Project 4 Task 2 – Distributed Application and Dashboard

Project 4 Task 2 – Distributed Application and Dashboard

Team

Team Member 1

- **Name:** Zoey Chou
- **AndrewID:** ichou
- **Email:** ichou@andrew.cmu.edu

Team Member 2

- **Name:** Sheldon Shi
- **AndrewID:** lijuns
- **Email:** lijuns@andrew.cmu.edu

Description

```
The application will query related GIF files based on the user's input
information and return a set of five related GIF files.
Upon receiving a GIF file from the server, the application displays the
file.
The returned GIF link will be displayed under each GIF.
```

Distributed Application Requirements

1. Implement a native Android application

The name of the native Android application project in Android Studio is: **GIFBot** .
The project folder is: **Project4Task2AndroidApp**

a. Has at least three different kinds of views in your Layout (TextView, EditText, ImageView, etc.)

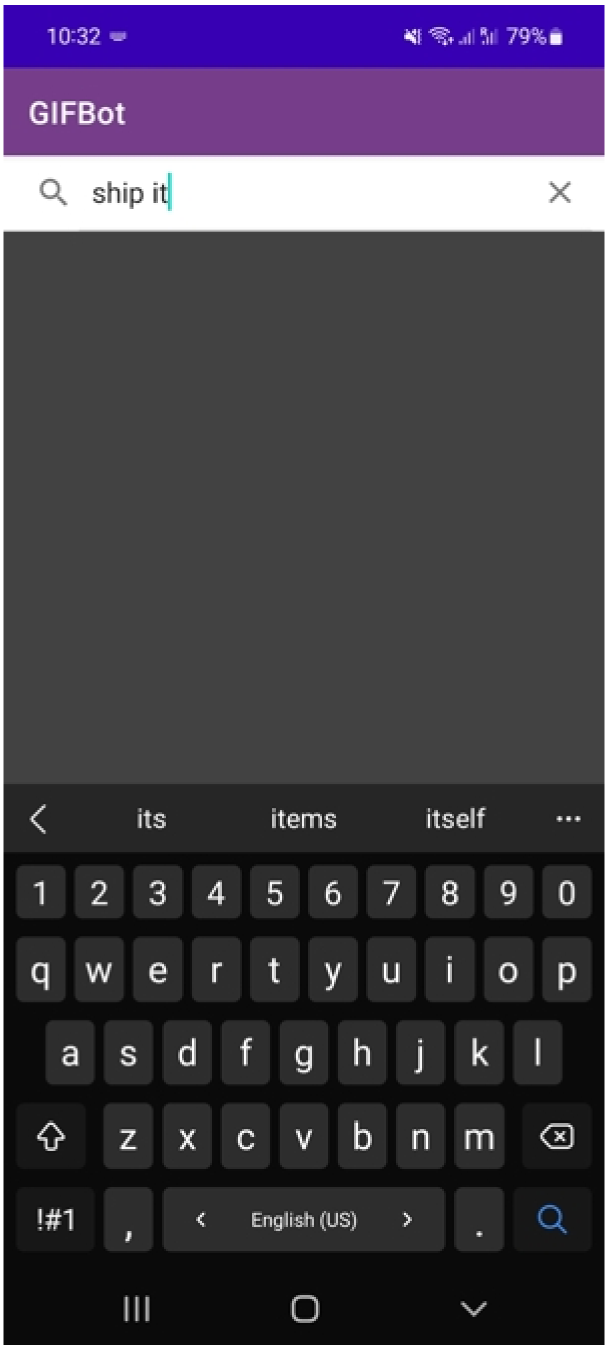
The application uses **TextView**, **SearchView**, **ImageView**, and **ListView**.
Refer to **activity_gifbot.xml** and **gif_list_item.xml** for details of how they are arranged in the Layout.

Below is a screenshot of the Android layout after starting the application

10:30
79%
GIFBot
Search all the GIFs

b.Requires input from the user

Below is a screenshot of the user searching for a GIFs of a 'ship it'.



