

מבחן מסכם - אנדרואיד

פיתוח אפליקציה מלאה



TMDB



GSON



Picasso



AsyncTasks



RecyclerView



SharedPreferences



TMDB

ה-API החינמי הגדול ביותר לסרטים ולסדרות.
ה-API מאפשר שאילתות על סרטים וסדרות ואף לנהל חשבונות משתמשים.

יש להרשם ל-API כדי לקבל Api Key.

שאילתה למציאת הפופולריים ביותר בשנת 2015:

Most Popular Movies of 2017

[http://api.themoviedb.org/3/discover/movie?](http://api.themoviedb.org/3/discover/movie?api_key=b3b1492d3e91e9f9403a2989f3031b0c&primary_release_year=2017&sort_by=popularity.desc)

[api_key=b3b1492d3e91e9f9403a2989f3031b0c&primary_release_year=2017&sort_by=popularity.desc](http://api.themoviedb.org/3/discover/movie?api_key=b3b1492d3e91e9f9403a2989f3031b0c&primary_release_year=2017&sort_by=popularity.desc)

התוצאה הינה JSON שמכיל מערך `.results[]`.

שאילתה לקבלת תמונה לסרט. הפרמטר כאן הוא הנתבי לתמונה.

Image For specific movie - Using the poster_path / backdrop_path

http://image.tmdb.org/t/p/w185//dkMD5qlogeRMiEixC4YNPUvax2T.jpg?api_key=b3b1492d3e91e9f9403a2989f3031b0c

שאילתה לקבלת כל המידע לסרט מסויים. הפרמטר כאן הוא ה-id של הסרט:

//Details for movie by movie id

https://api.themoviedb.org/3/movie/14564?api_key=b3b1492d3e91e9f9403a2989f3031b0c

תיעוד של ה-API:

<https://www.themoviedb.org/documentation/api/discover>

TMDB

תוצאת השאילתה מעמוד קודם:

```
▼{
  "page": 1,
  "results": [
    ▼{
      "adult": false,
      "backdrop_path": "\/biN2sqExViEh8IYSJrXlNKjpjxx.jpg",
      "genre_ids": [
        27
      ],
      "id": 14564,
      "original_language": "en",
      "original_title": "Rings",
      "overview": "Julia becomes worried about her boyfriend, Holt when he explores the dark  
the watcher seven days after viewing. She sacrifices herself to save her boyfriend a  
a \"movie within the movie\" that no one has ever seen before.",
      "popularity": 110.576208,
      "poster_path": "\/tNKdx3pWs3Y65juoujoqylWKiUO.jpg",
      "release_date": "2017-02-01",
      "title": "Rings",
      "video": false,
      "vote_average": 5.3,
      "vote_count": 161
    },
    ▼{
      "adult": false,
      "backdrop_path": "\/4G6FNNLSIVrwsRZyFs91hQ3lZtD.jpg",
      "genre_ids": [
        18,
        27,
        53
      ],
      "id": 381288,
      "original_language": "en",
      "original_title": "Split",
      "overview": "Though Kevin has evidenced 23 personalities to his trusted psychiatrist, D  
set to materialize and dominate all the others. Compelled to abduct three teenage gi  
a war for survival among all of those contained within him—as well as everyone aroun  
apart.",
      "popularity": 67.473076,
      "poster_path": "\/rXMWOZiCt6eMX22jWuTOSdQ98bY.jpg",
    }
  ]
}
```

GSON

ספריית קוד שמטרתה לאפשר המרת JSON לאובייקטים ללא מאמץ.

כל מה שנדרש הוא ליצור מחלקה מתאימה והספרייה פשוט ממירה במקומנו את ה-JSON.

