

---

# Android alapú szoftverfejlesztés

## 7. Labor

### Szolgáltatások készítése és Content Provider használata Android platformon

#### Tartalom

1	Felkészülés a laborra.....	2
2	Az alkalmazás működése.....	2
3	API kulcs igénylése Google Maps használatához .....	4
3.1.1	Manifest paraméterek.....	7
4	Laborfeladatok .....	9

## 1 Felkészülés a laborra

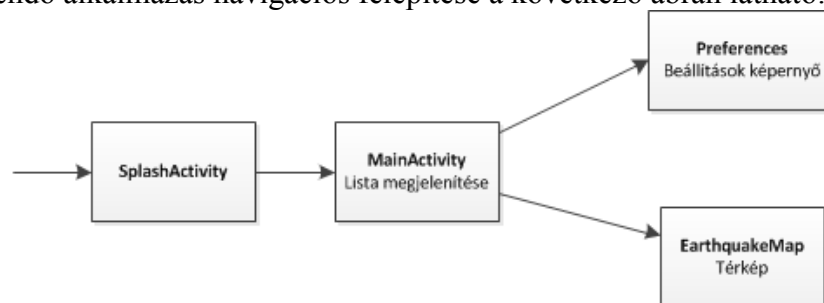
A labor célja az Android **Service** használatának, valamint az alkalmazáson belüli adat réteg elkülönítését szolgáló saját **ContentProvider** létrehozásának és használatának bemutatása. Az elkészülő alkalmazás egy nyilvános webszolgáltatás segítségével földrengések adatait tárolja és jeleníti meg listában illetve térképen. Az adatok frissítését egy Service végzi, amit az AlarmManager osztály segítségével periodikusan újraindítunk a felhasználó által megadott beállításoknak megfelelően. A szerverről letöltött XML feldolgozása után az adatok SQLite adatbázisba kerülnek, melyet az alkalmazás nem közvetlenül, hanem ContentProvideren keresztül olvas és ír. Új földrengés letöltésekor Notification jelenik meg, melyre kattintva elindul az alkalmazás, és megjeleníti a listát.

Felhasznált technológiák:

- Service
- AsyncTask
- ContentProvider
- AlarmManager
- Broadcast Receiver
- Preferences Framework
- Map Overlay
- Notification

## 2 Az alkalmazás működése

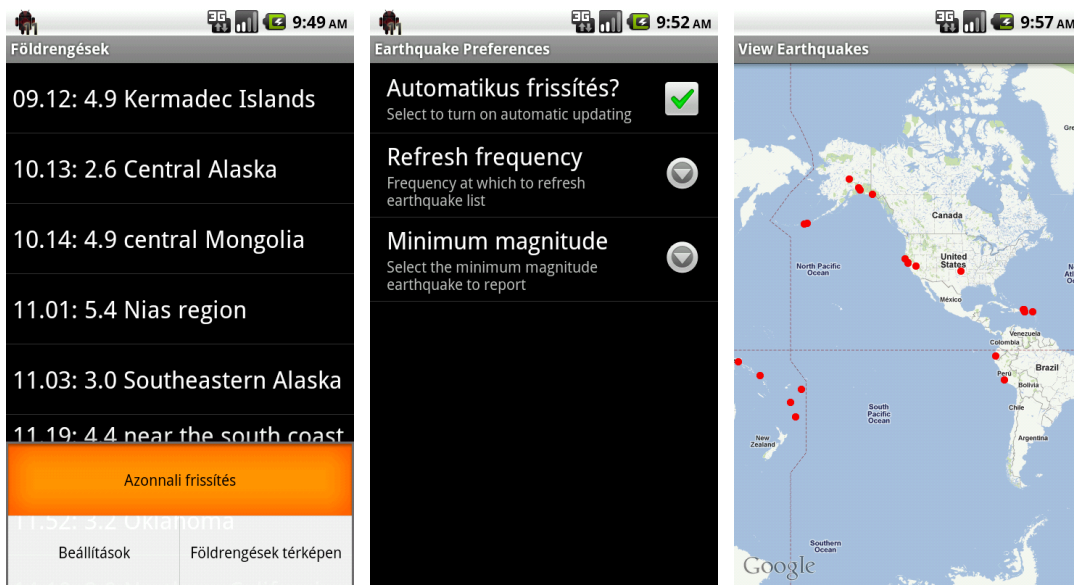
Az elkészítendő alkalmazás navigációs felépítése a következő ábrán látható:



