Project 4 Task 2

Distributed Systems

Name : Abhishek Venkatadri

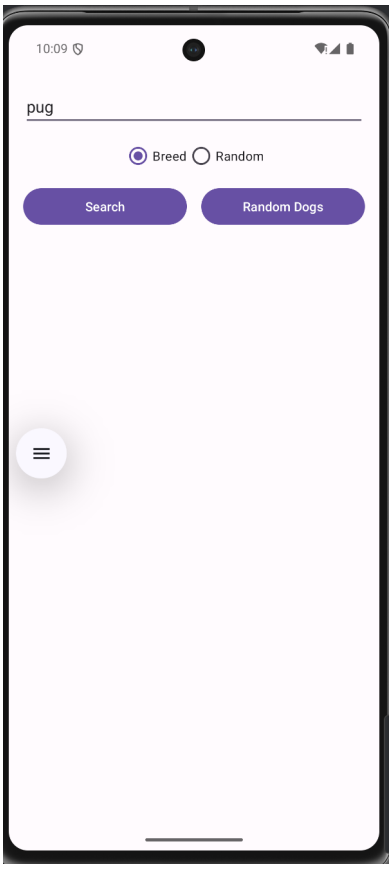
Andrew id : av3

This project illustrates an Android application and a Java Servlet-based web service that I developed to allow users to browse and search various dog breeds using The Dog API. I designed and implemented both the frontend (native Android) and backend (Java Servlet) components. The backend logs all user requests to MongoDB Atlas, and I developed a web-based dashboard to display operational statistics such as request counts, status codes, and API response times. The name of my native Android application project in Android Studio is: MobileApplication

a. **Has at least three different kinds of views in your Layout (TextView, EditText, ImageView, etc.)** My application uses **TextView**, **EditText**, **Button**, **RadioButton**, **ImageView**, and a **RecyclerView** to display the search results. These views are incorporated using **ConstraintLayout**, and the layout is defined in activity_main.xml.

Here is screenshot of the layout before the picture has been fetched

b) Requires input from the user



c. Makes an HTTP request (using an appropriate HTTP method) to your web service

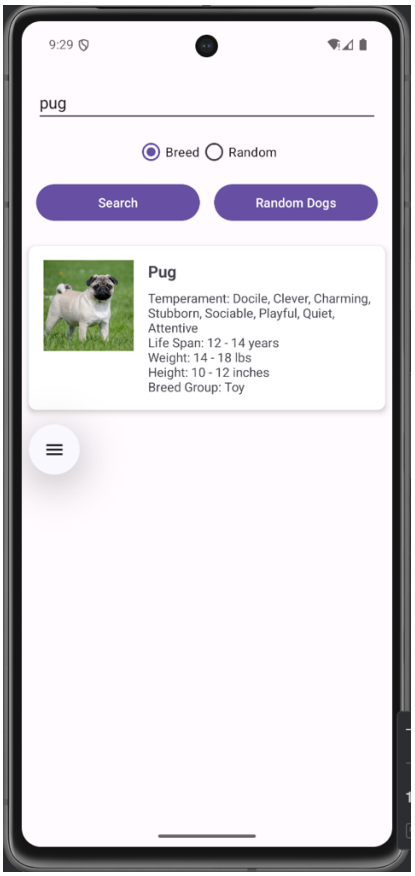
My application makes an HTTP GET request using the NetworkTask.java class. Depending on the user interaction, the app sends requests to:

https://fuzzy-space-pancake-jj4pxvq6xgg6hw4x-8080.app.github.dev/api/breeds – to get all dog breeds.