לדוגמא:

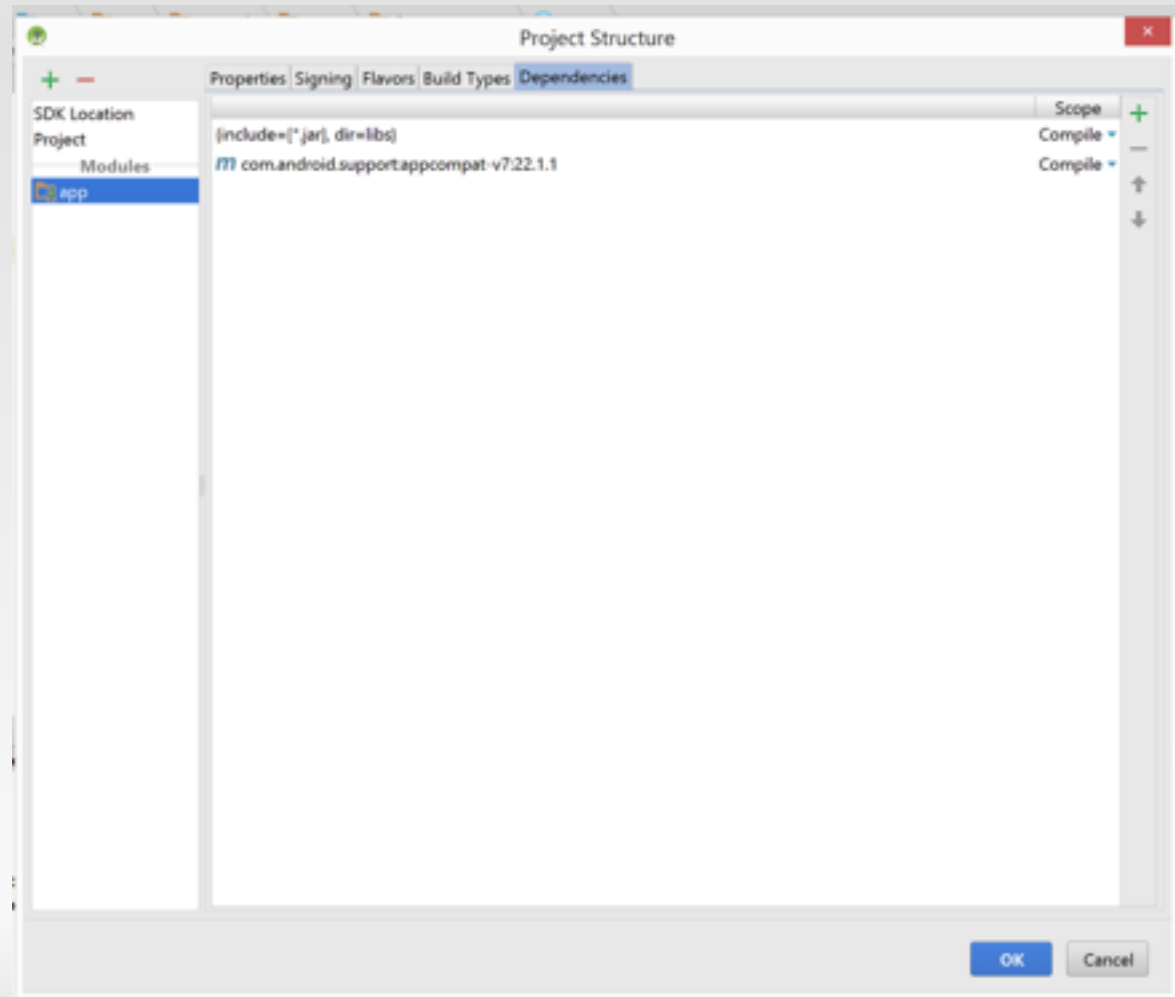
```
Gson gson = new Gson();  
Person[] people = gson.fromJson(jsonString, Person[].class);
```

(1) יצרנו מחלקה בשם Person שמתאימה ל-JSON שקיבלנו מהענן.

(2) יצירת אובייקט של GSON והמרה של ה-JSONString שקיבלנו מהענן למערך Person!

