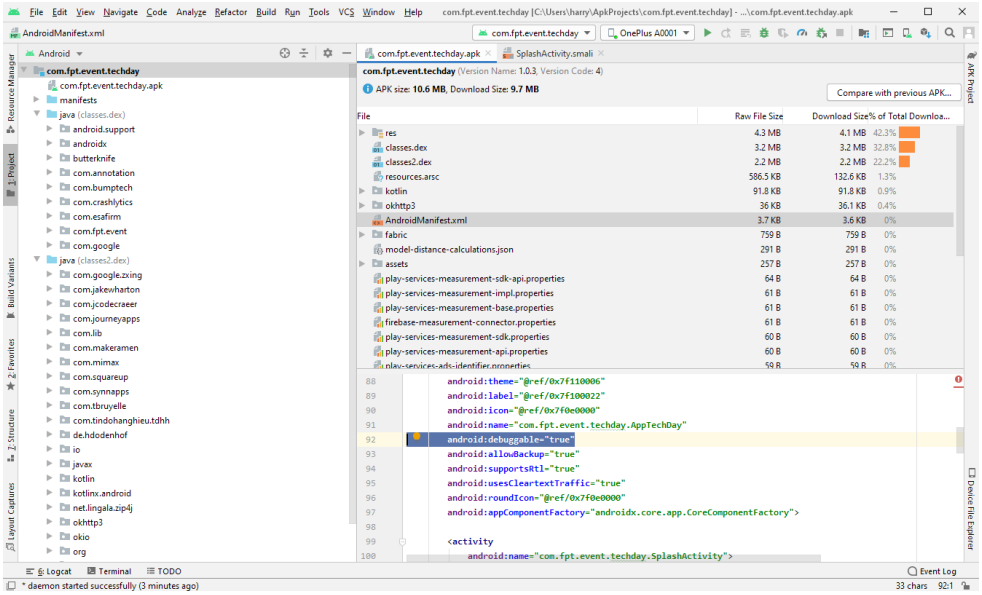
Lưu ý: Quá trình Decompile sẽ mất nhiều thời gian, nên hãy đi pha một cốc cà phê hoặc đọc một cuốn sách để chờ đợi nha




## Debug Android Studio

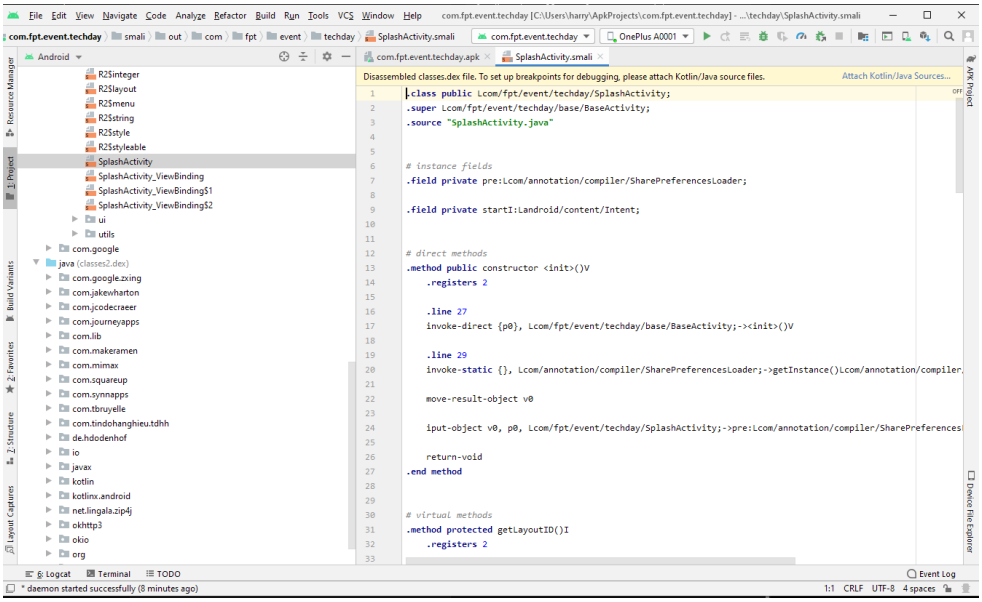
Tới bước này thì rất đơn giản, bạn mở `Androi Studio` phiên bản mình đang sử dụng là `3.6.2`.

Sau đó chọn `Profile or Debug APK`, và lựa chọn tới file APK mà bạn vừa re-build lại phía trên. Android Studio lúc này sẽ thực hiện decompile lại toàn bộ các resource.

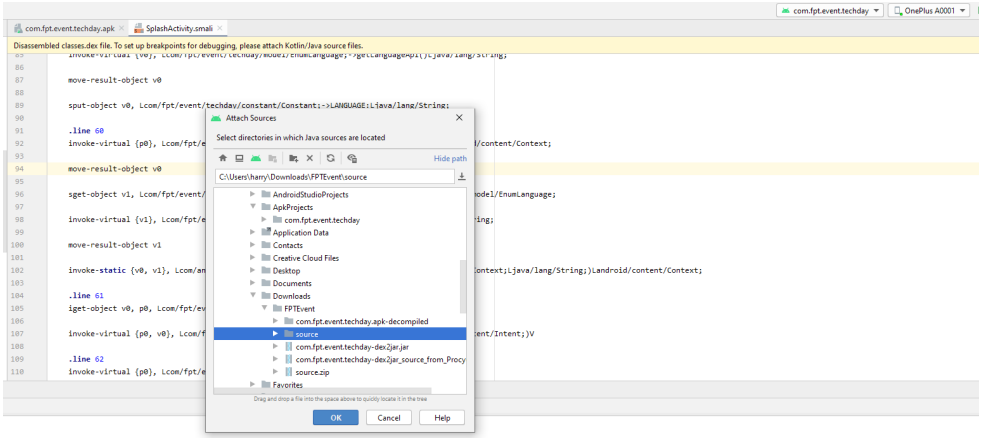


Chúng ta có thể nhận thấy, nội dung trong Manifest của APK đã được thay đổi như mong muốn.

