

Project4 Task1

API:

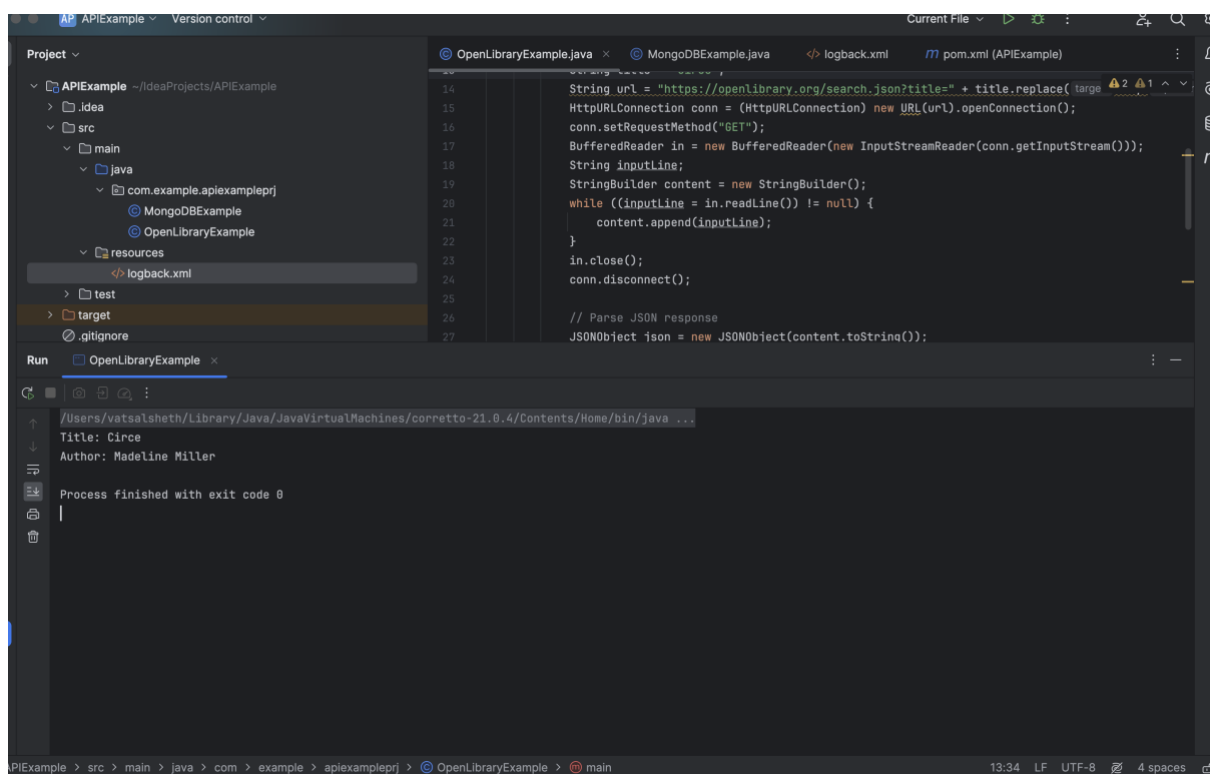
Open Library API

<https://openlibrary.org/developers/api>

App Description:

The app, called “Book Vault”, will allow users to search for books, get personalized recommendations based on reading history, and save favourite books to a wishlist. The backend, developed in Java, will fetch book data from the Open Library API, store user preferences in MongoDB Atlas, and serve data to the Android app.

Task 1b:

