Project 4: FindMyEvent - Ticketmaster Event Discovery System

My Application finds the events happening around you and provides you the details about the events like Date, Venue, Price Range,etc. You can search the event by category, Country Code or look through events using All option

1. Android Application

User Interface Components

The Android application includes the following view:

- **TextView**: For displaying event names, dates, locations, and other text information
- **EditText**: For user input when searching for events
- **RadioButtons**: For selecting search type (Category, Country Code, All Events)
- **Button**: For initiating searches and viewing tickets
- **RecyclerView**: For displaying lists of events
- **CardView**: For styling event items in the list
- **ProgressBar**: For indicating loading status
- **ScrollView**: For the details screen to accommodate varying content length

User Input

The app requires and processes user input in multiple ways:

- Search term input via EditText. You can search for Music, Dance, Sports, etc.
- You can search by Country Code: UK,US,AU,AS, etc.
- You can also search for all events.
- The search is capped at 10 results.

HTTP Requests

The application makes HTTP requests to the web service using OkHttp:

- Requests are executed on a background thread using AsyncTask
- Different endpoints are called based on search type (category, country, all)
- Proper error handling is implemented for network failures

Data Parsing

The app parses JSON responses from the web service:

- Uses Gson library to convert JSON to Java objects
- Maps the response to Event model objects
- Properly handles various data formats (dates, prices, locations)

Information Display

Event information is clearly displayed to the user:

- List view shows basic event information (name, date, venue, location)
- Detail view shows comprehensive information including pricing and ticket links
- Empty state and error messages are shown when appropriate

Repeatable Functionality

The application allows users to:

- Perform multiple searches without restarting
- Toggle between different search types
- View event details and return to the search results

Demo:

Search By All Events:

Find My Event

Search for events around the world

Category(e.g Music) Country Code All

SEARCH

SACRAMENTO KINGS VS. PHOENIX SUNS

Date: 2025-04-13 at 12:30:00
Venue: Golden 1 Center
Location: Sacramento, California, United States Of America

Phoenix Suns vs. Golden State Warriors

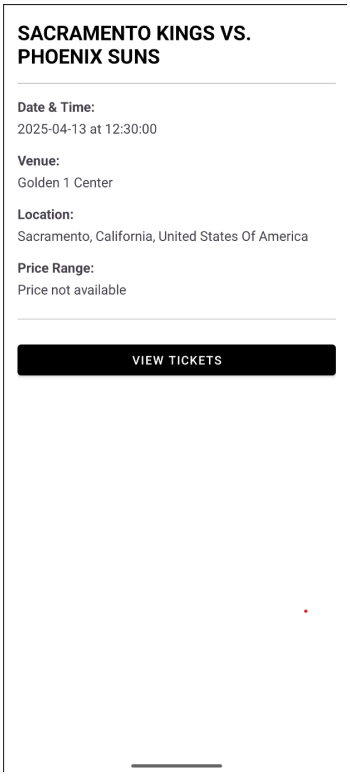
Date: 2025-04-08 at 19:00:00
Venue: PHX Arena (formerly Footprint Center)
Location: Phoenix, Arizona, United States Of America

Phoenix Suns vs. San Antonio Spurs

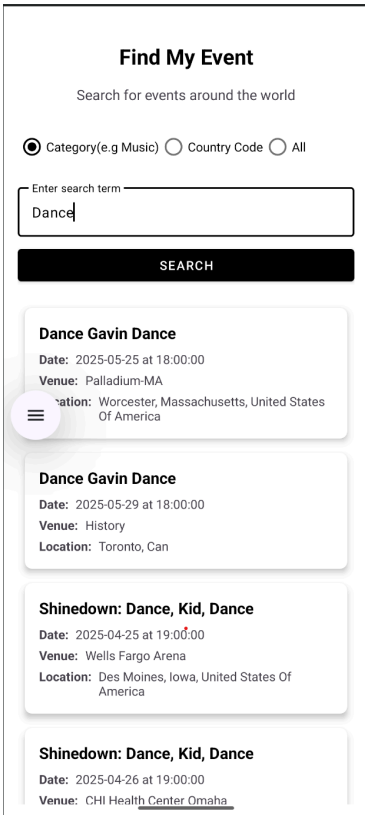
Date: 2025-04-11 at 19:00:00
Venue: PHX Arena (formerly Footprint Center)
Location: Phoenix, Arizona, United States Of America

