# Project 4 Task 2 – Book Master App (with Dashboard)
by Xiaotong Luo (xluo2)

**Description:**
My application will take a search string from the user and then uses it to fetch information from Google Books and display a list (only include five books) of relevant books' information, including a thumbnail picture, book title, author, publication time, and a hyperlink that can direct the user to the home page of the book for more information.

My application can also log useful information in MongoDB and display a dashboard for interesting metrics and data logs.

Here is how my application meets the task requirements.

## 1. Log useful information
My web project logs 7 pieces of information:

**Information about the request from the mobile phone:** the timestamp when my web application receives the user request, the searchTerm from the user

**Information about the request to Google Books API:** the timestamp when my web application sends the request

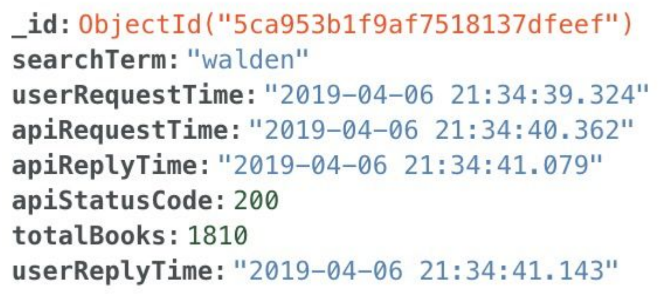
**Information about the reply from Google Books API:** the timestamp when my web application receives the reply, the number of total books for that search in results from Google Books API, the status code of the response

**Information about the reply to the mobile phone:** the timestamp when my web application sends the reply

## 2. Store the log information in a database

My web project can connect to MongoDB and insert information (documents) into the collection, which is implemented in BookDB.java. The information retrieval is implemented in DashboardModel.java.

Here is a screenshot of an example of the document stored in MongoDB:



```
_id: ObjectId("5ca953b1f9af7518137dfeef")
searchTerm: "walden"
userRequestTime: "2019-04-06 21:34:39.324"
apiRequestTime: "2019-04-06 21:34:40.362"
apiReplyTime: "2019-04-06 21:34:41.079"
apiStatusCode: 200
totalBooks: 1810
userReplyTime: "2019-04-06 21:34:41.143"
```

## 3. Display operations analytics and full logs on a web-based dashboard

### 3.1. A unique URL addresses a web interface dashboard for the web service

https://arcane-bastion-60747.herokuapp.com/
or https://arcane-bastion-60747.herokuapp.com/Dashboard

In my web app project:

1) REST web service:
- **Model:** BookModel.java
- **Controller:** BookServlet.java

2) Dashboard:
- **Model:** DashboardModel.java
- **View:** dashboard.jsp
- **Controller:** Dashboard.java

### 3.2. The dashboard displays at least 3 interesting operations analytics

**The operations analytics displayed are:**

- Most Popular Search Term - the most frequent search term

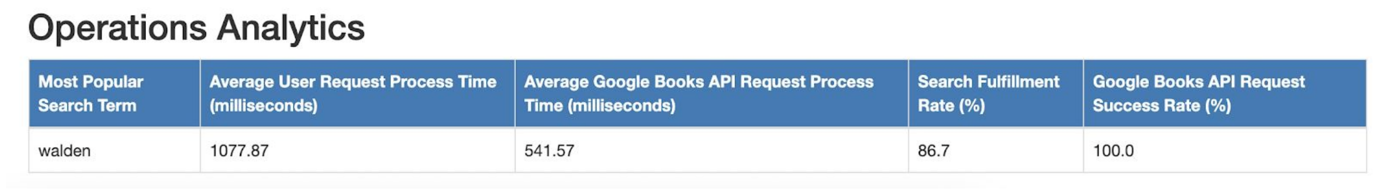
- Average User Request Process Time (milliseconds) - the time difference between the timestamp when my web application sends the reply to the mobile phone and the timestamp when my web application receives the request from the mobile phone user

- Average Google Books API Request Process Time (milliseconds) - the time difference between the timestamp when my web application receives the reply from Google Books API and the timestamp when my web application sends the request

- Search Fulfillment Rate (%) - the percentage of the mobile phone users' requests that are satisfied (failure is defined as 0 book found in results from Google Books API)

- Google Books API Request Success Rate (%) - the percentage of the successful requests to Google Books API (a request is successful if the response code is 200).

Here is a screenshot of the operations analytics table in my dashboard:



