Project 4 Task 2 – iTunes Songs App

Author – Annamalai Kathirkamanathan

Description:

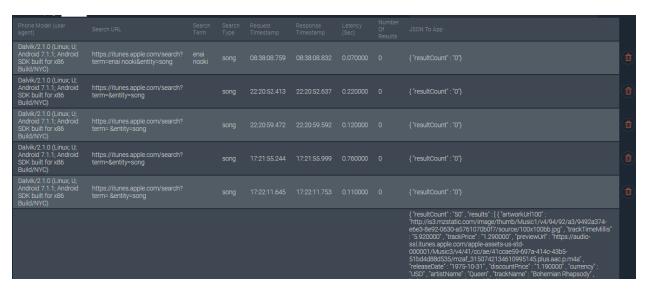
This task includes task 1 where it takes a search string (song or artist or movie name) from the user and uses it to fetch song information from the server which in-turn connects to the Itunes API and displays a list of all the songs that matches the search. In task 2 there are two different URL within the servlet "/getResults/song,entity" and "/getDashboard". The first URL returns JSON results that is sent to the app and the second URL displays the dashboard which shows the operational analytics and the logs fetched from mlab's mongodb.

1. Log useful information

6 pieces of information logged are:

- a. Phone model (user agent) the phone that makes the request
- b. Search URL the URL used to request to the API
- c. Search term the search string entered by the user
- d. Search type this is always song, but can be changes and hence logging this info
- e. Request timestamp the time of request
- f. Response timestamp the time of response
- g. Latency (Sec) the time difference between response and request in seconds
- h. Number of results the total number of results of songs sent by the API
- i. JSON to app the JSON results that is sent to the app

The screenshot from the mlab site -



2. Store the log information in a database

The web service connects to the mlabs mongo dB using:
"mongodb://"+username+":"+password+"@ds251245.mlab.com:51245/project4database"
Where the username = database and password is also database

It stores information using the doPost method in Project4Task2Servlet.java by adding information to the documents in the collection

It retrieves information from the database using the doGet method when "/getDashboard" is called. It internally uses doMongodbfetch method in Project4Task2Model.java to connect to the database and retrieve logs and calculate analytics which is displayed by the dashboard.jsp on a browser.

- 3. Display operation analytics and full logs on a web-based dashboard
 - 3.1 A unique URL addresses a web interface dashboard for the web service

/getDashboard is used by Project4Task2Servlet.java to display the web interface dashboard 3.2 The dashboard displays at least 3 interesting operations analytics.

The screenshot from dashboard is:

	Operational Analytics			
Field	Value			
Average count of results	37.1			
Average time taken (sec)	0.43			
Average discount per song (\$)	0.131			
Searches in descending order	[mersal, queen, taylor, enai nooki, metallica, kadhal kanmani, enai noki]			

Dashboard shows:

- a. Average count of results the average number of results per search
- b. Average time taken (sec) the average latency of the app per search
- c. Average discount per song (\$) the average difference between actual price and discount price that my application provides per song
- d. Search in descending order the list of all the searches in descending order of its count
- 3.3 The dashboard displays the full logs.

The screenshot from dashboard is

	Logo								
Phone Model (User Agent)	Search URL	Search Term	Search Type	Request Timestamp	Response Timestamp	Latency (sec)	Number of results	JSON output from API	
	https://itunes.apple.com/search? term=enai nooki&entity=song	enai nooki	song	08:38:08.759	08:38:08.832	0.07	0	("resultCount":0}	
								["resultCount":50, "results" [["artworkU100", "http://is3.mzstatic.com/image/thumb/Music1/v4/94/92/a/0630-a57610700077000re100x100bb.jpg", "trackTimeMillis":592, "trackPrice":129, "previewUrl" "http://is1mus.pule.com/apple-assets-us-sid-000001/Music3/v4/41/ce4-de14cce59-96/3-414-c43b5-31bd4688d35/mzaf_3150/42134610995145, plus.ace_pm.4a", "releaseDate"; "1975-10-31", "discountFriee":11.9, "currency", "USD", "artisName", "Queen", "trackName", "Bohemian Rhapsody", "primaryGenreName", "Rock"), ("artworkU10") "http://isa.ustatic.com/image/thumb/Music1/v4/94/92/a3/9492a374-e6e3-8e92-0630.a5761070b07/source100x100bb.jpg", "trackTimeMillis":2.04, "trackPrice":1.29, "previewUrl": "https://au	

The logs in dashboard shows all the 9 information that was discussed in point 1 (log useful information)

4. Deploy the web service to Heroku

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

The web service of project task 2 has:

The model is – Project4Task2Model View is – dashboard.jsp Controller is - Project4Task2Servlet

The dashboard URL is - https://frozen-woodland-67953.herokuapp.com/getDashboard

The search URL would be -

https://frozen-woodland-67953.herokuapp.com/getResults/clear bandit,song