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

The application does an HTTP GET request in **GIFProcessor.java**. The HTTP request is: https://ds-project4-gifbot.herokuapp.com/Project4Task2WebService-1.0-SNAPSHOT/api/v1/gif?search=ship it as the case of searching a `ship it` GIFs.
It also sends the following data as the http headers for the statistic use.

```
Manufacture: samsung
Brand: samsung
Model: SM-A226B
AndroidVersion: 12
```

The search method makes this request using **OkHttpClient** and parses the returned JSON with **Jackson** to find the GIFs URL, fetches the GIFs, and returns the GIFs of the picture.

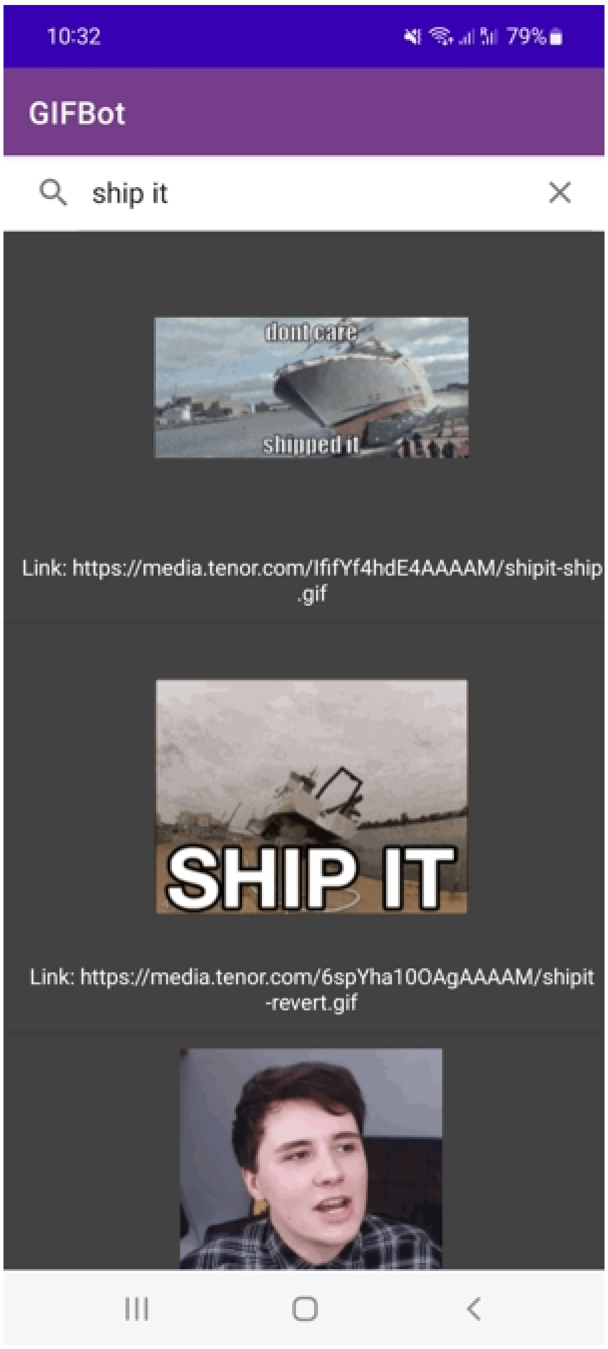
d. Receives and parses an XML or JSON formatted reply from the web service