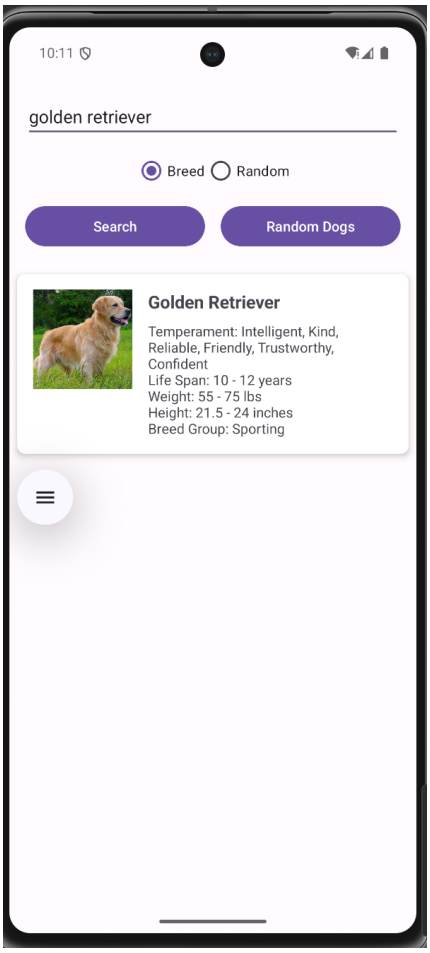
https://fuzzy-space-pancake-jj4pxvq6xgg6hw4x-8080.app.github.dev/api?limit=5 – to fetch random dog images.

d)These endpoints are handled by a Java servlet (ApiServlet.java) on the backend. It forwards the request to The Dog API, retrieves the data, logs the request, and sends the response back to the app.

e) Displays new information to the user



f) Is repeatable when you give another data



2. Implement a web application, deployed to GitHub Codespaces

The URL of my web service deployed to GitHub Codespaces is:

https://fuzzy-space-pancake-jj4pxvq6xgg6hw4x-8080.app.github.dev

The project directory name is: Project4Task2

a. Using an HttpServlet to implement a simple (can be a single path) API

In my web application project:

Model: Logic is handled within ApiServlet.java and JSON parsing is done using org.json libraries.

Controller:

ApiServlet.java — Handles /api and /api/breeds endpoints

DashboardServlet.java — Handles /dashboard endpoint for analytics

These servlets act as the controller and directly handle routing and response formatting. A separate DAO layer isn't needed because MongoDB operations are handled through a singleton helper class: MongoDBConnection.java.

b. Receives an HTTP request from the native Android application

The Android app sends a GET request to the web service using NetworkTask.java.

Example URLs:

/api/breeds for listing breeds

/api?limit=5 for random dogs

These requests are received by ApiServlet.java, which handles parsing, validation, logging, and forwarding the request to The Dog API.

c. Executes business logic appropriate to your application

ApiServlet.java performs the following:

Parses the user query or default parameters

Forwards the request to The Dog API (https://api.thedogapi.com/v1/breeds or /v1/images/search)

Parses the JSON response from The Dog API

Logs request data into MongoDB Atlas

Sends back a formatted JSON response to the Android app

This enables seamless integration between the mobile app and the external API, with analytics tracked through MongoDB.

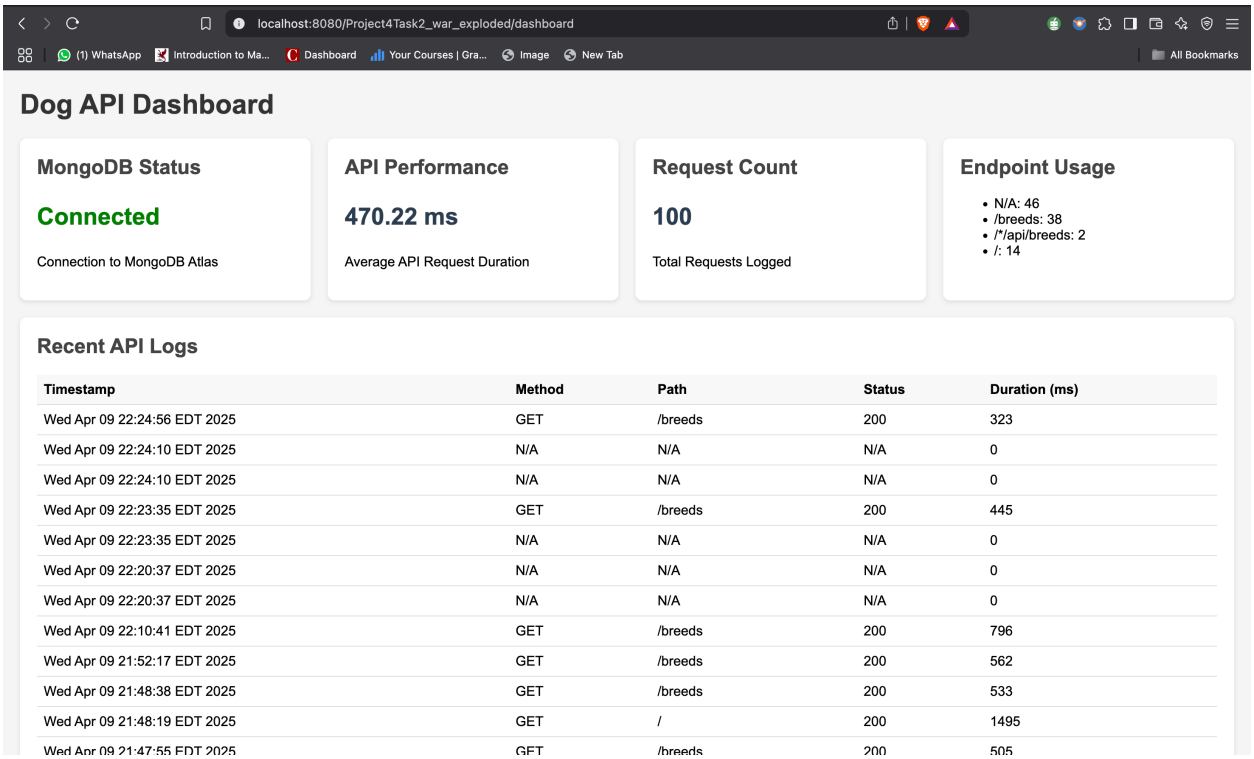
d. Replies to the Android application with an XML or JSON formatted response

