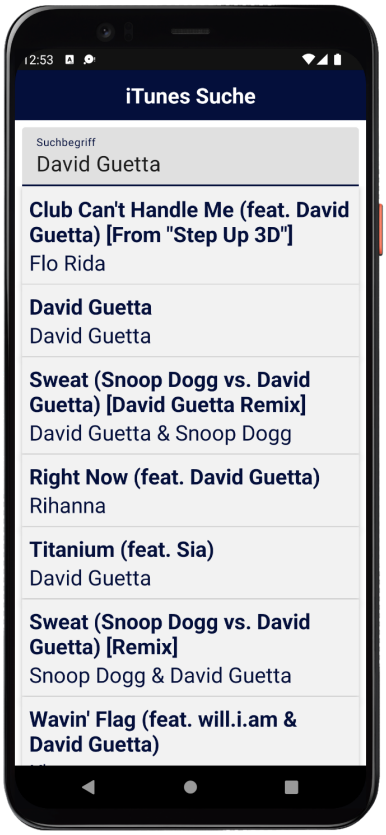
Kotlin 4.4 - API Wiederholung

Aufgaben

Hinweis: Zu bearbeiten ist Aufgabe 1.

1. iTUNES SUCHE - API Calls

In dieser Aufgabe programmierst du eine App für die Suche von Inhalten in der iTunes Mediathek. Dafür benutzt du die offizielle Apple iTunes API.



- Die Dokumentation für die iTunes findest du hier: iTunesAPI
- So sieht die JSON Antwort einer Suchanfrage aus, das Ergebnis enthält viele Informationen:


```
{
 "resultCount":50,
 "results": [
        {
                "wrapperType":"track",
                "kind":"song",
                "artistId":255303209,
                "collectionId":378650465,
                "trackId":378650466,
                "artistName":"Flo Rida",
                "collectionName":"Club Can't Handle Me - Single",
                "trackName":"Club Can't Handle Me (feat. David Guetta) [From \"Step Up 3D\"]",
                "collectionCensoredName":"Club Can't Handle Me - Single",
                "trackCensoredName":"Club Can't Handle Me (feat. David Guetta) [From \"Step Up 3D\"]",
                "artistViewUrl":"https://music.apple.com/us/artist/flo-rida/255303209?uo=4",
                "collectionViewUrl":"https://music.apple.com/us/album/club-cant-handle-me-feat-david-guetta...",
                "trackViewUrl":"https://music.apple.com/us/album/club-cant-handle-me-feat-david-guetta-from...",
                "previewUrl":"https://audio-ssl.itunes.apple.com/itunes-assets/AudioPreview122/v4/27/f5/b9/...",
                "artworkUrl30":"https://is2-ssl.mzstatic.com/image/thumb/Music125/v4/c5/10/ba/c510baab-02c1...",
                "artworkUrl60":"https://is2-ssl.mzstatic.com/image/thumb/Music125/v4/c5/10/ba/c510baab-02c1...",
                "artworkUrl100":"https://is2-ssl.mzstatic.com/image/thumb/Music125/v4/c5/10/ba/c510baab-02c...",
                "collectionPrice":1.98,
                "trackPrice":0.69,
                "releaseDate":"2010-06-28T07:00:00Z",
                "collectionExplicitness":"notExplicit",
                "trackExplicitness":"notExplicit",
                "discCount":1,
                "discNumber":1,
                "trackCount":2, "trackNumber":1,
                "trackTimeMillis":232616,
                "country":"USA",
                "currency":"USD",
                "primaryGenreName":"Hip-Hop/Rap",
                "isStreamable":true
        },
                ...
        ]
}
```