- The response JSON from the web service

```
{
    "gifs": [
        "https://media.tenor.com/exU0RKqTu00AAAAM/i-see-it-i-ship-it.gif",
        "https://media.tenor.com/1Hoy6ABLpIcAAAAM/binoculars-ship.gif",
        "https://media.tenor.com/jllngY0wdqwAAAAM/i-ship-it.gif",
        "https://media.tenor.com/pAc4dATt5E4AAAAM/i-ship-it-yes.gif",
        "https://media.tenor.com/QG9WKduTbw4AAAAM/ship-it.gif",
        "https://media.tenor.com/y5dXmGmoBCQAAAAM/msftgarage-
microsoftgarage.gif",
        "https://media.tenor.com/NtLKdRlH9HgAAAAM/i-ship-it-shipped.gif",
        "https://media.tenor.com/IfifYf4hdE4AAAAM/shipit-ship.gif",
        "https://media.tenor.com/6spYha10OAgAAAAM/shipit-revert.gif",
        "https://media.tenor.com/H5Ml6GdT12cAAAAM/i-ship-it-ship-it.gif"
    ],
    "statusCode": 200
}
```

e. Displays new information to the user

Below is the screen shot after the five GIFs has been returned.

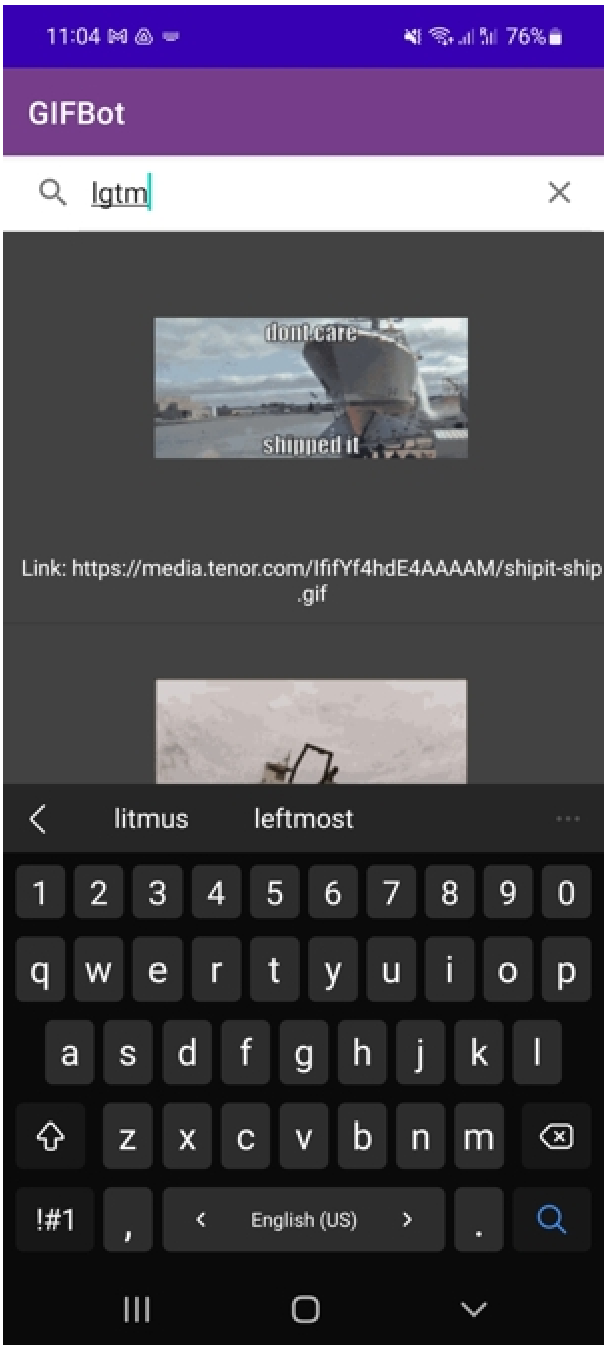


Link: https://media.tenor.com/IfifYf4hdE4AAAAM/shipit-ship.gif

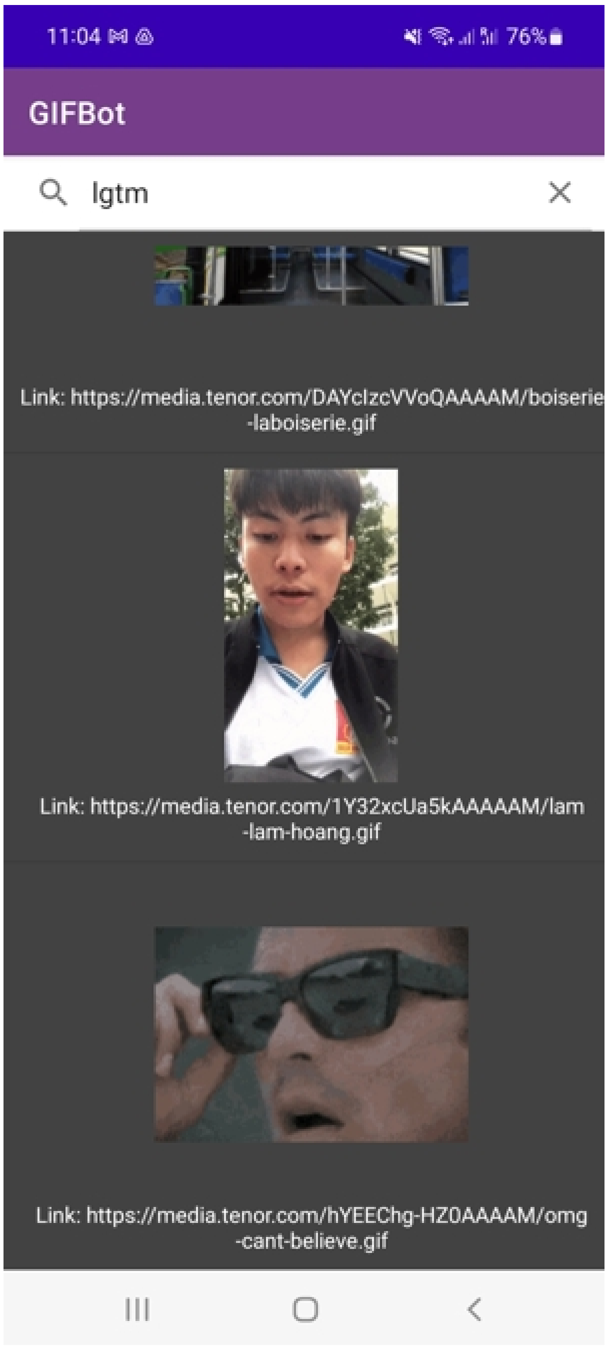
Link: https://media.tenor.com/6spYha10OAgAAAAM/shipit-revert.gif

f. Is repeatable (I.e. the user can repeatedly reuse the application without restarting it.)

The user can type in another search term and hit Submit.
Below is an example of searching **lgtm** after seaching **ship it** GIFs.



Link: https://media.tenor.com/IfifYf4hdE4AAAAM/shipit-ship.gif



Link: https://media.tenor.com/DAYclzcVVoQAAAAM/boiserie-laboiserie.gif

Link: https://media.tenor.com/1Y32xcUa5kAAAAAM/lam-lam-hoang.gif

Link: https://media.tenor.com/hYEEChg-HZ0AAAAM/omg-cant-believe.gif

2. Implement a web application, deployed to Heroku

The URL of the web service deployed to Heroku is: **ds-project4-gifbot**
The project directory name is **Project4Task2WebService**.

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

There are two parts in the web application project, one is for the endpoint **/api/v1/gif** to search gifs based on the user input, and the other one is for the endpoint **/dashboard** to generate and display the dashboard page.

