

Phone Model (user agent)	Search URL	Search Term	Search Type	Request Timestamp	Response Timestamp	Latency (Sec)	Number Of Results	JSON To App
Dalvik/2.1.0 (Linux; U; Android 7.1.1; Android SDK built for x86 Build/NYC)	https://itunes.apple.com/search?term=enai nooki&entity=song	enai nooki	song	08:38:08.759	08:38:08.832	0.070000	0	{ "resultCount": "0" }
Dalvik/2.1.0 (Linux; U; Android 7.1.1; Android SDK built for x86 Build/NYC)	https://itunes.apple.com/search?term=&entity=song		song	22:20:52.413	22:20:52.637	0.220000	0	{ "resultCount": "0" }
Dalvik/2.1.0 (Linux; U; Android 7.1.1; Android SDK built for x86 Build/NYC)	https://itunes.apple.com/search?term= &entity=song		song	22:20:59.472	22:20:59.592	0.120000	0	{ "resultCount": "0" }
Dalvik/2.1.0 (Linux; U; Android 7.1.1; Android SDK built for x86 Build/NYC)	https://itunes.apple.com/search?term=&entity=song		song	17:21:55.244	17:21:55.999	0.760000	0	{ "resultCount": "0" }
Dalvik/2.1.0 (Linux; U; Android 7.1.1; Android SDK built for x86 Build/NYC)	https://itunes.apple.com/search?term= &entity=song		song	17:22:11.645	17:22:11.753	0.110000	0	{ "resultCount": "0" }
								{ "resultCount": "50", "results": [{ "artworkUrl100": "http://s3.mzstatic.com/image/thumb/Music1/v4/94/92/a3/9492a374-e6e3-9e92-0630-a5761070b0f7/source/100x100bb.jpg", "trackTimeMillis": "592000", "trackPrice": "1.290000", "previewUrl": "https://audio-ssl.itunes.apple.com/apple-assets-us-std-000001/Music3/v4/41/cc/ae/41ccae59-697a-414c-43b5-51bd4d88d535/mzaf_3150742134610995145.plus.aac.p.m4a", "releaseDate": "1975-10-31", "discountPrice": "1.190000", "currency": "USD", "artistName": "Queen", "trackName": "Bohemian Rhapsody" }

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:

- Average count of results – the average number of results per search
- Average time taken (sec) – the average latency of the app per search
- Average discount per song (\$) – the average difference between actual price and discount price that my application provides per song
- 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

Phone Model (User Agent)	Search URL	Search Term	Search Type	Request Timestamp	Response Timestamp	Latency (sec)	Number of results	JSON output from API
Dalvik/2.1.0 (Linux; U; Android 7.1.1; Android SDK built for x86 Build/NYC)	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":[{"artworkUrl100":"http://is3.mzstatic.com/image/thumb/Music1/v4/94/92/a1/0630-a5761070b0f7/source/100x100bb.jpg","trackTimeMillis":5.92,"trackPrice":1.29,"previewUrl":"http://ssl.itunes.apple.com/apple-assets-us-std-000001/Music3/v4/41/cc/ae/ae41ccae59-697a-414e-43b5-51bd4d88d535/mzaf_3150742134610995145.plus.aac.p.m4a","releaseDate":"1975-10-31","discountPrice":1.19,"currency":"USD","artistName":"Queen","trackName":"Bohemian Rhapsody","primaryGenreName":"Rock"}, {"artworkUrl100":"http://is3.mzstatic.com/image/thumb/Music1/v4/94/92/a3/9492a374-e6e3-8e92-0630-a5761070b0f7/source/100x100bb.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>