

Android

Kitekintés

Dr. Ekler Péter

peter.ekler@aut.bme.hu



Department of
Automation and
Applied Informatics

NFC

Near Field Communication

Near Field Communication (NFC)

- Rövidtávú vezeték nélküli kommunikációs technológia
- <4cm távolságon belül működik
- NFC tag és mobil telefon közti kis méretű adatátvitel (payload)
- Mobil telefon és mobil telefon közti kis méretű adatátvitel
- NFC Forum által meghatározott formátum: NDEF (NFC Data Exchange Format)

NFC Tag-ek jellemzői

- Írható/olvasható/egyszer írható
- Komplex Tag-ek tartalmazhatnak matematikai műveleteket is és lehet külön kriptográfia hardver egységük autentikáció céljából
- Még bonyolultabb Tag-ek akár saját működési környezettel is rendelkezhetnek és egyszerűbb kód végrehajtására is alkalmasak

NFC olvasás lépései

- Rendszerbeállításokban engedélyezni kell
- Feloldott képernyőzár esetén működik
- Tag Dispatch System a felelős
- Beolvasott NFC Tag automatikus felderítése és a megfelelő alkalmazás indítása (Intent)
- Az alkalmazások *IntentFilter* formájában jelezhetik az őket érdeklő NFC formátumokat
- Érdemes “egyedi” Tag-eket definiálni, amiket csak a saját alkalmazásunk kezel (ne szakadjon meg a kapcsolat mozgatás miatt)

Tag Dispatcher System működése

- NFC Tag parse-olása és a MIME type v. URI felderítése
- MIME type / URI azonosítja a payload (adat) típusát
- MIME type / URI és payload becsomagolása egy Intent-be
- Activity indítása az Intent alapján

NFC írás lépései

- TAG felderítése
- NDEF üzenet megfelelő összeállítása
- NDEF üzenet küldése a felderített TAG-re
- Írás sikerességének ellenőrzése