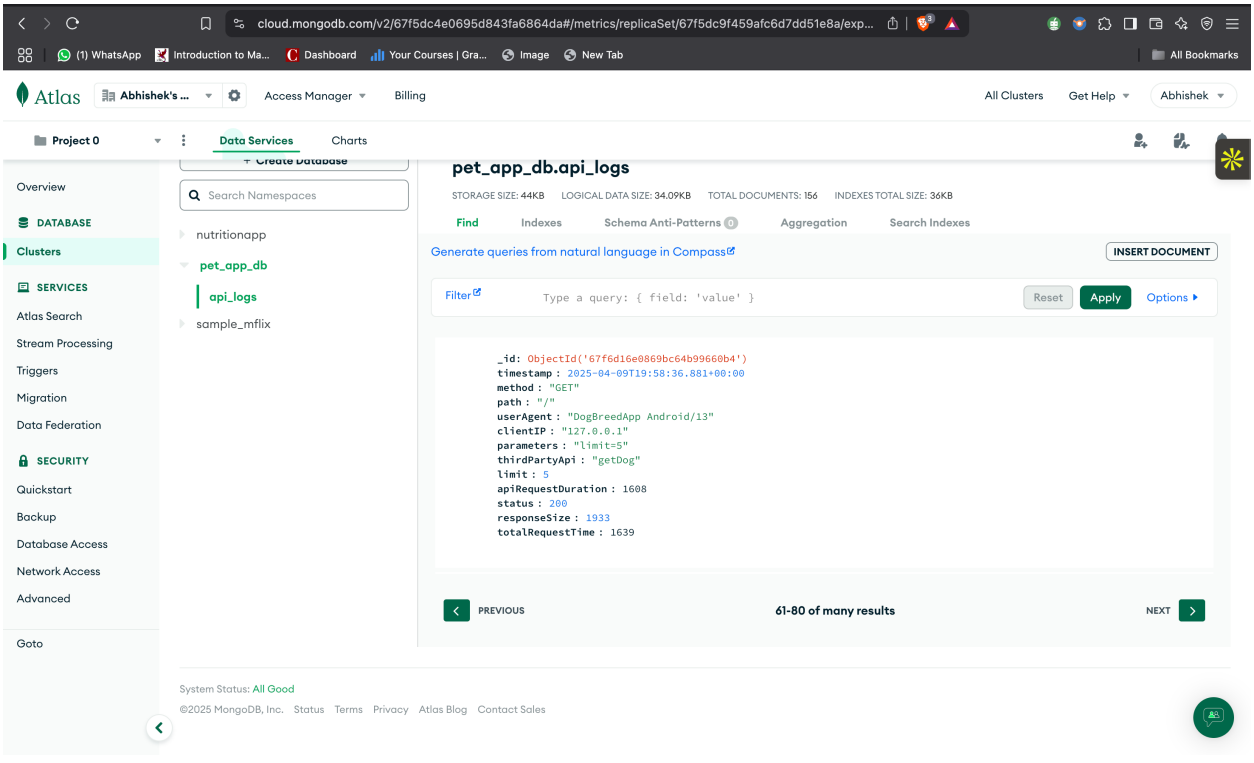
The backend replies to the Android application in JSON format. The response is constructed inside the ApiServlet.java based on data retrieved from The Dog API. Depending on the endpoint (/api or /api/breeds), the servlet processes the external API response and returns a simplified and structured JSON array back to the client.