A splash screen megjelenítése után az alkalmazás a lista képernyővel folytatódik, ami első induláskor üres. Az itt lévő menü három lehetőséget biztosít a felhasználó számára:

- Azonnali frissítés
- Beállítások
- Térkép megjelenítése

A képernyők mintaképei:



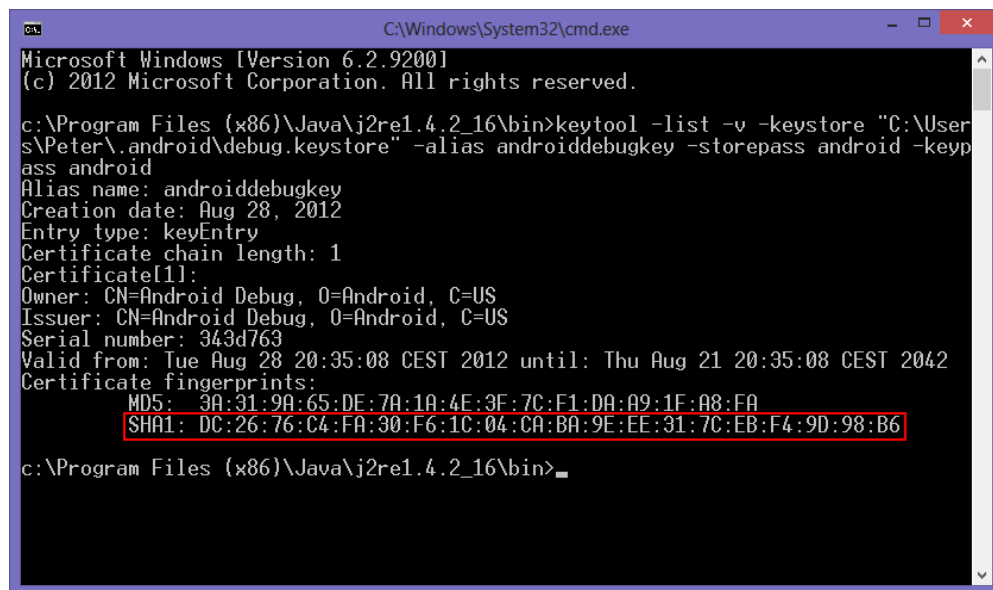
Az alkalmazás felépítése és működése a következő:

- Az entitás osztályunk a Quake, ez reprezentálja a földrengéseket a memóriában.
- Az adatokat az EarthQuakeProvider saját ContentProvider leszármazott szolgáltatja. Ez SQLite adatbázist használ a tényleges tárolásra, azonban elfedi azt a kötelezően megvalósítandó metódusai segítségével. Érdekes megfigyelni a statikus UriMatcher objektum létrehozását és használatát arra, hogy a lekérésekben használt Uri-kat elkülöníthessük
- A MainActivity jeleníti meg a földrengések listáját, valamint menün keresztül lehetőséget biztosít a szinkronizáló Service indítására
- Preferences Framework használatával beállítások képernyő. Megadhatjuk, hogy legyen-e automatikus frissítés, annak gyakoriságát, valamint hogy legalább milyen erősségű földrengéseket szeretnénk látni.
- Az EarthquakeService szolgáltatás végzi a földrengések szinkronizálását. Mivel egy nyilvános webservice-t hívunk, a blokkoló hálózati műveletet, valamint a CPU igényes XML feldolgozást nem a Service fő szálában, hanem AsyncTask segítségével egy háttérszámban végezzük. Érdekes megfigyelni, hogy az onStartCommand callback függvényben – a beállításnak megfelelően – ismétlődő riasztást hozunk létre. Az AlarmManager osztály a nevével ellentétben nem csak a felhasználó értesítésével járó eseményket (pl. ébresztő) képes generálni, hanem a setRepeating metódusával bármilyen Intent elsüthető beállított időközönként. Ezt használjuk a Service időzített futtatására, a webservice periodikus lekérdezésére.