The endpoint **/api/v1/gif**
Model :

```
model
├── ClientRequestInfo.java
├── ClientResponseInfo.java
├── DeviceInfo.java
├── ErrorResponse.java
├── GIFsResponse.java
├── IResponse.java
├── Latency.java
├── LogEvent.java
├── tenor
│   ├── TenorMediaFormat.java
│   ├── TenorResponse.java
│   ├── TenorResult.java
│   └── TenorTinyGif.java
├── TenorRequestInfo.java
├── TenorResponseInfo.java
```

Controller:

```
web
└── GIFBotServlet.java
```

Service:

```
service
├── GIFBotService.java
└── LoggingService.java
```

Repository:

```
repository
├── LoggingRepository.java
└── MongoDBClient.java
```

The endpoint **/dashboard**
Model:

```
model
├── Dashboard.java
├── Latency.java
├── LogEvent.java
├── SystemLog.java
└── TopDeviceInfo.java
```

Controller:

```
web
├── DashboardServlet.java
```

View:

```
webapp
├── index.jsp
```

Service:

```
service
├── DashboardService.java
└── LoggingService.java
```

Repository:

```
repository
├── LoggingRepository.java
└── MongoDBClient.java
```

b. Receives an HTTP request from the native Android application

GIFBotService.java receives the HTTP GET request with the argument "search".
It passes the search string on to the **GIFBotService.java** to search the gifs from the 3rd party api.

c. Executes business logic appropriate to your application

GIFBotService.java makes an HTTP request to:
https://tenor.googleapis.com/v2/search
It then parses the JSON response and extracts the parts it needs to respond to the Android application.

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

GIFBotService.java formatted the response based on the resopnse of the 3rd party api. If the the 3rd party api responds successfully, it will return the following JSON resoponse to the mobile app.

```
{
    "gifs": [
        "https://media.tenor.com/exU0RKqTu00AAAAM/i-see-it-i-ship-it.gif",
        "https://media.tenor.com/1Hoy6ABLpIcAAAAM/binoculars-ship.gif",
        "https://media.tenor.com/jllngY0wdqwAAAAM/i-ship-it.gif",
        "https://media.tenor.com/pAc4dATt5E4AAAAM/i-ship-it-yes.gif",
        "https://media.tenor.com/QG9WKduTbw4AAAAM/ship-it.gif",
        "https://media.tenor.com/y5dXmGmoBCQAAAAM/msftgarage-
microsoftgarage.gif",
        "https://media.tenor.com/NtLKdRlH9HgAAAAM/i-ship-it-shipped.gif",
        "https://media.tenor.com/IfifYf4hdE4AAAAM/shipit-ship.gif",
        "https://media.tenor.com/6spYha1OOAgAAAAM/shipit-revert.gif",
        "https://media.tenor.com/H5Ml6GdT12cAAAAM/i-ship-it-ship-it.gif"
    ],
    "statusCode": 200
}
```

Otherwise, it will retrun an error resoponse as following JSON.

```
{
    "message": "invalid input parameter",
    "statusCode": 400
}
```

3. Handle error conditions - Does not need to be documented.

- Invalid mobile app input
  - If the user input is invalid, the service returns the error message on the screen
- Invalid server-side input (regardless of mobile app input validation)
  - If the query parameter `search` is invalid(empty or longer than 50 characters), it will return the error response as follows:

```
{
    "message": "invalid input parameter",
    "statusCode": 400
}
```

- Mobile app network failure, unable to reach server
  - The app will display an error message
- Third-party API unavailable
- Third-party API invalid data
  - Both of the above will be handle on the the exception cause section.

```
// GIFBotService.java, search method
catch (IOException e) {
            return new ErrorResponse(e.getMessage(),
HttpServletResponse.SC_BAD_GATEWAY);
        }
```

4. Log useful information - Itemize what information you log and why you chose it.

- Request information from the mobile phone
  - This information can be use to record the details of the request. When there is an issue, the developer can know what the user's input.
  - The serarch term is also used for the statistic analysis.
- Request device information
  - This can help resolve the issuse with the particular user's device

    - It is also used for the statistic analysis.
- Response to the mobile
    - This information is useful for track the content sent to the mobile
    - It is also used for the statistic analysis.
- Request information to the 3rd party API
    - When the 3rd party API returns an error, this log can help the develop know what it sent to the 3rd party.
- Resopnse from the 3rd party API
    - When there is an issue about the response, it can know the raw response from the 3rd party API.
- The latency of the web service
    - This can help the develop know the performance of the system.
    - It is also used for the statistic analysis.
- The latency of the 3rd party API
    - This can help the develop know the performance of the 3rd party.
    - It is also used for the statistic analysis.
- Date and time for each request and response
    - When there is an issue, it can help to find the correct information based on the time.