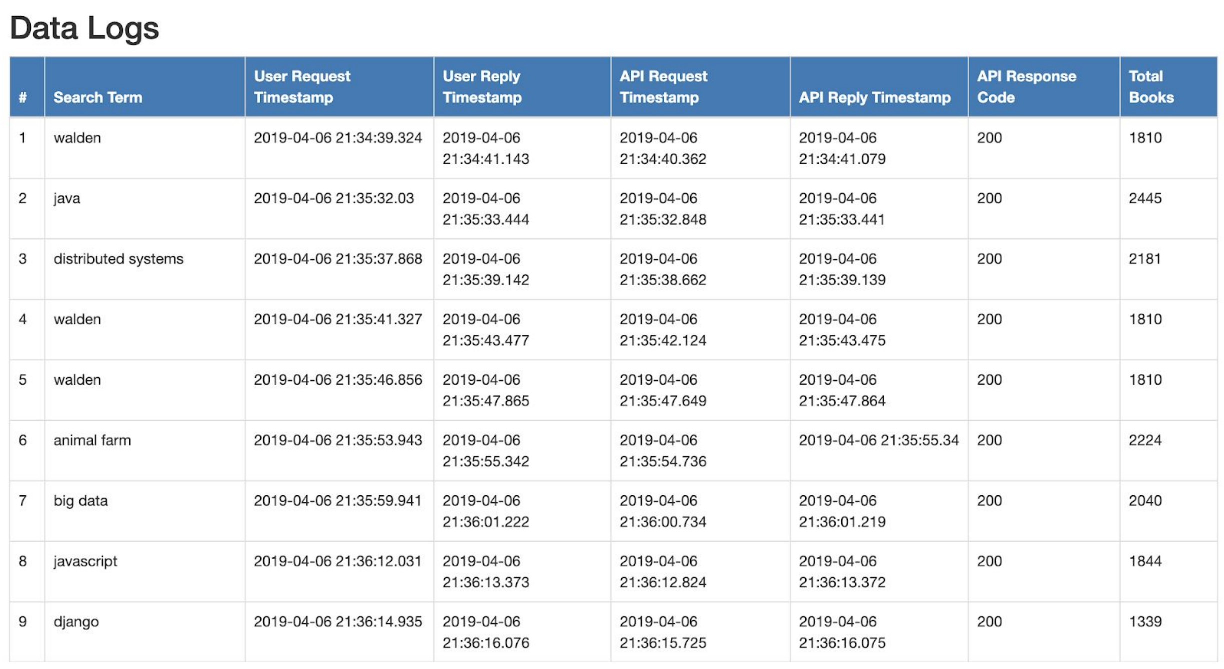
## Operations Analytics

| Most Popular Search Term | Average User Request Process Time (milliseconds) | Average Google Books API Request Process Time (milliseconds) | Search Fulfillment Rate (%) | Google Books API Request Success Rate (%) |
|---|---|---|---|---|
| walden | 1077.87 | 541.57 | 86.7 | 100.0 |

## 3.3. The dashboard displays the full logs

The data logs table displays the logs being stored for each mobile phone user interaction with my web service (a row contains the 7 pieces of information in a document).

Here is a screenshot of part of the data logs table in my dashboard:



Data Logs

| # | Search Term | User Request Timestamp | User Reply Timestamp | API Request Timestamp | API Reply Timestamp | API Response Code | Total Books |
|---|---|---|---|---|---|---|---|
| 1 | walden | 2019-04-06 21:34:39.324 | 2019-04-06 21:34:41.143 | 2019-04-06 21:34:40.362 | 2019-04-06 21:34:41.079 | 200 | 1810 |
| 2 | java | 2019-04-06 21:35:32.03 | 2019-04-06 21:35:33.444 | 2019-04-06 21:35:32.848 | 2019-04-06 21:35:33.441 | 200 | 2445 |
| 3 | distributed systems | 2019-04-06 21:35:37.868 | 2019-04-06 21:35:39.142 | 2019-04-06 21:35:38.662 | 2019-04-06 21:35:39.139 | 200 | 2181 |
| 4 | walden | 2019-04-06 21:35:41.327 | 2019-04-06 21:35:43.477 | 2019-04-06 21:35:42.124 | 2019-04-06 21:35:43.475 | 200 | 1810 |
| 5 | walden | 2019-04-06 21:35:46.856 | 2019-04-06 21:35:47.865 | 2019-04-06 21:35:47.649 | 2019-04-06 21:35:47.864 | 200 | 1810 |
| 6 | animal farm | 2019-04-06 21:35:53.943 | 2019-04-06 21:35:55.342 | 2019-04-06 21:35:54.736 | 2019-04-06 21:35:55.34 | 200 | 2224 |
| 7 | big data | 2019-04-06 21:35:59.941 | 2019-04-06 21:36:01.222 | 2019-04-06 21:36:00.734 | 2019-04-06 21:36:01.219 | 200 | 2040 |
| 8 | javascript | 2019-04-06 21:36:12.031 | 2019-04-06 21:36:13.373 | 2019-04-06 21:36:12.824 | 2019-04-06 21:36:13.372 | 200 | 1844 |
| 9 | django | 2019-04-06 21:36:14.935 | 2019-04-06 21:36:16.076 | 2019-04-06 21:36:15.725 | 2019-04-06 21:36:16.075 | 200 | 1339 |

## 4. Deploy the web service to Heroku

The address of my web application deployed at Heroku is:
https://arcane-bastion-60747.herokuapp.com/

The homepage URL is the dashboard URL.

The android phone will send HTTP GET request to
“https://arcane-bastion-60747.herokuapp.com/BookServlet/” + searchTerm

where searchTerm is the user's search term.

**Reference:**

Bootstrap Tables:
https://getbootstrap.com/docs/4.0/content/tables/

MongoDB documentation:
https://docs.mongodb.com/guides/server/insert/

Convert datetime string to timestamp type:
https://stackoverflow.com/questions/18915075/java-convert-string-to-timestamp