### 3 API kulcs igénylése Google Maps használatához

Elsőként indítsunk egy parancssori ablakot és olvassuk ki a debug kulcsból az SHA1 lenyomatot konzolból a következő paranccsal (a keytool.exe a JDK/bin mappában található):

```
keytool -list -v -keystore  
"C:\Users\your_user_name\.android\debug.keystore" -alias androiddebugkey -  
storepass android -keypass android
```



```
C:\Windows\System32\cmd.exe  
Microsoft Windows [Version 6.2.9200]  
(c) 2012 Microsoft Corporation. All rights reserved.  
  
c:\Program Files (x86)\Java\j2re1.4.2_16\bin>keytool -list -v -keystore "C:\User  
s\Peter\.android\debug.keystore" -alias androiddebugkey -storepass android -keyp  
ass android  
Alias name: androiddebugkey  
Creation date: Aug 28, 2012  
Entry type: keyEntry  
Certificate chain length: 1  
Certificate[1]:  
Owner: CN=Android Debug, O=Android, C=US  
Issuer: CN=Android Debug, O=Android, C=US  
Serial number: 343d763  
Valid from: Tue Aug 28 20:35:08 CEST 2012 until: Thu Aug 21 20:35:08 CEST 2042  
Certificate fingerprints:  
MD5: 3A:31:9A:65:DE:7A:1A:4E:3F:7C:F1:DA:A9:1F:A8:FA  
SHA1: DC:26:76:C4:FA:30:F6:1C:04:CA:BA:9E:EE:31:7C:EB:F4:9D:98:B6  
  
c:\Program Files (x86)\Java\j2re1.4.2_16\bin>
```

Ezt követően regisztráljuk a projektünket a Google APIs Console-on:

<https://code.google.com/apis/console/?noredirect>

A Console-on bal oldalt a **Services** menüpontot választva kapcsoljuk be a „**Google Maps Android API v2**” szolgáltatást.

	Google Cloud Storage JSON API		<input type="checkbox"/> OFF	
	Google Compute Engine		<input type="checkbox"/> OFF	<a href="#">Pricing</a>
	Google Contacts CardDAV API		<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
	Google Maps Android API v2		<input checked="" type="checkbox"/> ON	
	Google Maps API v3		<input checked="" type="checkbox"/> ON	Courtesy limit: 25,000 requests/day • <a href="#">Pricing</a>
	Google Maps Coordinate API		<input type="checkbox"/> OFF	Courtesy limit: 1,000 requests/day
	Google Maps Engine API		<input type="checkbox"/> OFF	Courtesy limit: 10,000 requests/day
	Google Maps Geolocation API		<input type="checkbox"/> OFF	Courtesy limit: 0 requests/day • <a href="#">Pricing</a>

Ezt követően a Console-on szintén bal oldalt az **API Access** menüt választva, az új oldalon alul válasszuk a „**Create new Android key..**” menüt.

API Project

Overview

Services

Team

API Access

Billing

Reports

Quotas

BigQuery

### API Access

To prevent abuse, Google places limits on API requests. Using a valid OAuth token or API key allows you to exceed anonymous limits by connecting

#### Authorized API Access

OAuth 2.0 allows users to share specific data with you (for example, contact lists) while keeping their usernames, passwords, and other information private. A single project may contain up to 20 client IDs. [Learn more](#)

[Create an OAuth 2.0 client ID...](#)

#### Simple API Access

Use API keys to identify your project when you do not need to access user data. [Learn more](#)

##### Key for Android apps (with certificates)

API key: [REDACTED]

Android apps: DC:5B:CE:88:1F:7E:87:42:37:D7:9E:C6:4E:AB:7A:80:D3:54:5F:FB;bute.copi

Activated on: Jul 28, 2013 10:07 AM

Activated by: maco13@gmail.com – you

##### Key for Android apps (with certificates)

API key: [REDACTED]

Android apps: 73:7D:C2:55:74:B0:BB:D7:99:8D:8A:EC:1F:80:1D:DD:E0:AA:BA:B5;com.example.jon8xz.mapslabor