5. Store the log information in a database

The web service can connect, store, and retrieve information from a MongoDB database in the cloud.

- MongoDBClient.java handles the connect to the MongoDB and it is a singleton.

```java
package edu.cmu.andrew.project4task2webservice.repository;

import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoClients;

public class MongoDBClient {
    // mongodb connecting string
    private final static String CONNECTION_STRING =
"mongodb+srv://dsteamproject4:JyHsGiSpAuU44rrL@cluster0.y3xuxkb" +
            ".mongodb.net/?retryWrites=true&w=majority";

    // create instance of MongoDBClient
    private static MongoDBClient instance = null;
    // create MongoClient object
    private final MongoClient client;

    // constructor of MongoDBClient
    private MongoDBClient() {
        client = MongoClients.create(CONNECTION_STRING);
    }

    // get instance of MongoDBClient
    public static synchronized MongoDBClient getInstance() {
        // check if MongoClient instance is null, then create new one
        if (instance == null) {
            instance = new MongoDBClient();
        }
        return instance;
    }

    // get MongoClient object
    public MongoClient getClient() {
        return client;
    }
}
```

- LoggingRepository.java handles the log insertion and query.
    - In this web server, it use the `upsert` to handle the log. It will insert a new log if the log with same id is not in the database. Otherwise, it will update the record to add more log information.
    - `_id` is an **UUID** as the request id. For each request, the request id is used for the whole processing. So that there is one log record for each request.

```java
// part of the LoggingRepository.java code
/**
 * upsert the logEvent data to the MongoDB
 *
 * @param logEvent
 * @return
 */
public Document upsert(LogEvent logEvent) {
    // define the filer with _id
    Bson filter = Filters.eq("_id", logEvent.getRequestID());
    // set up the find and update option
    FindOneAndUpdateOptions upsertOptions = new FindOneAndUpdateOptions();
    upsertOptions.returnDocument(ReturnDocument.AFTER);
    upsertOptions.upsert(true);
    // do the inert or update
    return getCollection().findOneAndUpdate(filter, toBson(logEvent),
upsertOptions);
}
```

