Project 4 Task 1 – iTunes Songs App

Author – Annamalai Kathirkamanathan

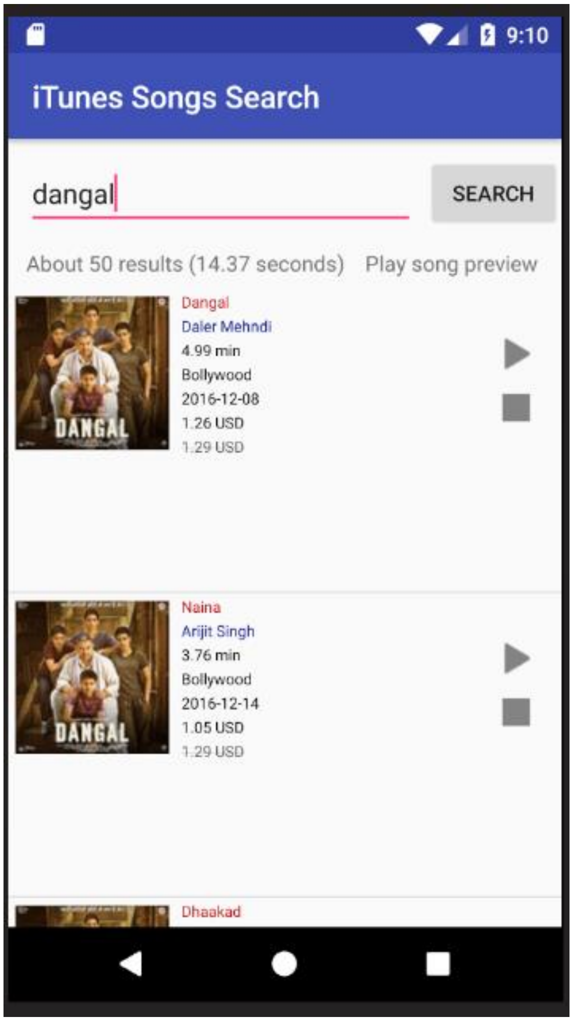
Description:

The application takes a search string (song or artist or movie name) from the user and uses it to fetch song information from the server which in-turn connects to the Itunes API and displays a list of all the songs that matches the search.

Here is how my app meets the task requirements:

1. Implement a native android app
   The name of my native Android application project in Android Studio is: Project4Android
   1.1 Has at least three different kinds of views in your Layout (TextView, EditText, ImageView, etc.)

   My application uses text view, edit text, button, list view – with image view, text view and button like image view of play and stop. Here is a screenshot of the layout



dangal

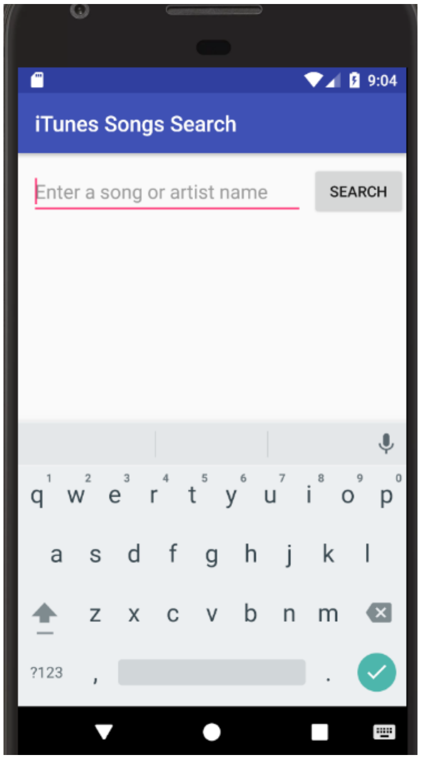
About 50 results (14.37 seconds) Play song preview