Activated on: Apr 3, 2013 5:31 AM

Activated by: maco13@gmail.com – you

##### Key for browser apps (with referers)

API key: AIzaSyC5G8Y3Nufmz0HU81svYMDGgXUx1AraU1Q

Referers: Any referer allowed

Activated on: Jan 8, 2012 8:45 AM

Activated by: maco13@gmail.com – you

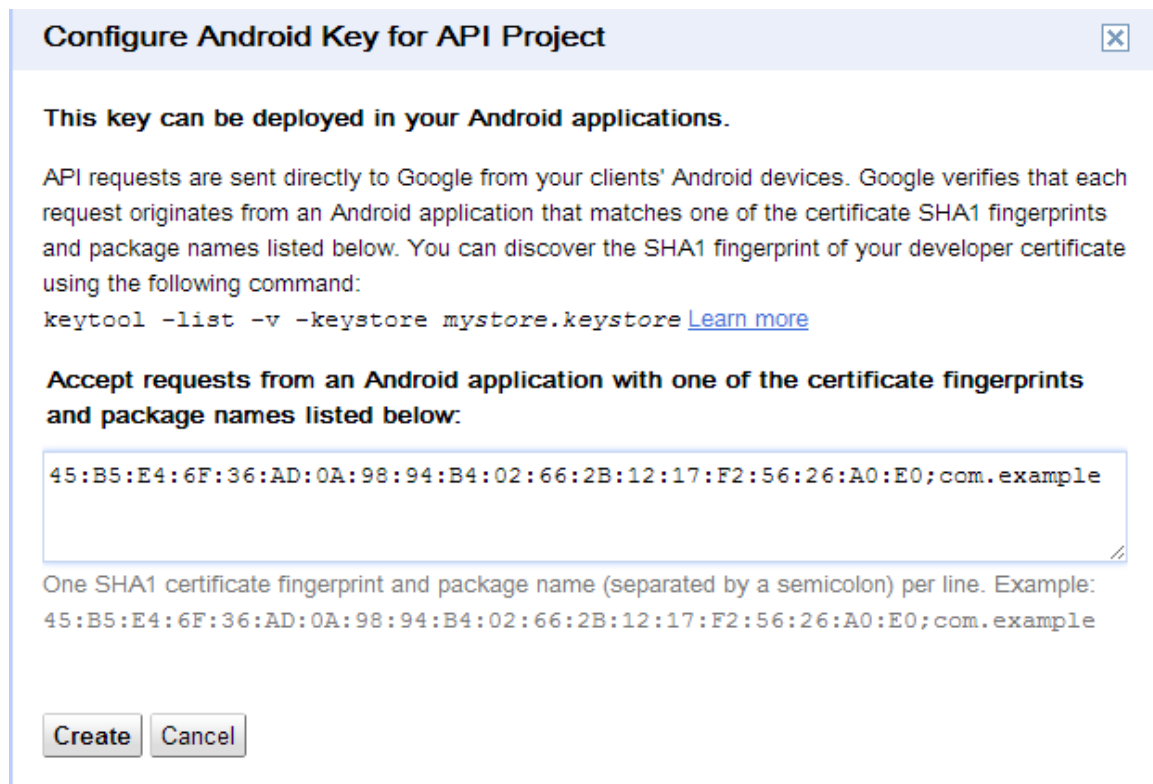
[Create new Server key...](#) [Create new Browser key...](#) [Create new Android key...](#) [Create new iOS key...](#)

#### Notification Endpoints

Use notification endpoints to identify domains that may receive webhook notifications from your API. [Learn more](#)

Allowed Domains: No domains allowed

Adjuk meg a keytool-al kiolvasott SHA1 kulcsot, majd utána ; jellel az Android alkalmazásunk fő package azonosítóját ([SHA1];package).



Sikeres generálás után másoljuk ki az API key-t, mivel szükségünk lesz rá a Manifest állományba.