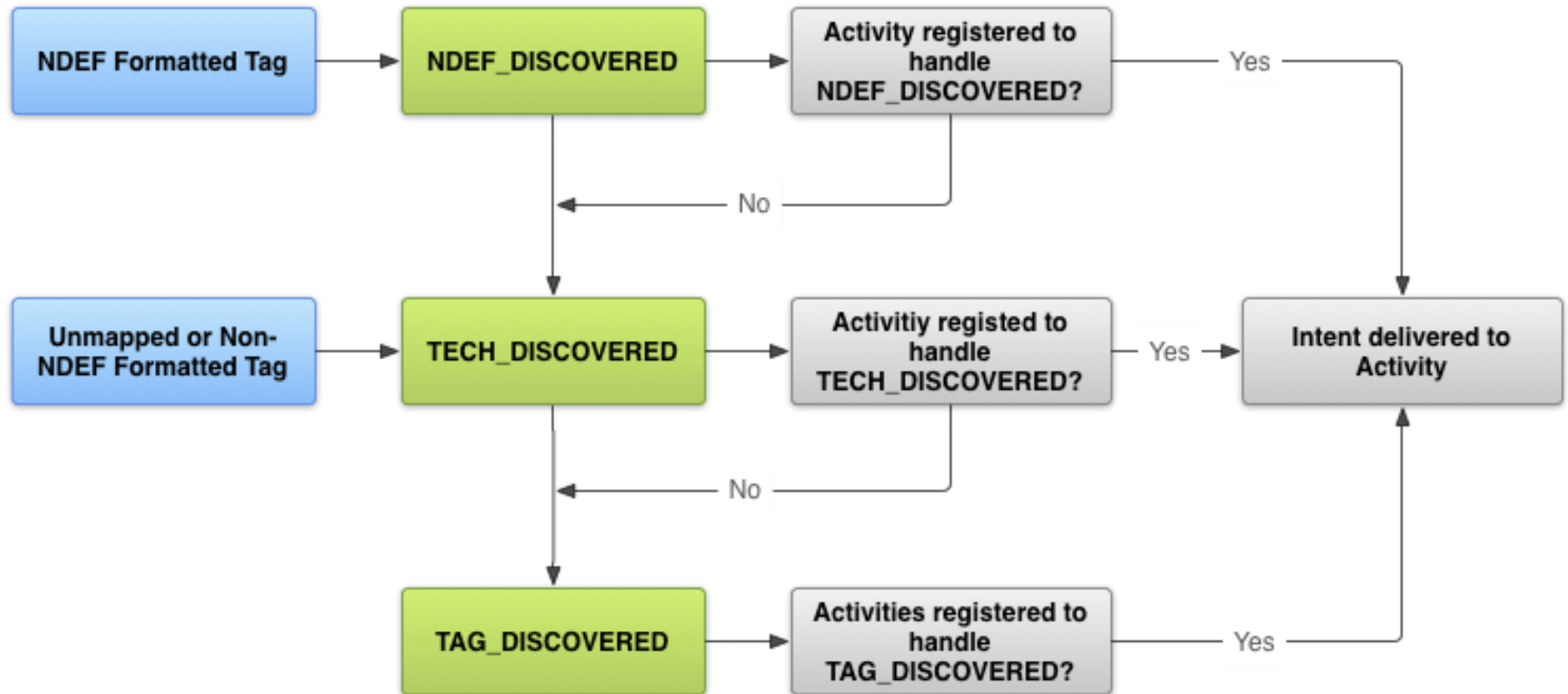
NFC Beam

- NDEF üzenet küldése két Android készülék között
- Nincs szükség készülék felderítésre
- A kapcsolat automatikusan felépül, ha két eszköz hatótávon belülre kerül
- Beépített alkalmazások is támogatják adatszerére (pl.: Browser, Névjegyzék, YouTube stb.)

NDEF adat szerkezete

- *NdefMessage*: NDEF adatot tartalmazza
- *NdefRecord*: egy *NdefMessage* egy vagy több *NdefRecord*-ot tartalmaz
- Minden *NdefRecord* egy előre meghatározott struktúrát követ
- Az *NdefMessage* első *NdefRecord*-ja határozza meg tipikusan a továbbiak feldolgozását
- Nem NDEF formátum is támogatott Android-on, de nem ajánlott, kezelésük:
 - > *android.nfc.tech* csomag osztályai

Tag Dispatch System működése



NFC használat lépései

- Permission

- > `<uses-permission android:name="android.permission.NFC" />`

- Minimum Android verzió

- > `<uses-sdk android:minSdkVersion="10" />`

- NFC hardware ellenőrzése (így nem jelenik meg azon eszközök számára, amik nem támogatják az NFC-t)

- > `<uses-feature android:name="android.hardware.nfc" android:required="true" />`

Szöveges tartalom detektálása

- Manifest-en belül a megfelelő *<activity>*-elembe:

```
<intent-filter>
```

```
<action android:name="android.nfc.action.NDEF_DISCOVERED"/>
```

```
<category android:name="android.intent.category.DEFAULT"/>
```

```
<data android:mimeType="text/plain" />
```

```
</intent-filter>
```

URI detektálása

- Szűrés a <http://www.aut.bme.hu/> URI-ra

```
<intent-filter>
```

```
<action android:name="android.nfc.action.NDEF_DISCOVERED"/>
```

```
<category android:name="android.intent.category.DEFAULT"/>
```

```
<data android:scheme="http" android:host="www.aut.bme.hu"  
android:pathPrefix="/" />
```

```
</intent-filter>
```

Beam detektálása

```
<intent-filter>
```

```
<action android:name=
```

```
"android.nfc.action.NDEF_DISCOVERED"/>
```

```
<category android:name=
```

```
"android.intent.category.DEFAULT"/>
```

