Project4

by：JingBo Lin

AndrewID：jingbol

Description：

The mobile app will allow users to search for books by title using the Open Library API. It will display book details such as title, author, and first publish year. Users can save their search history, which will be stored in MongoDB. A cloud-based dashboard will analyze and display frequent search terms.

1：Android app code

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    // Bind views
    etSearch = findViewById(R.id.et_search);
    tvResult = findViewById(R.id.tv_result);
    Button btnSearch = findViewById(R.id.btn_search);

    // Button click event
    btnSearch.setOnClickListener(onClick(v) -> {
            String query = etSearch.getText().toString();
            if (query.isEmpty()) {
                Toast.makeText(MainActivity.this, "请输入书名", Toast.LENGTH_SHORT).show();
            } else {
                searchBooks(query);
            }
    });
}
```

```
private void searchBooks(String query) {
    // Construct request URL (replace with your actual backend address)
    String url = "https://effective-guide-7v54vp597xpr2xpgg-8080.app.github.dev/api/books?q=" + query;

    // Create request object
    Request request = new Request.Builder()
            .url(url)
            .build();

    // Send asynchronous requestSend asynchronous request

    client.newCall(request).enqueue(new Callback() {

        @Override
        public void onFailure(Call call, IOException e) {
            runOnUiThread(run() -> {
                    Toast.makeText(MainActivity.this, "Request Fail: " + e.getMessage(), Toast.LENGTH_SHORT).show();
            });
        }


        @Override
        public void onResponse(Call call, Response response) throws IOException {
            if (response.isSuccessful()) {
                String json = response.body().string();
                Book book = gson.fromJson(json, Book.class);
                runOnUiThread(run() -> {
                        tvResult.setText("Book Name: " + book.getTitle() + "\nAuthor: " + book.getAuthor() + "\nYear: " + book.getYear());
                });
```

2：Servlet requests and logging requests to MongoDB

```java
@Override
protected void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
    long startTime = System.currentTimeMillis();
    // 1. Get user Input
    String query = request.getParameter( s: "q");
    if (query == null || query.isEmpty()) {
        response.sendError(HttpServletResponse.SC_BAD_REQUEST, s: "Missing 'q' parameter");
        return;
    }

    // 2. Open Library API
    String apiResponse = callOpenLibraryAPI(query);

    // 3. Parse and extract key data
    JsonObject bookData = parseApiResponse(apiResponse);

    // 4. Log to MongoDB
    logRequestToMongo(request, bookData);

    // 5. Return simplified JSON response
    response.setContentType("application/json");
    response.setCharacterEncoding("UTF-8");
    response.getWriter().write(bookData.toString());

    long apiLatency = System.currentTimeMillis() - startTime;
    logToMongoDB(request, query, bookData, apiLatency);
}
```

3：Exception handling：

```java
// 1. Get user Input
String query = request.getParameter( s: "q");
if (query == null || query.isEmpty()) {
    response.sendError(HttpServletResponse.SC_BAD_REQUEST, s: "Missing 'q' parameter");
    return;
}

// 2. Open Library API
String apiResponse = callOpenLibraryAPI(query);

// 3. Parse and extract key data
JsonObject bookData = parseApiResponse(apiResponse);
```

4，5，6：Recording and display of logs

```java
    private static final String COLLECTION_NAME = "searches";

    1 usage
    private void logToMongoDB(HttpServletRequest request, String query, JsonObject bookData, long apiLatency) {
        try (MongoClient client = MongoClients.create(CONN_STRING)) {
            MongoCollection<Document> logs = client.getDatabase(DB_NAME).getCollection(COLLECTION_NAME);

            // Build log document
            Document logEntry = new Document()
                    .append("timestamp", new Date())
                    .append("clientIp", request.getRemoteAddr())
                    .append("userAgent", request.getHeader( s: "User-Agent"))
                    .append("searchTerm", query)
                    .append("apiResponseTime", apiLatency + "ms")
                    .append("bookTitle", bookData.get("title").getAsString())
                    .append("httpStatus", HttpServletResponse.SC_OK);