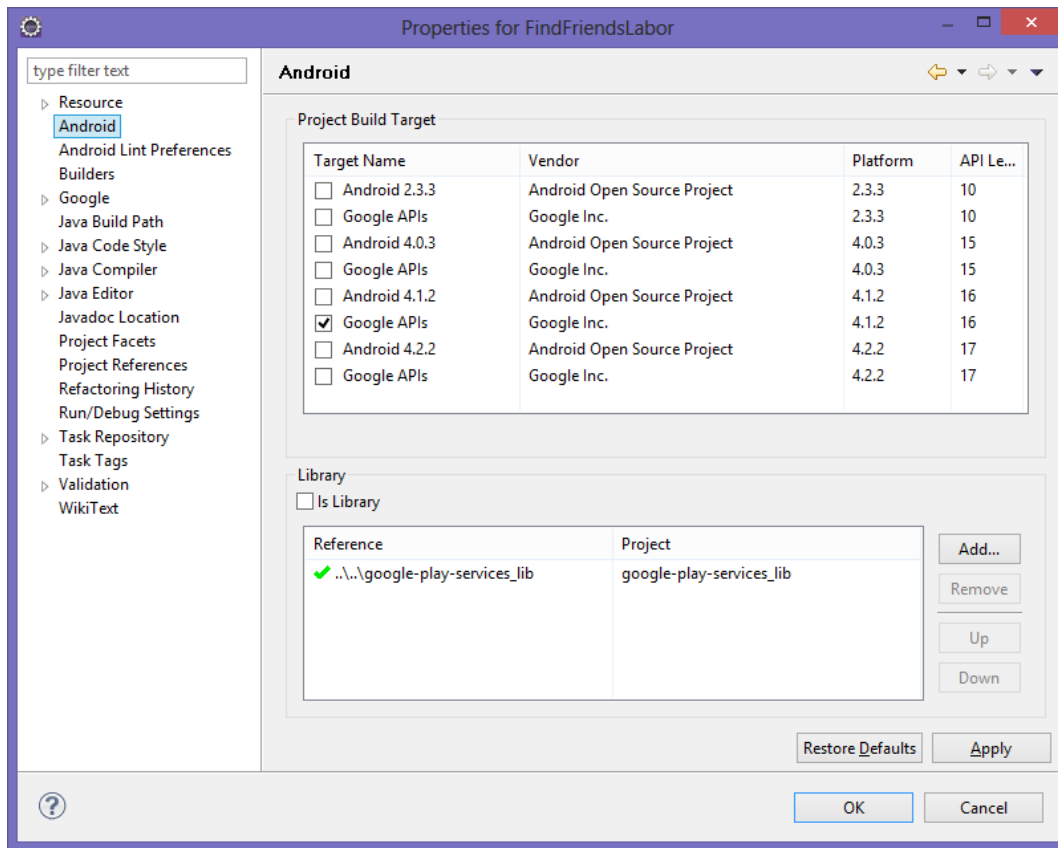
A Maps API V2 használatához a Google Services library betöltése szükséges, ez azonban nem egy egyszerű .jar betöltését jelenti, hanem az SDK-ban található google-play-services\_lib projektet kell importálnunk (mivel a projekt több erőforrást is tartalmaz, amit nem lehetne .jar-ba csomagolni)

Eclipse-be importáljuk be ezt a library projektet: Jobbegér->import->"Existing Android Code into Workspace".

A library projektet a labor anyagában is elhelyeztük, de az SDK-ban elvileg az alábbi helyen érhető el, ha letöltöttük az SDK manager-el:

[Android-sdk]\extras\google\google\_play\_services\libproject\google-play-services\_lib

Ezt követően a saját projektünk beállításában (jobbégér->properties) vegyük fel a hivatkozást a library projektre:



### 3.1.1 Manifest paraméterek

A **Laborfeladatok** 1. lépése után az AndroidManifest állományba vegyük fel a következő engedélyeket:

```
<uses-feature
    android:glEsVersion="0x00020000"
    android:required="true" />

<permission
    android:name="amorg.lab011.service.permission.MAPS_RECEIVE"
    android:protectionLevel="signature" />

<uses-permission
    android:name="amorg.lab011.service.permission.MAPS_RECEIVE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"
/>

<uses-permission
    android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission
    android:name="com.google.android.providers.gsf.permission.READ_GSERVICES"
/>
```

**Fontos, hogy az engedélynél a saját alkalmazásunk csomag azonosítóját írjuk be a megjelölt helyen!**

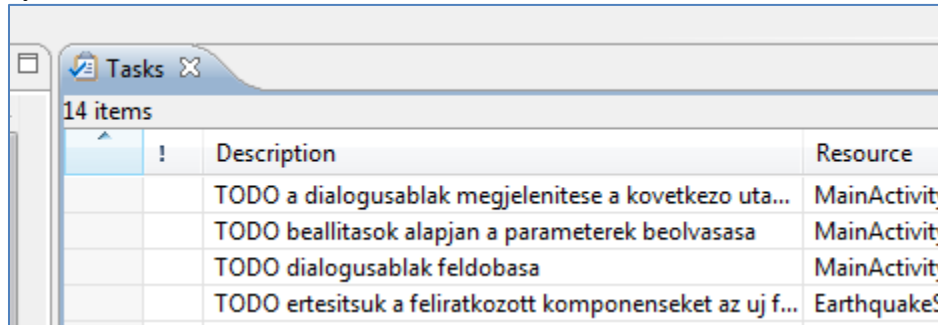
Illetve a Manifest állományban még az *<application>* tag-en belül be kell állítani az API kulcsot, amit korábban generáltunk:

```
<application
  ... >
  <meta-data
    android:name="com.google.android.maps.v2.API_KEY"
    android:value="AIzaSyAlrRYT_x5GRwY5y17qMkyFxZzvzGqBw4c" />
```



## 4 Laborfeladatok

A labor során teljesítendő feladatokat a skeletonban „// TODO” előtaggal láttuk el, így érdemes az Eclipse „Tasks” nézetét megjeleníteni, ahol a fejlesztő környezet listáz minden ilyen elemet.