Dangal
Daler Mehndi
4.99 min
Bollywood
2016-12-08
1.26 USD
1.29 USD

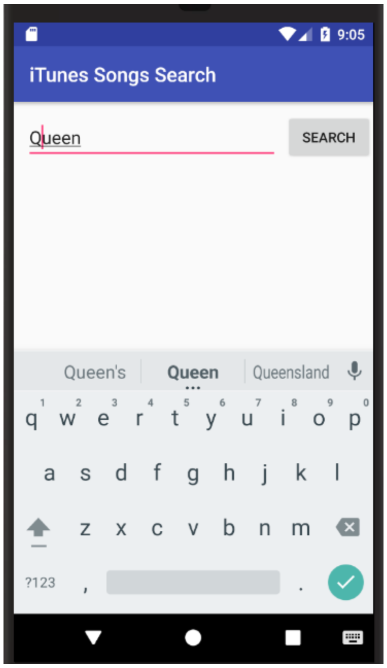
Naina
Arijit Singh
3.76 min
Bollywood
2016-12-14
1.05 USD
1.29 USD

Dhaakad

The screenshot of the layout before a song has been fetched or searched:



1.2 Requires input from the user:



Queen

1.3 Makes an HTTP request (using an appropriate HTTP method) to your web service

The app does a HTTP GET request in GetInfo.java. The HTTP request is: “https://shrouded-meadow-27857.herokuapp.com/getResults//”+input (task 1), where input in the app is the user’s search term (e.g. Queen). The app converts it to Queen,song because the default type is “song”

The search method makes a request to the web server which parses the search term and the type and send it to the Itunes API using: "https://itunes.apple.com/search?term="+ search+"&entity=" + entity. Gets back a JSON of the results and sends only a selected set of information as JSON to the android studio to be displayed in the app.

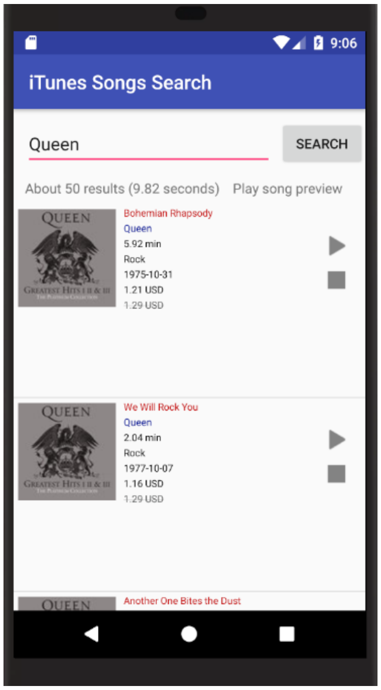
1.4 Receives and parses an XML or JSON formatted reply from your web service

The example of a JSON reply from the API (has more information) and to the app is:

{"resultCount":2,"results":[{"artworkUrl100":"http://is4.mzstatic.com/image/thumb/Music7/v4/f5/63/f5/f563f527-e021-c320-1e62-7ea463d881ae/source/100x100bb.jpg","trackTimeMillis":4.45,"trackPrice":1.29,"previewUrl":"https://audio-ssl.itunes.apple.com/apple-assets-us-std-000001/Music1/v4/6d/5b/0c/6d5b0c38-35fb-ed1f-a0c4-c6a6daacd572/mzaf_3057992870330552589.plus.aac.p.m4a","releaseDate":"2014-05-12","discountPrice":1.14,"currency":"USD","artistName":"Clean Bandit","trackName":"Dust Clears (feat. Noonie Bao)","primaryGenreName":"Dance"},{"artworkUrl100":"http://is1.mzstatic.com/image/thumb/Music/v4/17/2c/16/172c1667-e0e5-f411-1acf-35e4d6563d0a/source/100x100bb.jpg","trackTimeMillis":6.26,"trackPrice":0.99,"previewUrl":"https://audio-ssl.itunes.apple.com/apple-assets-us-std-000001/Music/d7/95/26/mzm.dbqgxvpw.aac.p.m4a","releaseDate":"2007-10-16","discountPrice":0.9,"currency":"USD","artistName":"Amir Baghiri & Wee Bandits","trackName":"Clear Blue Sea","primaryGenreName":"Electronic"}]}

1.5 Displays new information to the user

Here is a screenshot after the songs information has been returned



iTunes Songs Search

Queen

About 50 results (9.82 seconds)