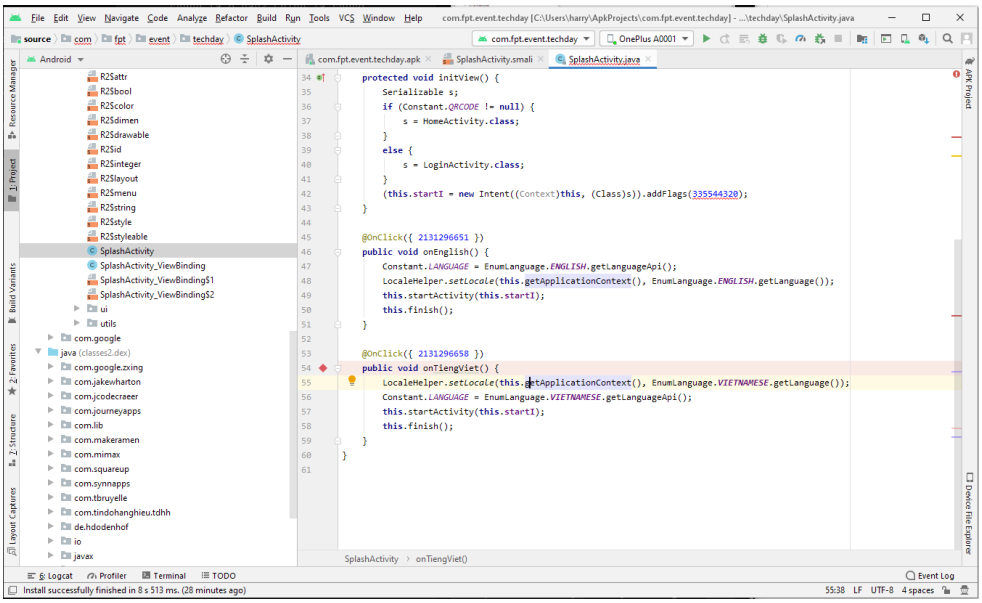
À tới đoạn này rất thú vị, bạn nhìn thấy toàn là mã smali đúng không, rất khó hiểu. Android Studio sẽ hỏi bạn có muốn `Attach Kotlin/Java Source` vào không? Tính năng này nó sẽ match các đoạn mã smali vào code Java tương ứng. Code của chúng ta ở đâu? Chính là chúng ta vừa lấy ra bằng `JD-Gui` đó.



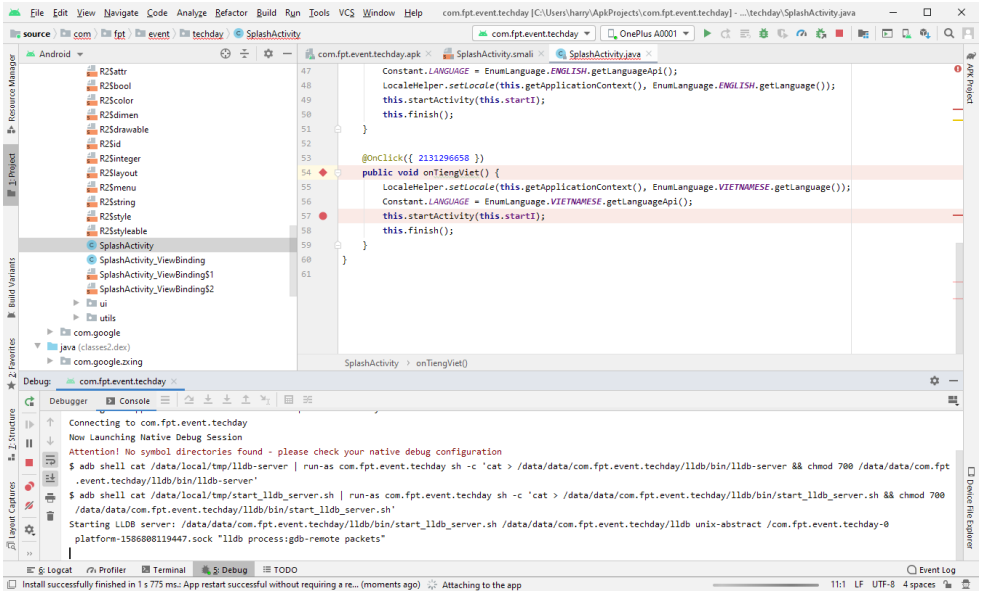
Sau khi chúng ta load tới thư mục code đã exported của Bytecode-Viewer, chúng ta sẽ thấy màn hình code lúc này đã rất rõ ràng rồi phải không?




Lúc này bạn chỉ cần lựa chọn Activity hoặc Method mà bạn muốn Debug, và đặt Break Point tại điểm đó. Ở trong ví dụ này, mình sẽ Debug màn hình chính trong package sau `com.fpt.event -> techday`, class `SplashActivity`, method `onTiengViet()`



Nhấn vào biểu tượng Debug hình con bọ màu xanh hoặc `Shift + F9`, hoặc vào `Run -> Debug 'com.fpt.event.techday'`. Lúc này chỉ cần ngồi đợi `LLDB Debuger` trình Debuger remote khởi tạo và ứng dụng trên Android kết nối tới.



Chúng ta có thể hoàn toàn kiểm soát được các flow của chương trình thông qua các lênh `Step Over - F8`, `Step Into - F7`