Here's an example of the JSON response returned for a breed query:

API JSON


```
[
  {
    "name": "Pug",
    "temperament": "Docile, Clever, Charming",
    "life_span": "12 - 14 years",
    "weight": "14 - 18 lbs",
    "height": "10 - 12 inches",
    "breed_group": "Toy",
    "image_url": "https://cdn2.thedogapi.com/images/BJa4kxc4X.jpg"
  }
]
```


localhost:8080/Project4Task2_war_exploded/dashboard
(1) WhatsApp
Introduction to Ma...
Dashboard
Your Courses | Gra...
Image
New Tab
All Bookmarks
Dog API Dashboard
MongoDB Status
Connected
Connection to MongoDB Atlas
API Performance
470.22 ms
Average API Request Duration
Request Count
100
Total Requests Logged
Endpoint Usage
N/A: 46
/breeds: 38
/*/api/breeds: 2
/: 14
Recent API Logs
Timestamp | Method | Path | Status | Duration (ms)
Wed Apr 09 22:24:56 EDT 2025 | GET | /breeds | 200 | 323
Wed Apr 09 22:24:10 EDT 2025 | N/A | N/A | N/A | 0
Wed Apr 09 22:24:10 EDT 2025 | N/A | N/A | N/A | 0
Wed Apr 09 22:23:35 EDT 2025 | GET | /breeds | 200 | 445
Wed Apr 09 22:23:35 EDT 2025 | N/A | N/A | N/A | 0
Wed Apr 09 22:20:37 EDT 2025 | N/A | N/A | N/A | 0
Wed Apr 09 22:20:37 EDT 2025 | N/A | N/A | N/A | 0
Wed Apr 09 22:10:41 EDT 2025 | GET | /breeds | 200 | 796
Wed Apr 09 21:52:17 EDT 2025 | GET | /breeds | 200 | 562
Wed Apr 09 21:48:38 EDT 2025 | GET | /breeds | 200 | 533
Wed Apr 09 21:48:19 EDT 2025 | GET | / | 200 | 1495
Wed Apr 09 21:47:55 EDT 2025 | GET | /breeds | 200 | 505


cloud.mongodb.com/v2/67f5dc4e0695d843fa6864da#/metrics/replicaSet/67f5dc9f459afc6d7dd51e8a/exp...
Atlas
Abhishek's ...
Access Manager
Billing
All Clusters
Get Help
Abhishek
Project 0
Data Services
Charts
Overview
DATABASE
Clusters
SERVICES
Atlas Search
Stream Processing
Triggers
Migration
Data Federation
SECURITY
Quickstart
Backup
Database Access
Network Access
Advanced
Goto
Search Namespaces
nutritionapp
pet_app_db
api_logs
sample_mflix
pet_app_db.api_logs
STORAGE SIZE: 44KB
LOGICAL DATA SIZE: 34.09KB
TOTAL DOCUMENTS: 156
INDEXES TOTAL SIZE: 36KB
Find
Indexes
Schema Anti-Patterns
Aggregation
Search Indexes
Generate queries from natural language in Compass
INSERT DOCUMENT
Filter
Type a query: { field: 'value' }
Reset
Apply
Options
_id: ObjectId('67f6d16e0869bc64b99660b4')
timestamp : 2025-04-09T19:58:36.881+00:00
method : "GET"
path : "/"
userAgent : "DogBreedApp Android/13"
clientIP : "127.0.0.1"
parameters : "limit=5"
thirdPartyApi : "getDog"
limit : 5
apiRequestDuration : 1608
status : 200
responseSize : 1933
totalRequestTime : 1639
PREVIOUS
61-80 of many results
NEXT
System Status: All Good
©2025 MongoDB, Inc. Status Terms Privacy Atlas Blog Contact Sales

Note = Not every dog might have an image but all dogs will have description