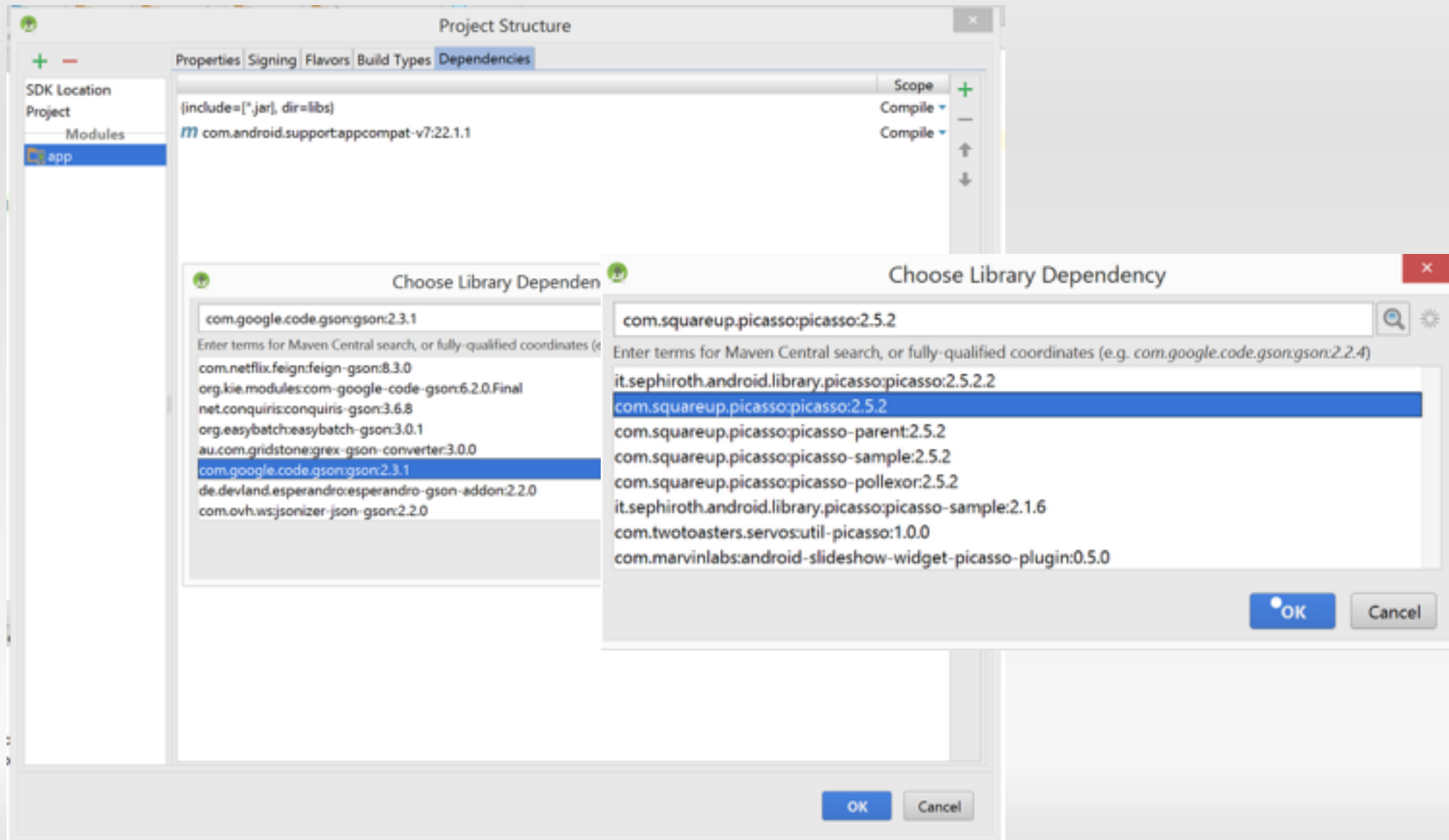
Import GSON

Right click on app module and goto module settings:



Import GSON and Picasso

Choose + and Library dependency. In the box search for gson:



Also add Picasso library dependency.