            logs.insertOne(logEntry); // Insert into database
        } catch (Exception e) {
            System.err.println("MongoDB record Fail: " + e.getMessage());
        }
    }
}
```

Activity Analysis Dashboard

Top 5 Search Terms

Search Term	Count
null	3

Average API Response Time

0 ms

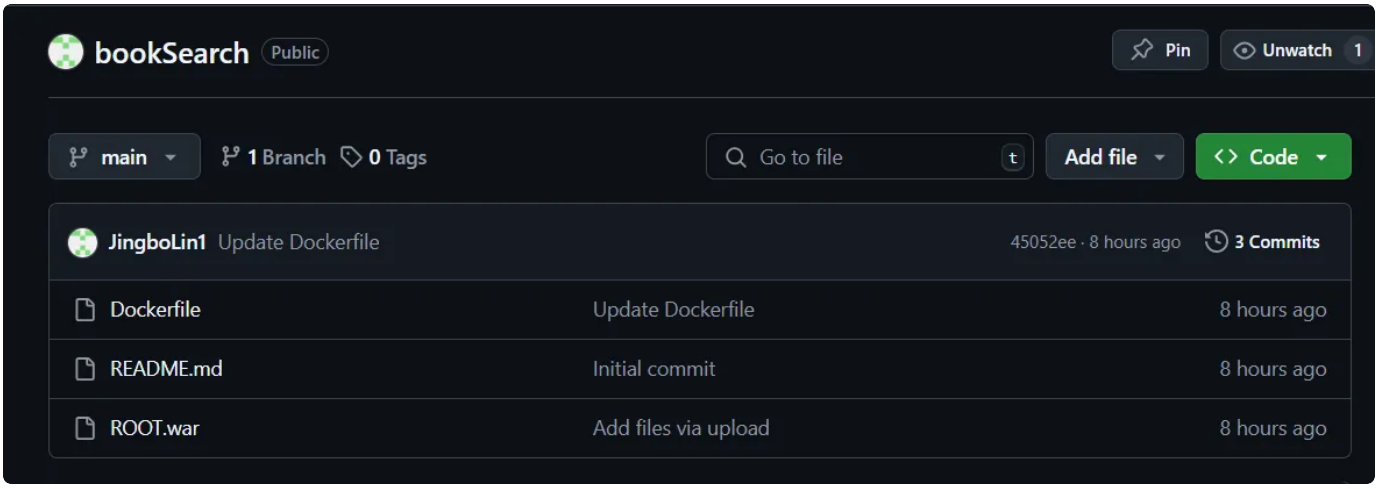
Device Type Distribution

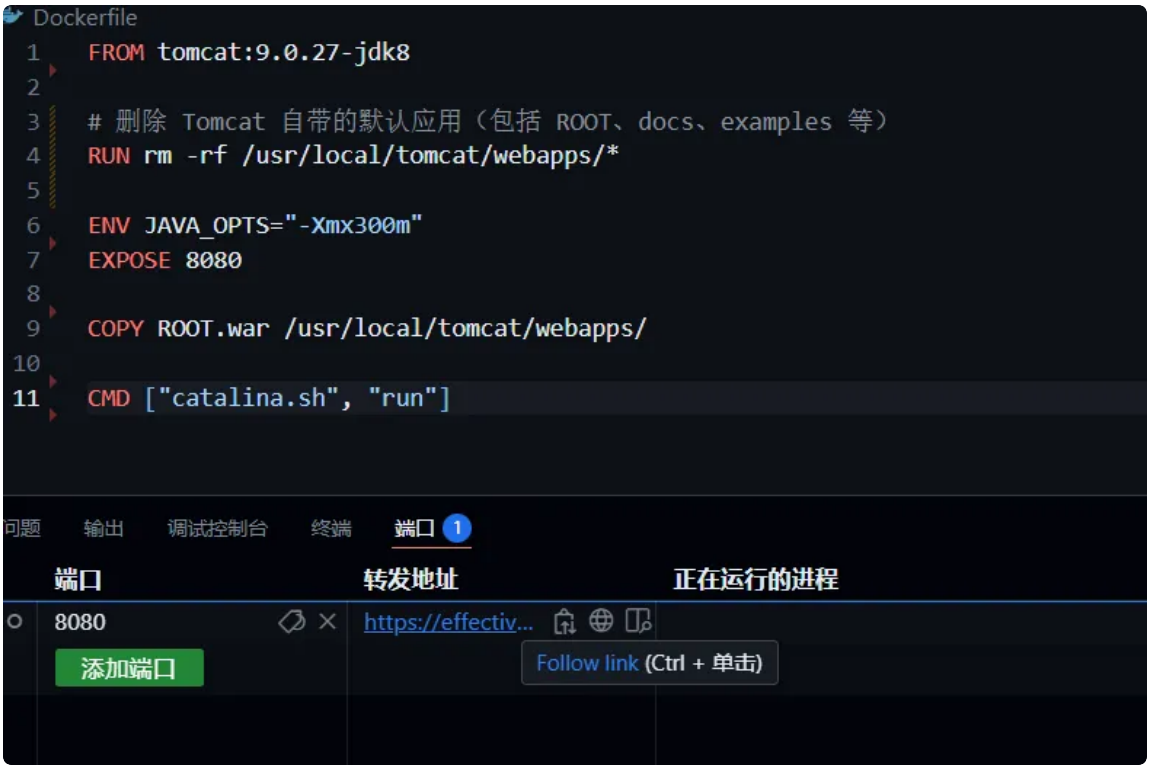
Device Type	Request Count
Other	3

Full Request Logs

Timestamp	IP Address	Search Term	User Agent	Response Time	Book Title
No Timestamp	Unknown IP	No Search Term	No Device Info	No Response Time	No Title
No Timestamp	Unknown IP	No Search Term	No Device Info	No Response Time	No Title
No Timestamp	Unknown IP	No Search Term	No Device Info	No Response Time	No Title

7：Deploy to git


bookSearch
Public
Pin
Unwatch 1
main
1 Branch
0 Tags
Go to file
t
Add file
Code
JingboLin1 Update Dockerfile
45052ee · 8 hours ago
3 Commits
Dockerfile
Update Dockerfile
8 hours ago
README.md
Initial commit
8 hours ago
ROOT.war
Add files via upload
8 hours ago


Dockerfile
FROM tomcat:9.0.27-jdk8
删除 Tomcat 自带的默认应用（包括 ROOT、docs、examples 等）
RUN rm -rf /usr/local/tomcat/webapps/*
ENV JAVA_OPTS="-Xmx300m"
EXPOSE 8080
COPY ROOT.war /usr/local/tomcat/webapps/
CMD ["catalina.sh", "run"]
问题
输出
调试控制台
终端
端口 1
端口
转发地址
正在运行的进程
8080
https://effectiv...
添加端口
Follow link (Ctrl + 单击)