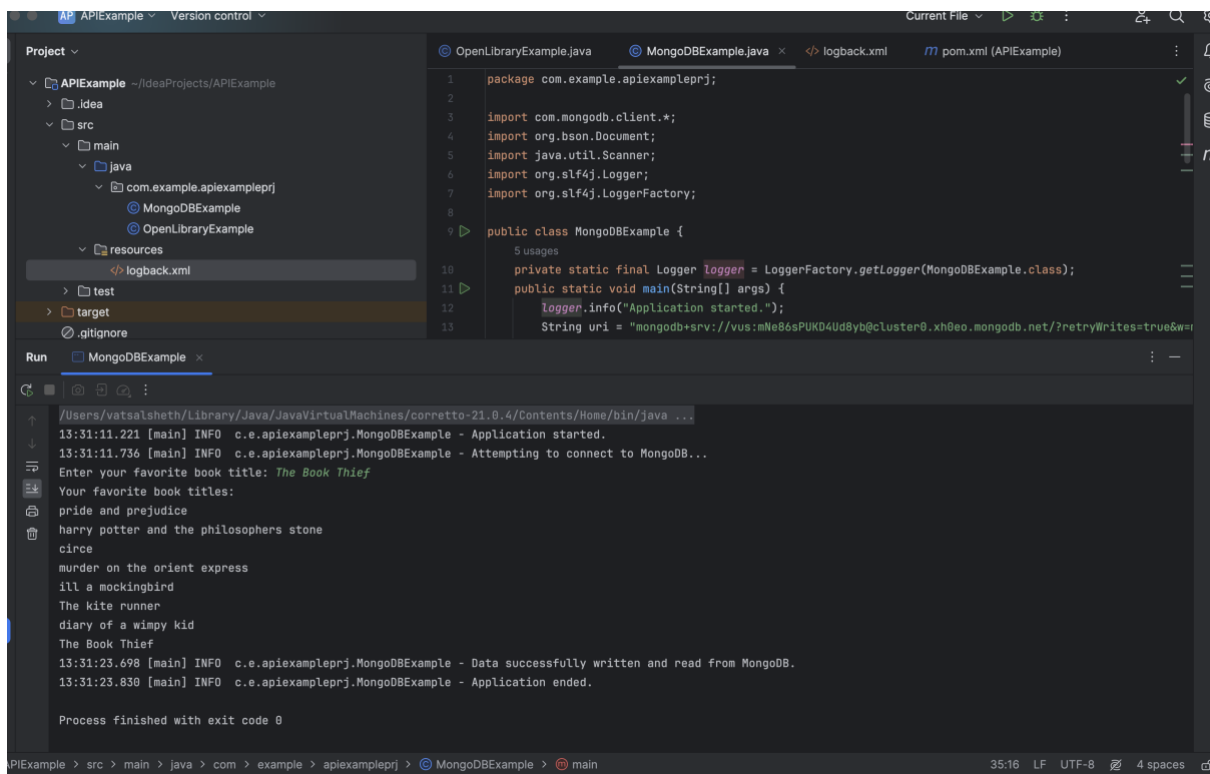


The screenshot shows an IDE with the following components:

- Project View:** A tree structure on the left showing the project layout. The 'src' directory contains 'main' and 'test'. The 'main' directory contains 'java' and 'resources'. The 'java' directory contains 'com.example.apilexampleprj', which includes 'MongoDBExample' and 'OpenLibraryExample'. The 'resources' directory contains 'logback.xml'.
- Code Editor:** The main window displays the code for 'OpenLibraryExample.java'. The code is as follows:

```
14 String url = "https://openlibrary.org/search.json?title=" + title.replace(" ", "%20");
15 HttpURLConnection conn = (HttpURLConnection) new URL(url).openConnection();
16 conn.setRequestMethod("GET");
17 BufferedReader in = new BufferedReader(new InputStreamReader(conn.getInputStream()));
18 String inputLine;
19 StringBuilder content = new StringBuilder();
20 while ((inputLine = in.readLine()) != null) {
21     content.append(inputLine);
22 }
23 in.close();
24 conn.disconnect();
25
26 // Parse JSON response
27 JSONObject ison = new JSONObject(content.toString());
```
- Run View:** The bottom panel shows the output of the run command. It displays the command path: `/Users/vatsalsheth/Library/Java/JavaVirtualMachines/corretto-21.0.4/Contents/Home/bin/java ...`. The output shows the title and author of a book: `Title: Circe` and `Author: Madeline Miller`. The process finished with exit code 0.

Task 2c:



The screenshot shows an IDE with the following components:

- Project View:** A tree view on the left showing the project structure. The 'resources' folder is selected, containing 'logback.xml'.
- Code Editor:** The main window displays the 'MongoDBExample.java' file. The code includes imports for MongoDB client, BSON, Scanner, and SLF4J logging. The 'main' method logs the application start, attempts to connect to MongoDB, prompts the user for a book title, reads the titles from the database, and logs the application end.
- Run Console:** The bottom panel shows the execution output. It displays the application start time, the attempt to connect to MongoDB, the user input 'The Book Thief', the list of book titles read from the database, and the application end time. The process finished with exit code 0.

```
package com.example.apilexampleprj;

import com.mongodb.client.*;
import org.bson.Document;
import java.util.Scanner;
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;

public class MongoDBExample {
    private static final Logger logger = LoggerFactory.getLogger(MongoDBExample.class);
    public static void main(String[] args) {
        logger.info("Application started.");
        String uri = "mongodb+srv://vus:mNe86sPUK04Ud8yb@cluster0.xh8eo.mongodb.net/?retryWrites=true&w=1";
    }
}
```

Run Console Output:

```
13:31:11.221 [main] INFO c.e.apilexampleprj.MongoDBExample - Application started.
13:31:11.736 [main] INFO c.e.apilexampleprj.MongoDBExample - Attempting to connect to MongoDB...
Enter your favorite book title: The Book Thief
Your favorite book titles:
pride and prejudice
harry potter and the philosophers stone
circe
murder on the orient express
ill a mockingbird
The kite runner
diary of a wimpy kid
The Book Thief
13:31:23.698 [main] INFO c.e.apilexampleprj.MongoDBExample - Data successfully written and read from MongoDB.
13:31:23.830 [main] INFO c.e.apilexampleprj.MongoDBExample - Application ended.

Process finished with exit code 0
```