Using GSON with TMDB

```
<uses-permission android:name="ANDROID.PERMISSION.INTERNET" />
```

```
@Override
```

```
protected Movie[] doInBackground(String... params) {  
    String json = HTTPHandler.getUrl(params[0]);  
    String moviesJson = "";  
    Gson gson = new Gson();  
    try {  
        JSONObject o = new JSONObject(json);  
        JSONArray results = o.getJSONArray("results");  
        Log.d("Quiz", moviesJson = results.toString());  
    } catch (JSONException e) {  
        e.printStackTrace();  
    }  
    Movie[] movies = gson.fromJson(moviesJson, Movie[].class);  
    return movies;  
}
```

Picasso

ספריית קוד שמטרתה לטעון תמונות מהאינטרנט היישר לתוך ImageView
הספרייה מאפשרת הקטנת תמונות ו-CACHE.
אין צורך להוריד את התמונה ב-AsyncTask, להקטין אותה ולטעון ל-ImageView - הכל בשורה אחת!

```
ImageView ivPoster = (ImageView) findViewById(R.id.ivPoster);  
Picasso.with(this).load("http://image.tmdb.org/t/p/w150/t90Y3G8UGQp0f0DrP60wRu9gfrH.jpg/?  
api_key=b3b1492d3e91e9f9403a2989f3031b0c").into(ivPoster);
```


HTTP Request

- Networking must be made off the UI Thread.
- In order to access the Internet we need a uses-permission.

- We need a URL in order to open a Connection:

```
url = new URL("https://www.google.co.il/");  
con = (HttpURLConnection) url.openConnection();
```

- From the connection we get the result input stream:

```
inputStream = con.getInputStream();
```

HTTP Request

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

```
public String getUrl(String address) {  
    BufferedReader reader = null;  
    String total = "";  
    try {  
        URL url = new URL(address);  
        HttpURLConnection con = (HttpURLConnection) url.openConnection();  
        InputStream inputStream = con.getInputStream();  
        reader = new BufferedReader(new InputStreamReader(inputStream));  
        String line = "";  
        while ((line = reader.readLine()) != null) {  
            total += line;  
        }  
    } catch (MalformedURLException e) {  
        e.printStackTrace();  
    } catch (IOException e) {  
        e.printStackTrace();  
    } finally {  
        if (reader != null)  
            try {  
                reader.close();  
            } catch (IOException e) {  
                e.printStackTrace();  
            }  
    }  
    return total;  
}
```

Do We need A
Permission?
INTERNET

Do we need a Thread/
AsyncTask?

HTTP Request

Doing it Async

```
public class AsyncUrl extends AsyncTask<String ,Integer,String> {

    private final TextView textView;

    public AsyncUrl(TextView textView){
        this.textView = textView;
    }
    @Override
    protected String doInBackground(String... params) {
        return getUrl(params[0]);
    }

    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
        Log.d("Ness", result);
        textView.setText(result);
    }

    private String getUrl(String address) {
        ...see The method body in the previous slide
    }
}
```

HTTP Request

Calling our AsyncUrl task

```
public class MainActivity extends ActionBarActivity {  
    TextView mTextView;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        mTextView = (TextView) findViewById(R.id.tvWeb);  
        new AsyncUrl(mTextView).execute("https://developer.android.com");  
    }  
}
```

שמירה וטעינת מידע - SharedPreferences

שמירת מידע מתחלקת ל-2 חלקים - שמירת המידע וטעינתו.
כדי לשמור ולטעון נתונים נעשה שימוש במחלקה SharedPreferences.

לרוב נרצה להגדיר את המופע של SharedPreferences ברמת המחלקה כך
שיהיה נגיש מכל רחבי ה-Activity.

הגדרת המופע נעשית ע"י המתודה `getSharedPreferences` שמקבלת שני
פרמטרים - שם שתחתיו נרצה לשמור ופרמטר שני מסוג `int` שמייצג הרשאות
גישה - `Context.MODE_PRIVATE` ייקבע שהקובץ יהיה נגיש לאפליקציה בלבד.

```
pref = getSharedPreferences("Notes", Context.MODE_PRIVATE);
```

```
public class MainActivity extends ActionBarActivity {  
    SharedPreferences pref;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        pref = getSharedPreferences("Notes", Context.MODE_PRIVATE);  
    }  
}
```

שמירה וטעינת מידע - SharedPreferences

לאחר שהגדרנו את המופע נכתוב מתודות לשמירה וטעינה.

מתודת שמירה:

עבור שמירת מידע נשתמש באובייקט מהמחלקה Editor יצירת המופע מבוצעת

ע"י `pref.edit()`

כעת נשתמש במופע כדי לשמור תוכן ע"י המתודה:

`putString(String key, String value)`

המתודה מקבלת שני פרמטרים: הראשון - שם הערך שאנו רוצים לשמור.