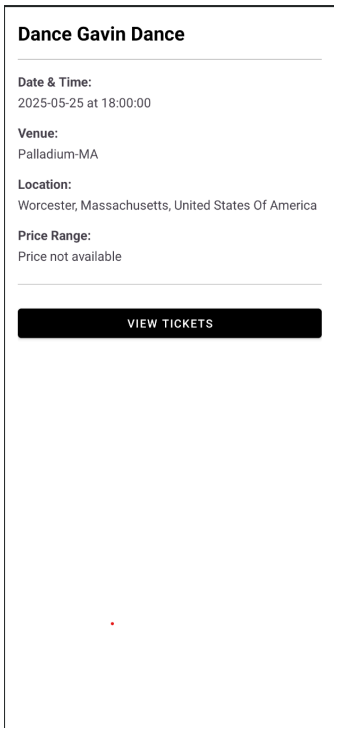
Phoenix Suns vs. Oklahoma City Thunder

Date: 2025-04-09 at 19:00:00
Venue: PHX Arena (formerly Footprint Center)

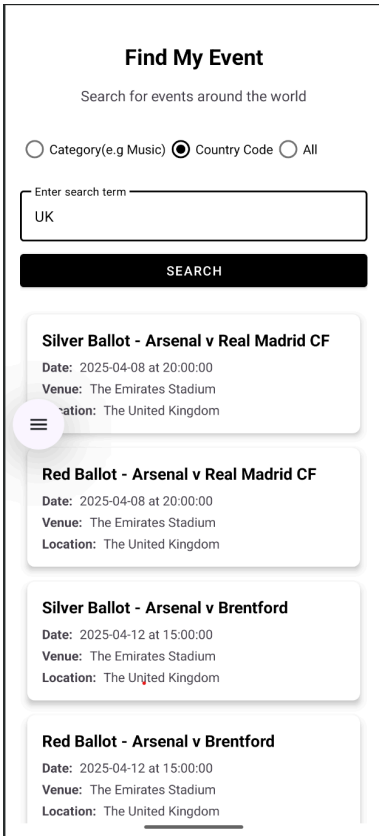


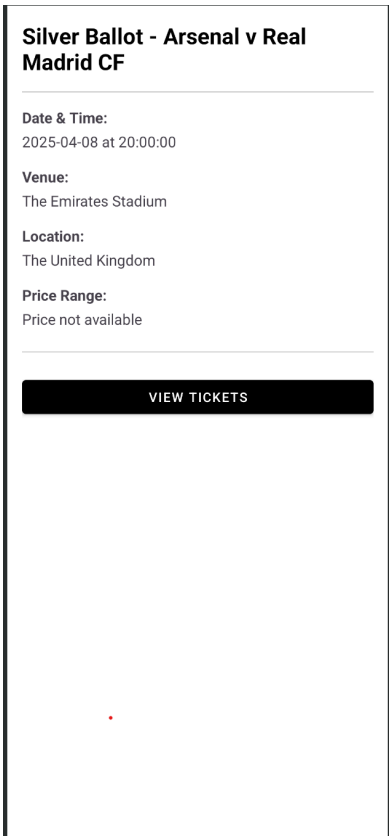
Search By Category:





Search By Country Code:


Find My Event
Search for events around the world
Category(e.g Music)
Country Code
All
Enter search term
UK
SEARCH
Silver Ballot - Arsenal v Real Madrid CF
Date: 2025-04-08 at 20:00:00
Venue: The Emirates Stadium
ation: The United Kingdom
Red Ballot - Arsenal v Real Madrid CF
Date: 2025-04-08 at 20:00:00
Venue: The Emirates Stadium
Location: The United Kingdom
Silver Ballot - Arsenal v Brentford
Date: 2025-04-12 at 15:00:00
Venue: The Emirates Stadium
Location: The United Kingdom
Red Ballot - Arsenal v Brentford
Date: 2025-04-12 at 15:00:00
Venue: The Emirates Stadium
Location: The United Kingdom


Silver Ballot - Arsenal v Real Madrid CF
Date & Time:
2025-04-08 at 20:00:00
Venue:
The Emirates Stadium
Location:
The United Kingdom
Price Range:
Price not available
VIEW TICKETS

2. Web Service

API Implementation

The web service implements a RESTful API that:

- Receives HTTP requests from the Android application
- Processes parameters (searchType, query)
- Executes business logic to fetch data from Ticketmaster using Ticketmaster API
- Returns structured JSON responses