	Description	Resource
!	TODO a dialogusablak megjelenítése a kovetkezo uta...	MainActivity
	TODO beallitasok alapjan a parameterek beolvasasa	MainActivity
	TODO dialogusablak feldobasa	MainActivity
	TODO ertesitsuk a feliratkozott komponenseket az uj f...	EarthquakeS

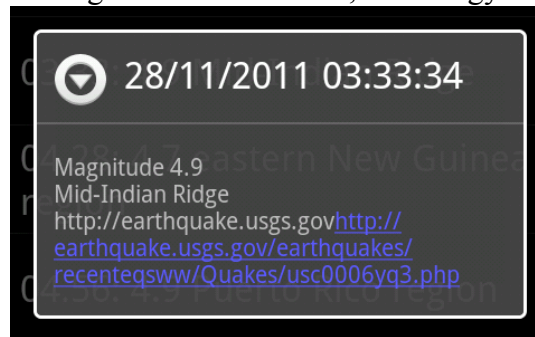
### A feladatok kifejtve:

1. A kiinduló alkalmazás **skeleton letöltése** a tárgyhonlapról
2. A kód felépítésének vizsgálata. Figyelje meg az egyes komponensek implementációját és együttműködését. **Értse meg a kódot!**
3. A **res/layout/earthquake\_map.xml** fájlban a MapView elemhez írja be a gépéhez generált **Maps API kulcsot**

A MainActivity-ben oldja meg a következő feladatokat:

4. Valósítsa meg az **addQuakeToArray** metódust. Ennek feladata, hogy a kapott Quake objektumot hozzáadja a memóriában lévő, földrengéseket tároló tömbhöz, amennyiben a beállításokban lévő minimális erősségnek megfelel. Ebben az esetben hozzáadás után értesítse a lista adapter az adatok változásáról
5. A **loadQuakesFromProvider** metódus az *EarthquakeProvider* segítségével betölti az összes földrengést a megjelenített listába. A cursor feldolgozó ciklus végén az új földrengés objektumot adja hozzá a listához! Használja az előző pontban megírt metódust.
6. Valósítsa meg a **menü megjelenítését**, illetve az egyes elemek kiválasztásának **kezelését**. A szükséges konstansok rendelkezésre állnak. Figyeljen arra, hogy a Beállítások képernyőről visszatérve nem szükséges frissíteni a paramétereket abban az esetben, ha a visszatérés RESULT\_CANCEL értékkel történt. Ellenkező esetben használja az *updateFromPreferences* metódust (megvalósítása később)

- Amennyiben a felhasználó valamely lista elemre kattint, jelenjen meg egy dialógusablak a kiválasztott földrengés minden adatával, a link legyen kattintható.



Egy példa a dialógusablakra:

- A **refreshEarthquakes** metódus hívásának hatására azonnal induljon a földrengések frissítésére hivatott Service
- Valósítsa meg az **updateFromPreferences** metódust, melyben a tárolt beállítások alapján értéket adunk a `minimumMagnitude`, `updateFreq` és `autoUpdate` változóknak

## A további feladatokat oldja meg a megfelelő forrásfájlokban!

- Kezelje le egy Intent beérkezését az **EarthquakeAlarmReceiver** osztályban. Ez a `BroadcastReceiver` leszármazott akkor lép életbe, ha valahonnan `ACTION_REFRESH_EARTHQUAKE_ALARM` akcióval rendelkező Intent-et süttöttek el, funkciója tehát a földrengések frissítésének indítása.
- Valósítsa meg az **EarthquakeService onBind** metódusát úgy, hogy más alkalmazásból ne lehessen elérni a szolgáltatást.
- A Service-en belüli **AsyncTask onPostExecute** callback metódusa akkor fut le, ha a háttérszál befejezte a `run` metódusban végzett érdemi munkát. Gondoskodjon arról, hogy a service megálljon, mivel további futására már nincs szükség.
- Az **announceNewQuake** metódus feladata egy Broadcast üzenet küldése az új földrengés adataival. Az Intent összeállítása után implementálja a feliratkozott komponensek értesítését!
- A kész, működő alkalmazást mutassa be a laborvezetőnek!