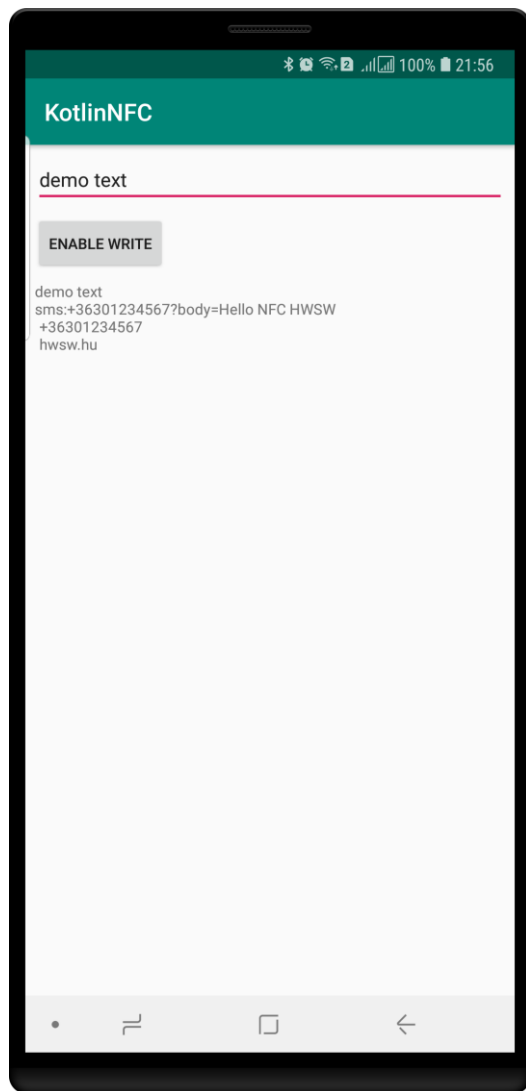
```
<data android:mimeType=
```

```
"application/vnd.com.example.android.beam"/>
```

```
</intent-filter>
```

Első NdefRecord payload-jának kiolvasása

```
override fun onResume() {  
    super.onResume()  
    if (intent.action == NfcAdapter.ACTION_NDEF_DISCOVERED) {  
        val parcelableArray = intent.getParcelableArrayExtra(  
            NfcAdapter.EXTRA_NDEF_MESSAGES)  
        parcelableArray?.forEach {  
            val ndefMsg = it as NdefMessage  
            ndefMsg?.records.forEach {  
                tvStatus.append("${String(it.payload)}\n")  
            }  
        }  
    }  
}
```



Gyakoroljunk!

- Készítsünk NFC író/olvasó alkalmazást
- Szöveges tartalom esetén induljon el automatikusan az Activity

Gyakran használt külső osztálykönyvtárak

Tippek & trükkök

- <https://github.com/nisrulz/android-tips-tricks>
- Eszközök, shortcutok
- Kódolási javaslatok
- UI/UX megoldások
- Kotlin ötletek
- Osztálykönyvtárak

Libek

- Retrofit
- RetroLambda
- Picasso
- ButterKnife
- Parceler
- IcePick
- LeakCanary
- Espresso
- Robolectric
- Dagger 2

Libek

- Zxing: QR kód olvasás
- Glide: képek betöltése hatékonyan
- MPAndroidChart
- OSMDroid: open street maps
- Google Protocol Buffers
- RxJava
- EventBus
- AndroidAnnotations

Libek

- Apache Commons
- OpenCV
- Vufoia
- Tesseract OCR
- KSOAP2
- Oauth-signpost
- ...

Játék motorok

- libGDX: <http://code.google.com/p/libgdx/>
- min3d: <http://code.google.com/p/min3d/>
- Ogre3D:
<http://www.ogre3d.org/tikiwiki/Ogre+Android>
- Unity:
<http://unity3d.com/unity/multiplatform/mobile>
- Box2D: <http://code.google.com/p/androidbox2d/>
- Cocos2D Android port:
<http://code.google.com/p/cocos2d-android/>

Lib collections

- <http://blog.autsoft.hu/the-android-and-ios-alternative-library-collection/>
- <https://github.com/wasabeef/awesome-android-ui/blob/master/README.md>
- https://github.com/codepath/android_guides/wiki/Must-Have-Libraries
- <https://android-arsenal.com/>
- ...