Business Logic

The web service implements below business logic:

- Fetches event data from the Ticketmaster API
- Filters and processes the response
- Returns only the necessary information to the Android app
- Handles various search types (name/category, country, all events)

Third-Party API Integration

The web service successfully integrates with the Ticketmaster API:

- Makes HTTP requests to Ticketmaster endpoints
- Passes appropriate parameters based on user search
- Processes XML/JSON responses from Ticketmaster
- Handles rate limiting and API errors appropriately

Response Formatting

The web service formats responses appropriately:

- Returns JSON-formatted data to the Android app
- Includes only necessary information (no extraneous data)
- Properly structures the response for easy parsing

3. Logging and Dashboard

Information Logging

The web service logs below pieces of information for each request/reply:

1. Client IP address
2. Search type
3. Search Query
4. Timestamp of request/response
5. Response status and time
6. Number of results returned

Data Storage

All logs are stored persistently in MongoDB:

- Log entries include request and response details
- Data is properly structured for analytics
- Timestamps allow for chronological analysis

Operations Analytics

The dashboard displays at least 3 interesting analytics:

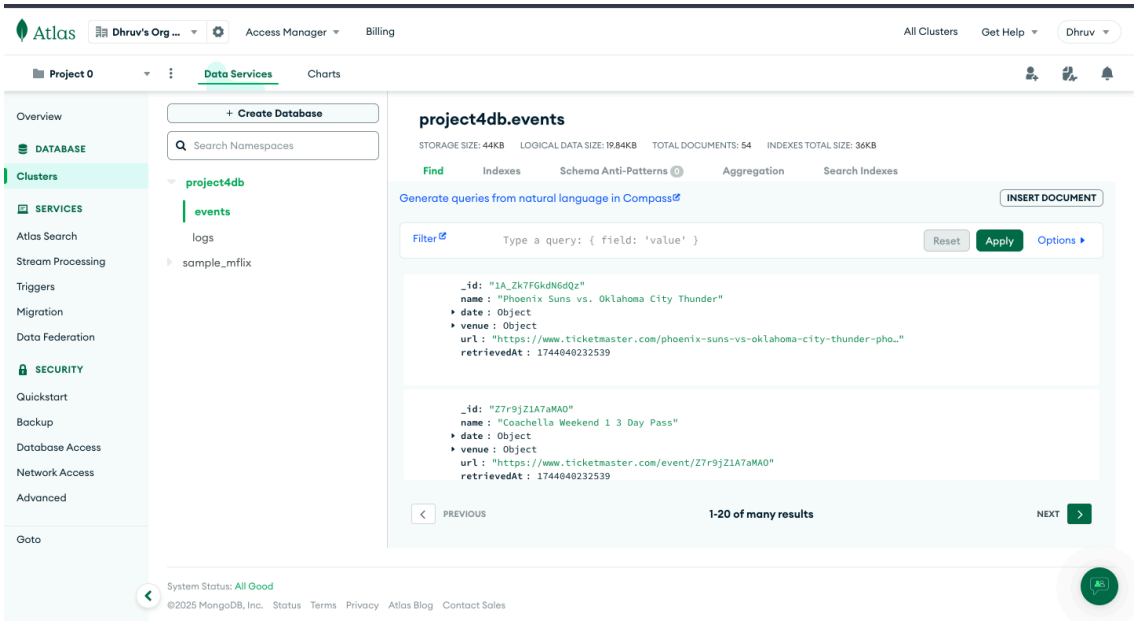
1. Most popular search terms
2. Average response time for queries
3. Distribution of search types (category, country, all)

Log Display

The dashboard provides a formatted display of logs:

- Shows detailed information for each request
- Formats data in a human-readable way (not as JSON or XML)
- Allows viewing historical activity

Mongo DB Logging:



Web Service Landing page:

