1. Definisemo unutar GradleScripts -> gradle.properties API\_KEY (MyTheMovieDBApiToken) koji cemo koristiti build.gradle (Module:app).
2. Definisemo zavisnosti (dependencies) unutar GradleScripts -> build.gradle (Module:app) koje su nam potrebne
3. **MainActivity.java** sadrzi funkijce za prelazak iz jedne u drugu aktivnost. Funkcije imaju parametar tipa View (View view), sto znaci da moraju biti dodijeljeni nekom od View elemanata u **activity\_main.xml**. **activity\_main.xml** se koristi za prikaz sadržaja **MainActivity.java**. U našem slučaju to su Buttoni GO TO REGISTER i GO TO LOGIN

**actitvity\_main.xml** se sastoji od slike koja se dodaje u res folder, podfolder drawabale. U atrbut **android:background** koji se odnosi na pozadinu prikaza, dodajemo sliku iz foldera drawable, koju smo prethodno skinuli sa interneta.

1. **Register.java**

**private void CreateAccount()** - Za dodavanje podataka u MySQL bazu podataka koristli smo objekat klase JSONObject , te metodu put. JSONObject se piše uvijek u paru ((request.put (KEY\_FIRST\_NAME, str\_first\_name)). Nakon toga se pravi objekat klase JSONObjectRequest koji sadrzi tip zahtjeva ( Request.Method.POST), apsolutnu putanju do php fajla (request\_url), zatim objekat klase JSONObject, novi slušatelj odgovara tipa klase JSONObject (new Response.Listener<JSONObject>(). Nakon toga se automatski dodaju funkcije public void onResponse (JSONObject response) i new Response.ErrorListener().

**public void displayLoader()** - vizuelni prikaz dok čekamo učitavanje podataka u bazu.

**public void getValues()**  - preuzimamo View elemente (EditText) iz activity\_register.xml po id-u.

**private boolean validateInputs()** – provjerava da li su svi podaci uneseni prije same registracije

**public void Restart()** – nakon uspjesne registracije, postavljamo vrijednosti View elemenata koje smo ranije preuzeli iz activity\_register.xml po id-u, na prazna polja.

**public void ConfirmRegister()** – Nakon uspjesne registracije prelazimo iz Register u Login.

Unutar **protected void onCreate(Bundle savedInstanceState)** se izvršavaju sve ranije kreirane funkcije.

Prvo pozivamo **getValues().**

Preuzimamo button po id-u iz activity\_register.xml.

Nakon toga definišemo sta ce se desiti kada korisnik klikne na dugme (register.setOnClickListener). Automatski se kreira **onClick (View view).**

Zatim unutar funckije **onClick** preuzimamo vrijednosti koji su uneseni u EditText (*str\_first\_name=et\_first\_name.getText().toString().trim()*;) I pohranjujemo u prethodno kreirane varijable)

Zatim pomoću if uslova provjeravamo da li su uneseni podaci. Ako jesu izvršvaju se sljedeće funkcije :

* CreateAccount();
* Restart();
* ConfirmRegister();

public void BackToLogin(View view) - postavljena na TextView activity\_register.xml. ( I have already an account).

1. Login radi na isti način kao i Register
2. MySingleton.java – koristimo za dodavanje podataka u MySQL bazu, preko php-a. I To na način da u niz redova dodaje niz ranije definisanih zahtjeva **addToRequestQueue(jsArrayRequest)**
3. MoviesAdapter klasa – definisanje informacija o filmu koji ce se prikazivati u card\_view
4. Client – definišemo API u JAVA interfejs
5. Interfejs Service – za preuzimanje podataka pomoću GET metode
6. Movie – definiše podatke o filmu
7. MovieResponse - definise podatake koji ce se povuci pomoću API-ija
8. Sadržaj MovieMainActivity.java se ispisuje unutar actiity\_movie\_main , koji uključuje content\_main
9. Sadržaj DetailActivity.java se ispisuje unutar activity\_detail i content\_detail.
10. Dobijenje apija:
    1. Kreiramo racun na themoviedb.org
    2. Account -> Settings -> API
    3. Popune se podaci
    4. Dobije se API\_KEY
    5. Napravi se api request
    6. Pokrene se u novom tabu
    7. Kopiramo te podatke
    8. Online JSONConverter i ubacimo podatke iz prethodnog koraka
    9. I dobijene podatake kopiramo u Movie i MovieResponse.