További eszközök

- Grafika/ikonok

- > <https://romannurik.github.io/AndroidAssetStudio/icons-launcher.html>
- > <https://unsplash.com/>
- > <https://github.com/nickbutcher/plaid>

- Android Studio Pluginek

- > <https://plugins.jetbrains.com/plugin/7275-codeglance>
- > <https://plugins.jetbrains.com/plugin/7495--ignore>
- > <https://plugins.jetbrains.com/plugin/8527-google-java-format>
- > <https://plugins.jetbrains.com/plugin/8533-json2pojo>

Statikus kódelemzés

Lint eszköz - kódvizsgálat

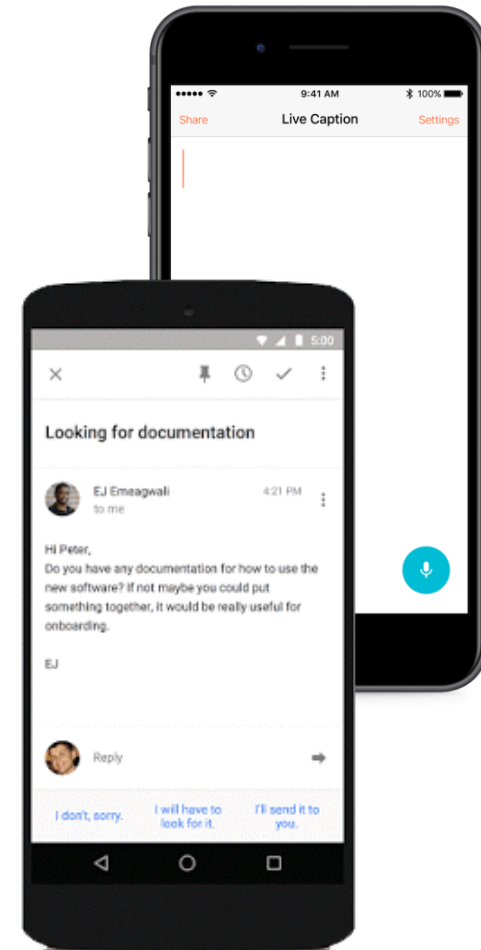
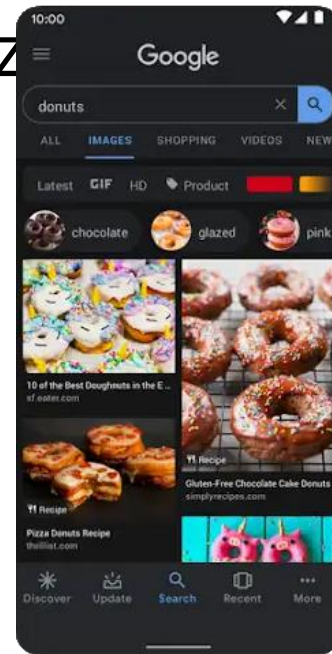
- Bekapcsolás:
 - > Window > Show View > Other > Android > Lint Warnings
- Kód strukturális problémák felderítése
- Hatékonyság, teljesítmény növelése (például nem használt XML namespacek kiszűrése)
- Deprecated kódrészek feltárása és javaslat tétel

Android történelem

[Video]

Android 10 - főbb újdonságok

- Élő feliratok
- Intelligens gyors válaszok
- Intelligens hangerő és zajszűrés
- Újfajta gesztusok
- Sötét téma
- Biztonsági fejlesztések
- Focus mode
- Family Link



További témák

- NFC
- Bluetooth
- Nearby API
- Architektúra komponensek
- Külső könyvtárak
- Statikus kódelemzés
- Android API érdekességek
- Android JetPack
- ...

Útravaló – szoftverfejlesztési elvek

Mi a mérnök feladata?