Ticketmaster API Web Service

API Endpoints

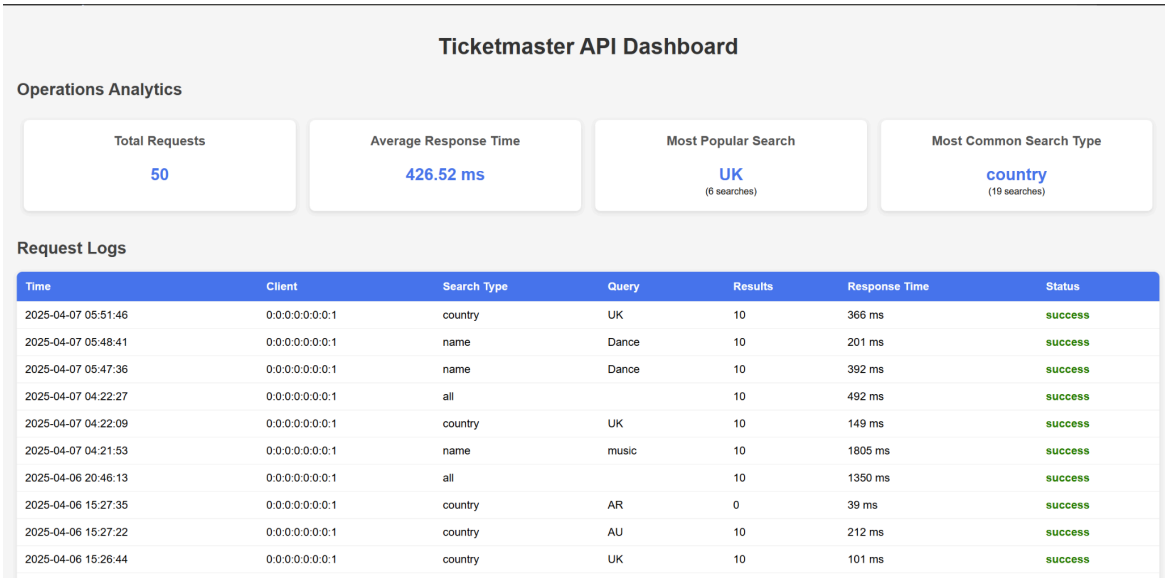
This service provides access to Ticketmaster event data.

- **GET /events** - Get all events
- **GET /events?searchType=name&query=YOUR_QUERY** - Search events by name
- **GET /events?searchType=country&query=COUNTRY_CODE** - Search events by country

Dashboard

View analytics and logs for the API service:

Open Dashboard



Ticketmaster API Dashboard

Operations Analytics

Total Requests	Average Response Time	Most Popular Search	Most Common Search Type
50	426.52 ms	UK (6 searches)	country (19 searches)

Request Logs

Time	Client	Search Type	Query	Results	Response Time	Status
2025-04-07 05:51:46	0:0:0:0:0:0:0:1	country	UK	10	366 ms	success
2025-04-07 05:48:41	0:0:0:0:0:0:0:1	name	Dance	10	201 ms	success
2025-04-07 05:47:36	0:0:0:0:0:0:0:1	name	Dance	10	392 ms	success
2025-04-07 04:22:27	0:0:0:0:0:0:0:1	all		10	492 ms	success
2025-04-07 04:22:09	0:0:0:0:0:0:0:1	country	UK	10	149 ms	success
2025-04-07 04:21:53	0:0:0:0:0:0:0:1	name	music	10	1805 ms	success
2025-04-06 20:46:13	0:0:0:0:0:0:0:1	all		10	1350 ms	success
2025-04-06 15:27:35	0:0:0:0:0:0:0:1	country	AR	0	39 ms	success
2025-04-06 15:27:22	0:0:0:0:0:0:0:1	country	AU	10	212 ms	success
2025-04-06 15:26:44	0:0:0:0:0:0:0:1	country	UK	10	101 ms	success

Dashboard that maintains logs and other parameters.

4. Deployment

The web service is successfully deployed to GitHub Codespaces using Docker

To run the CodeSpace you will have to create a new codespace and update the base url in the APIService file in the Android app.

Remember to make the port public once the codespace is running.

```
// Replace with your deployed web service URL
1 usage
private static final String BASE_URL = "https://symmetrical-spork-x4rr7grwxrxh6995-8080.app.github.dev/";
```

5. Run the application

To run the application follow the below steps:

1. Download EventFinderAndroidApp and from https://github.com/CMU-Heinz-95702/distributed-systems-project-04-TangriDhruv
2. Create a codespace by clicking on the <>Code button on the github repo and then select the codespace tab -> Create a new codespace.
3. Once the codespace is running click on the ports tab and make the port public by right clicking on the ports tab.
4. Click on the globe icon once you hover over the forward address column.

5. Copy the url and update it in APIService file in Android App as mentioned in Deployment.
6. Now run the android app.
7. If you want to run the webservice in local download FindMyEventWebservices file from the same github repo and run it in Intellij.