והפרמטר השני - הערך.

לבסוף תבוצע השמירה בפועל ע"י `editor.commit()`.

מתודת השמירה:

```
private void saveData(String data) {  
    Editor editor = pref.edit();  
    editor.putString("Score", data);  
    editor.commit();  
}
```

מתודות דומות לשמירת סוגי משתנים נוספים -

`putString, putBoolean, putFloat, putInt, putLong, putStringSet`

שמירה וטעינת מידע - SharedPreferences

לטעינת מידע נשתמש במופע של SharedPreferences ובמתודה getString המתודה מקבלת שני פרמטרים, הראשון - שם הערך שאנו מעוניינים לטעון והפרמטר השני - ערך ברירת מחדל למקרה שלא נמצא הערך המבוקש. מתודת טעינה:

```
private String loadData() {  
    String data = pref.getString("Score", "0");  
    return data;  
}
```

מבחן מסכם - אנדרואיד

כלים בשימוש:

AsyncTask

Json/Gson

Picasso for async image loading

RecyclerView

Activities and intents.

SharedPreferences

Internet Permission

מבחן מסכם - אנדרואיד



Dawn of the Planet of the Apes

Rating: 8.3

Action, Drama, Sci-Fi

2014



District 9

Rating: 8.0

Action, Sci-Fi, Thriller

2009



Transformers: Age of Extinction

Rating: 6.3

Action, Adventure, Sci-Fi

2014



X-Men: Days of Future Past

Rating: 8.4

Action, Sci-Fi, Thriller

2014

מסך ראשון - הסרטים הפופולריים לשנת 2017:
במסך זה יש להציג RecyclerView של כל הסרטים
מהשרת בנתיב:

[989f3031b0c&primary_release_year=2017&sort_by=popularity.desc](#)

מבחן מסכם - אנדרואיד



Jack Ryan: Shadow Recruit

Critics Score: 58%

Audience Score: 65%

Chris Pine, Keira Knightley, Kevin Costner, Peter Andersson, Kenneth Branagh

Consensus: It doesn't reinvent the action-thriller wheel, but Jack Ryan: Shadow Recruit offers a sleek, reasonably diverting reboot for a long-dormant franchise.

Synopsis: Based on the character created by bestselling author Tom Clancy, "Jack Ryan" is a global action thriller set in the present day. This original story follows a young Jack (Chris Pine) as he uncovers a financial terrorist plot. The story follows him from 9/11, through his tour of duty in Afghanistan, which scarred him forever, and into his early days in the Financial Intelligence Unit of the modern CIA where he becomes an analyst, under the guardianship of his handler, Harper (Kevin Costner). When Ryan believes he's uncovered a Russian plot to collapse the United

מסך שני - מסך פרטי הסרט -

בכל לחיצה על פריט ברשימת הסרטים נעבור למסך המציג את כל פרטי הסרט

https://api.themoviedb.org/3/movie/135397?api_key=b3b1492d3e91e9f9403a2989f3031b0c

במסך זה נציג תמונה מוגדלת וכן את תקציר הסרט. ואת מדד ה-popularity של הסרט.

מבחן מסכם - אנדרואיד

בחלק זה עליכם לבצע קריאה לאפליקציית חייגן, לאפליקציית דוא"ל וכן לאפליקציית GPS.

מבחן מסכם - אנדרואיד

בחלק זה עליכם להוסיף לוגין מקומי לאפליקציה.

הלוגין יהיה מבוסס על SharedPreferences.

הלוגין הינו לוגין מפושט, שמבקש מהמשתמש שם משתמש בלבד.

ללא מסך הרשמה ומסך התחברות.

הלוגין המפושט עובד כך:

בפעם הראשונה שהמשתמש מפעיל את האפליקציה, האפליקציה תבקש מהמשתמש כינוי.

לאחר שהמשתמש הזין כינוי, האפליקציה תציג למשתמש הודעת 'Welcome, John Doe'.

בכל פעם נוספת שהמשתמש יפעיל את האפליקציה. כאשר John Doe זה הכינוי שהמשתמש הזין.