** upload, independent of whether the original activity is paused, stopped, or finished.*

**/*

```
public class Activity extends ContextThemeWrapper
    implements LayoutInflater.Factory2,
        Window.Callback, KeyEvent.Callback,
        OnCreateContextMenuListener, ComponentCallbacks2,
        Window.OnWindowDismissedCallback {

    private static final String TAG = "Activity";
    private static final boolean DEBUG_LIFECYCLE = false;

    /** Standard activity result: operation canceled. */
    public static final int RESULT_CANCELED = 0;
    /** Standard activity result: operation succeeded. */
    public static final int RESULT_OK = -1;
    /** Start of user-defined activity results. */
    public static final int RESULT_FIRST_USER = 1;

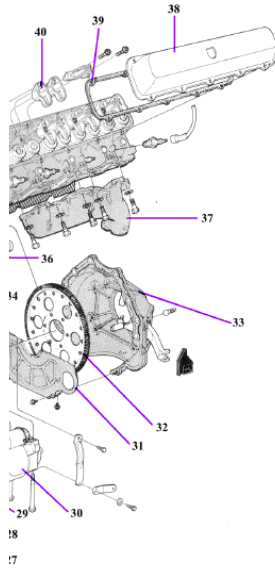
    static final String FRAGMENTS_TAG = "android:fragments";

    private static final String WINDOW_HIERARCHY_TAG = "android:viewHierarchyState";
    private static final String SAVED_DIALOG_IDS_KEY = "android:savedDialogIds";
    private static final String SAVED_DIALOGS_TAG = "android:savedDialogs";
    private static final String SAVED_DIALOG_KEY_PREFIX = "android:dialog_";
    private static final String SAVED_DIALOG_ARGS_KEY_PREFIX = "android:dialog_args_";

    private static class ManagedDialog {
        Dialog mDialog;
        Bundle mArgs;
    }

    private SparseArray<ManagedDialog> mManagedDialogs;

    // set by the thread after the constructor and before onCreate(Bundle savedInstanceState) is called.
    private Instrumentation mInstrumentation;
    private IBinder mToken;
    private int mIdem;
    /*package*/ String mEmbeddedID;
    private Application mApplication;
    /*package*/ Intent mIntent;
```



Kihívások

- Volt már valaha dolgunk rossz kóddal?
 - „Olvasni a kódot több idő, mint írni”
- Volt már olyan eset, hogy nem volt idő egy feladat „szakszerű” megoldására, a kód tisztítására, a rövid határidő miatt?

„Ahogy nő a kód mennyisége, úgy csökken a fejlesztők produktivitása.”



A PROGRAMOZÓ 15 VÁLASZA, HA NEM MŰKÖDIK A KÓDJA



Code rot (kód romlás)

- Az alkalmazások tipikusan letisztult, tiszta architektúrával indulnak
- Mi történik egy bizonyos idő után?
 - > A kód elkezd „rothadni” (romlani): kicsi hack itt-ott, egyre több *if* elágazás, mígnem az egész kódban ezek dominálnak -> átláthatatlan viselkedés
- Nehéz karbantartani, nehéz új funkciókat hozzáadni -> a fejlesztők egy idő után áttervezésért könyörögnek

Kód romlás

- A forráskód bizonyos szempontból a terv (design)
- A romlott design és a rossz architektúra tipikus tünetei
 - > Merevség
 - Folyamatosan nehezebb a kód módosítás
 - A változtatás költsége magas
 - > Törékenység
 - Apró változtatás egy modulon egy másik modulban okozhat hibás viselkedést
 - Például: egy bug javítás elront egy látszólag független részt
 - > Mozdulatlanság
 - Egy rendszer mozdulatlan, ha a részeit nem lehet könnyedén modulokba kiszervezni és máshol újra hasznosítani
 - Például: a login modul újra felhasználható legyen
 - Mozdulatlanság elkerülési stratégiák: rétegek kialakítása (adatbázis és UI különválasztása)
 - > Nyúlékonyság
 - A kód struktúra nyúlékonysága
 - Új feature implementálását könnyebb megoldani hackeléssel, mint új kód írással/új osztály bevezetésével
 - A környezet nyúlékonysága
 - Fordítás, teszt futtatás és becheckolás körülményes és sok ideig tart

Kód romlás – Mi az okozója?