- Öffne das Projekt “iTunes Suche”
- Das Projekt ist bereits vorbereitet. Es enthält schon alle Dependencies und Voreinstellungen. Das Layout und eine leere Recyclerview sind schon eingebaut.
- Überlege dir, welche Informationen aus der Antwort du einbauen willst und passe das Layout `list_item_result` so an, dass alle Informationen angezeigt werden können. Vergib den XML Elementen jeweils eine eindeutige ID
- Erstelle im Package `datamodels` die `data` Klassen um die Liste an Songs aus der API in Kotlin darzustellen. In den Klassenvariablen sollten alle Variablen enthalten sein, die du brauchst.
  Hinweis: Wie bei Aufgabe 4.2 brauchst du erst eine andere Datenklasse um auf results zuzugreifen.
- Programmiere den API-Service `iTunesApiService`
  Die API URL mit Beispielsuche “Pancakes” lautet:
  ```
  "https://itunes.apple.com/search?term=Pancakes"
  ```
  - Definiere die `BASE_URL`
  - Erstelle einen `Moshi` Konverter
  - Erstelle ein `Retrofit` Objekt
  - Erstelle ein `interface` `SearchApiService` in dem eine Funktion `getResults` mit einer `GET` Annotation enthalten ist. Die Funktion soll nach dem Suchbegriff suchen, der ihr als Parameter übergeben wird
  - Erstelle als Zugriffspunkt ein `object` `SearchApi`, in dem die Variable `retrofitService` enthalten ist

  Hinweis:
  ```
  /**
  * Für die URL "https://api.mysite.de/search?suchbegriff=Pancakes"
  * wäre die BASE_URL "https://api.mysite.de/"
  * Der Abruf der Suchergebnisse mit übergebenen Suchbegriffen
  * (und nicht jedes Mal "Pancakes") sähe so aus:
  */
  @GET("search")
  suspend fun getResults(@Query("suchbegriff") suchbegriff: String): SearchResult
  ```
- Richte die Klasse `AppRepository` ein. Hier sollen die Informationen für den Rest der App per API Call geholt und bereitgestellt werden.
  - Übergebe im Konstruktor ein Objekt von `SearchApi`
  - Erstelle ein LiveData Variable, in der die Liste aus dem API-Call gespeichert wird

    - Programmiere eine `suspend` Funktion `getResults`, die den Suchbegriff als Parameter übergeben bekommt und versucht, den API Call über den Retrofit Service der API mit dem Suchbegriff auszuführen. Falls der Versuch klappt, soll das Ergebnis in die LiveData Variable gespeichert werden, falls nicht, soll die Fehlermeldung in der Logcat ausgegeben werden.
- Erstelle eine veränderliche Live Data Variable im `SearchViewModel`. Hier soll der `text` aus dem `TextInputEditText` eingespeichert werden. Gehe in das Layout `fragment_search`. Binde hier das ViewModel als `variable` ein und weise der Live Data Variablen aus dem ViewModel den `text` des `TextInputEditText` zu.

  Hinweis: Vergiss nicht, in der Klasse `SearchFragment` das `viewModel` mit `binding` der ViewModel Variablen im XML Layout zuzuweisen
- Speichere in ViewModel die Suchergebnisse aus dem Repository in einer eigenen Variablen `results`, damit wir die Liste an Ergebnissen beobachten können
- Programmiere die Funktion `loadData`, die einen Suchbegriff als Parameter bekommt und in einer Coroutine den API-Call über das Repository startet.
- In der Klasse `SearchFragment` beobachten wir die beiden Variablen aus dem ViewModel:
    - Falls sich der Suchbegriff im Input Text ändert, soll ein neuer API-Call über das ViewModel gestartet werden
    - Falls der API Call erfolgreich ist und die Ergebnisliste ändert, soll der RecyclerView ein neuer Adapter verpasst werden, dieser muss in der Klasse `SearchAdapter` aber noch fertig eingerichtet werden:
        - Der Adapter braucht noch eine private Variable `dataset` im Kostruktor
        - Die Größe der Liste `dataset` soll in der Funktion `getItemCount` zurückgegeben werden
        - Außerdem sollen die Elemente des ViewHolder richtig mit Informationen befüllt und angezeigt werden
- Die Such App sollte nun funktionieren! Führe sie aus und führe ein paar Suchanfragen durch
- Pass die App an, bis sie optisch und Feature technisch deinen Wünschen entspricht!

Viel Erfolg!