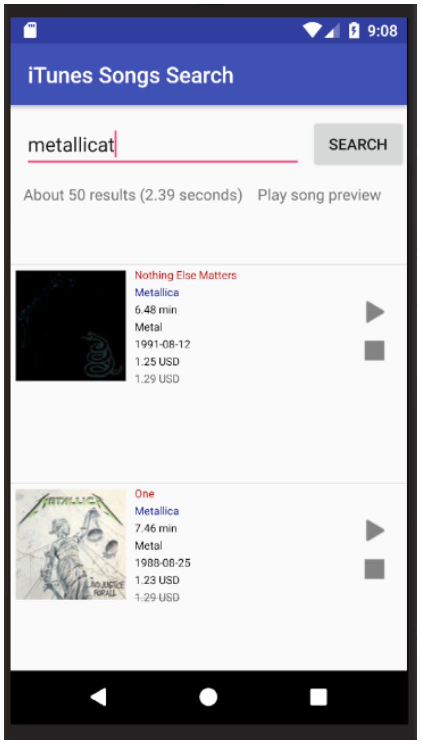
Bohemian Rhapsody
Queen
5.92 min
Rock
1975-10-31
1.21 USD
~~1.29 USD~~

We Will Rock You
Queen
2.04 min
Rock
1977-10-07
1.16 USD
~~1.29 USD~~

Another One Bites the Dust

1.6 Is repeatable (I.e. the user can repeatedly reuse the application without restarting it.)

The user can type in another search term of a song/artist/movie and hit search. Here is an example of typing in “Metallica” to the search



metallicat

About 50 results (2.39 seconds)

Nothing Else Matters
Metallica
6.48 min
Metal
1991-08-12
1.25 USD
1.29 USD

One
Metallica
7.46 min
Metal
1988-08-25
1.23 USD
~~1.29 USD~~

2. Implement a web service, deployed to Heroku

The URL of my web service deployed to Heroku is: for task 1 its - https://shrouded-meadow-27857.herokuapp.com/ and for task 2 its - https://frozen-woodland-67953.herokuapp.com/

The project directory names are Project4Task1 and Project4Task2 respectively.

2.1 Using an HttpServlet to implement a simple (can be a single path) API

In my web app project task 1:
The model is – Project4Task2Model
Controller is - Project4Task2Servlet

In my web app project task 2:
The model is – Project4Task2Model
View is – dashboard.jsp
Controller is - Project4Task2Servlet

2.2 Receives an HTTP request from the native Android application

Project4Task2Servlet.java receives a HTTP GET request with the argument “input” which has both the search term and type separated by comma. It passes both the search and type string on to the model to fetch a JSON list of songs with the search term.

2.3 Executes business logic appropriate to your application. This includes fetching XML or JSON information from some 3rd party API and processing the response.

Project4Task2Model.java makes a HTTP request to:
https://itunes.apple.com/search?term=search&entity=song (where search = search input)

2.4 Replies to the Android application with an XML or JSON formatted response

Project4Task2Servlet get specific JSON information extracted from API, converts to a JSON object and send it to android studio

{"resultCount":2,"results":[{"artworkUrl100":"http://is4.mzstatic.com/image/thumb/Music7/v4/f5/63/f5/f563f527-e021-c320-1e62-7ea463d881ae/source/100x100bb.jpg","trackTimeMillis":4.45,"trackPrice":1.29,"previewUrl":"https://audio-ssl.itunes.apple.com/apple-assets-us-std-000001/Music1/v4/6d/5b/0c/6d5b0c38-35fb-ed1f-a0c4-c6a6daacd572/mzaf_3057992870330552589.plus.aac.p.m4a","releaseDate":"2014-05-12","discountPrice":1.14,"currency":"USD","artistName":"Clean Bandit","trackName":"Dust Clears (feat. Noonie Bao)","primaryGenreName":"Dance"},{"artworkUrl100":"http://is1.mzstatic.com/image/thumb/Music/v4/17/2c/16/172c1667-e0e5-f411-1acf-35e4d6563d0a/source/100x100bb.jpg","trackTimeMillis":6.26,"trackPrice":0.99,"previewUrl":"https://audio-ssl.itunes.apple.com/apple-assets-us-std-000001/Music/d7/95/26/mzm.dbqgxvpw.aac.p.m4a","releaseDate":"2007-10-16","discountPrice":0.9,"currency":"USD","artistName":"Amir Baghiri & Wee Bandits","trackName":"Clear Blue Sea","primaryGenreName":"Electronic"}]}