- Változó követelmények
 - > Ha olyan a kódunk/architektúránk, hogy nehéz a változásokat kezelni, az a mi hibánk
 - > A kód/architektúra rugalmas kell legyen a változások követésére és meg kell akadályozza a kód romlást
- Milyen változások miatt kezd romlani a kód? *Olyan változások, amelyek új, nem tervezett dolgokat hoznak az osztály függőségek szintjén.*
- A legtöbb tünet direkt, vagy indirekt módon a modulok közti nem megfelelő függőségre vezethető vissza.
- Az objektum orientált tervezési elvek segítenek a modulok közti függőségek kezelésében.
 - > SOLID elvek

A szoftver két értéke

- **Másodlagos érték:** a szoftver viselkedése - a szoftver azt csinálja hibamentesen, amit a felhasználó elvár
- **Elsődleges érték:** Tolerálja és egyszerűen alkalmazkodik a folyamatos változásokhoz, tehát könnyű módosítani (software is soft)



Függőség kezelő OO elvek: SOLID elvek

- Single Responsibility Principle (S.R.P.)
- Open Closed Principle (O.C.P.)
- Liskov Substitution Principle (L.S.P.)
- Interface Segregation Principle (I.S.P.)
- Dependency Inversion Principle (D.I.P.)

Mit értünk TDD alatt?

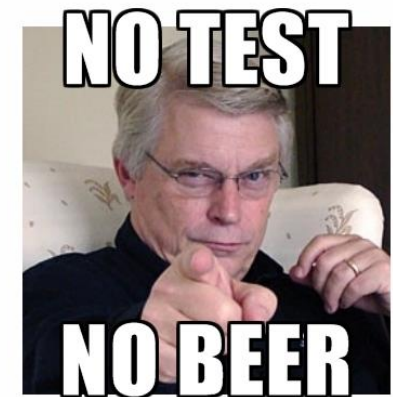
- Hagyományos fejlesztési ciklus (hosszú ciklus, több hónapos):



- Test-driven development ciklus (ismétlődő rövid ciklusok, pár perc):



- A TDD egy programozási technika, ami a következő gondolaton alapul: csak egy bukó teszt kizöldítésére írunk production kódot
- A TDD három szabálya:
 - Írunk egy elbukó tesztet
 - Írj csak annyi production kódot, ami kizöldíti a tesztet
 - Tisztítsd (refactor) a tesztet és a production kódot
- Red-green-refactor



memegenerator.co

Modern fejlesztő eszközök

- Continuous Integration
 - > Jenkins
 - > Project status
- Kódminőség ellenőrzés
 - > Sonar
- Elosztott verziókezelő!
 - > Pl.: GIT



Cserkész szabály

- Nem elég jól megírni a kódot, hanem tisztán is kell tartani
- „Hagyd a tábort tisztábban, mint ahogy kaptad!”

Irodalom

- Agile Software Development by Robert C. Martin
- Clean Code Video Series by Robert C. Martin
- www.objectmentor.com



Android interjú kérdések

1. Describe the APK format.
2. What are the different phases of the Activity life cycle?
3. What is the significance of the .dex files?
4. What is the difference between Service and Thread?
5. What is a Content Provider?
6. When does ANR occur?
7. What is the difference between a regular bitmap and a nine-patch image?

<https://github.com/MindorksOpenSource/android-interview-questions>

Tanfolyam összefoglalása

- Android bevezetés
- Kotlin alapok
- Activity életciklus
- Felhasználói felület, layoutok, fragmentek, animációk, stb.
- Intent
- BroadcastReceiver
- Engedélyek kezelése
- Perzisztens adattárolás (SharedPreferences, File, ORM – Room)
- Hálózati kommunikáció
- Helyfüggő szolgáltatások
- Multimédia
- Services
- ContentProvider
- Grafikonok
- Szenzorok

Köszönöm a figyelmet