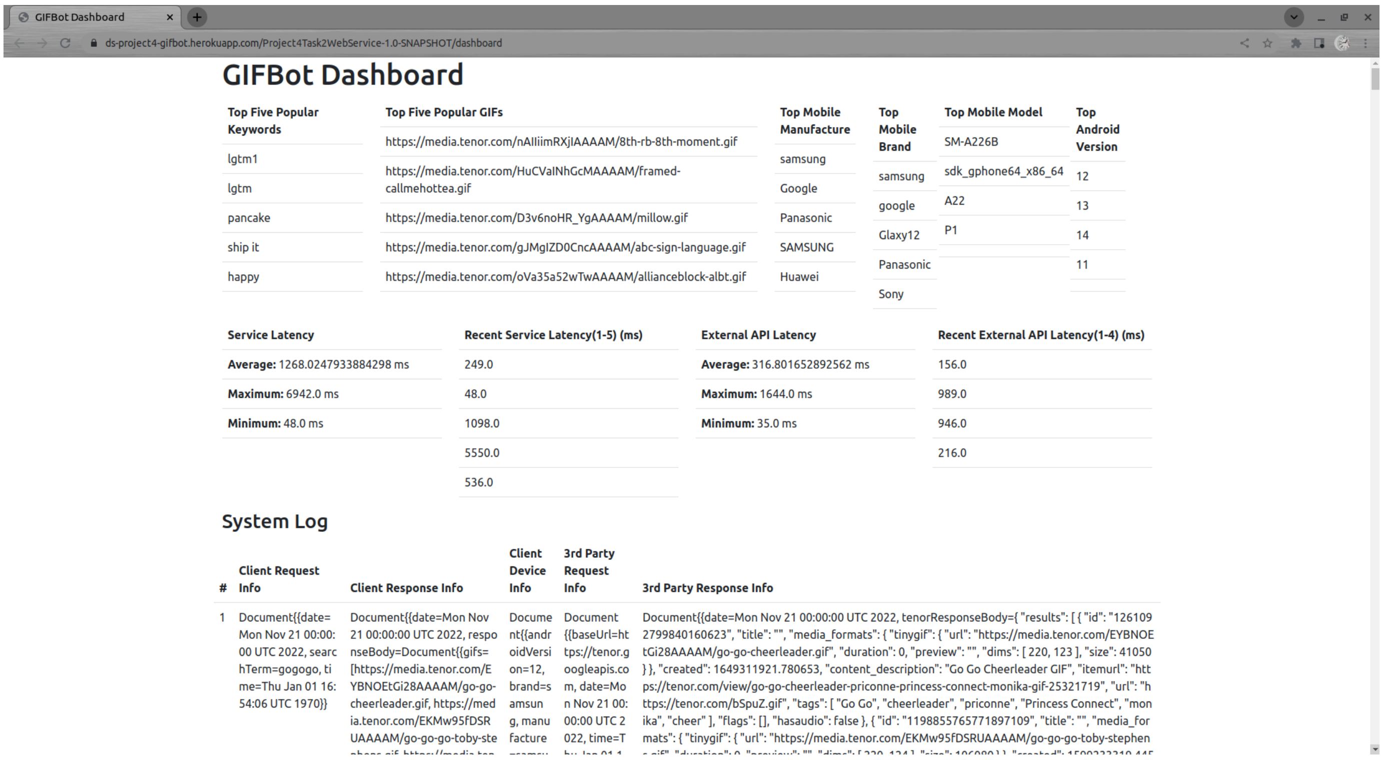
6. Display operations analytics and full logs on a web-based dashboard

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

- URL: (both are diected to the same page)
  - https://ds-project4-gifbot.herokuapp.com/Project4Task2WebService-1.0-SNAPSHOT/
  - https://ds-project4-gifbot.herokuapp.com/Project4Task2WebService-1.0-SNAPSHOT/dashboard

b. The dashboard displays at least 3 interesting operations analytics.

Below is a screenshot of the interesting operations analytics in the dashboard.


GIFBot Dashboard
ds-project4-gifbot.herokuapp.com/Project4Task2WebService-1.0-SNAPSHOT/dashboard

GIFBot Dashboard

Top Five Popular Keywords
lgtm1
lgtm
pancake
ship it
happy

Top Five Popular GIFs
https://media.tenor.com/nAIliimRXjIAAAAM/8th-rb-8th-moment.gif
https://media.tenor.com/HuCVaINhGcMAAAAM/framed-callmehottea.gif
https://media.tenor.com/D3v6noHR_YgAAAAM/millow.gif
https://media.tenor.com/gJMgIZD0CncAAAAM/abc-sign-language.gif
https://media.tenor.com/oVa35a52wTwAAAAM/allianceblock-albt.gif

Top Mobile Manufacture
samsung
Google
Panasonic
SAMSUNG
Huawei

Top Mobile Brand
samsung
google
Glaxy12
Panasonic
Sony

Top Mobile Model
SM-A226B
sdk_gphone64_x86_64
A22
P1

Top Android Version
12
13
14
11

Service Latency
Average: 1268.0247933884298 ms
Maximum: 6942.0 ms
Minimum: 48.0 ms

Recent Service Latency(1-5) (ms)
249.0
48.0
1098.0
5550.0
536.0

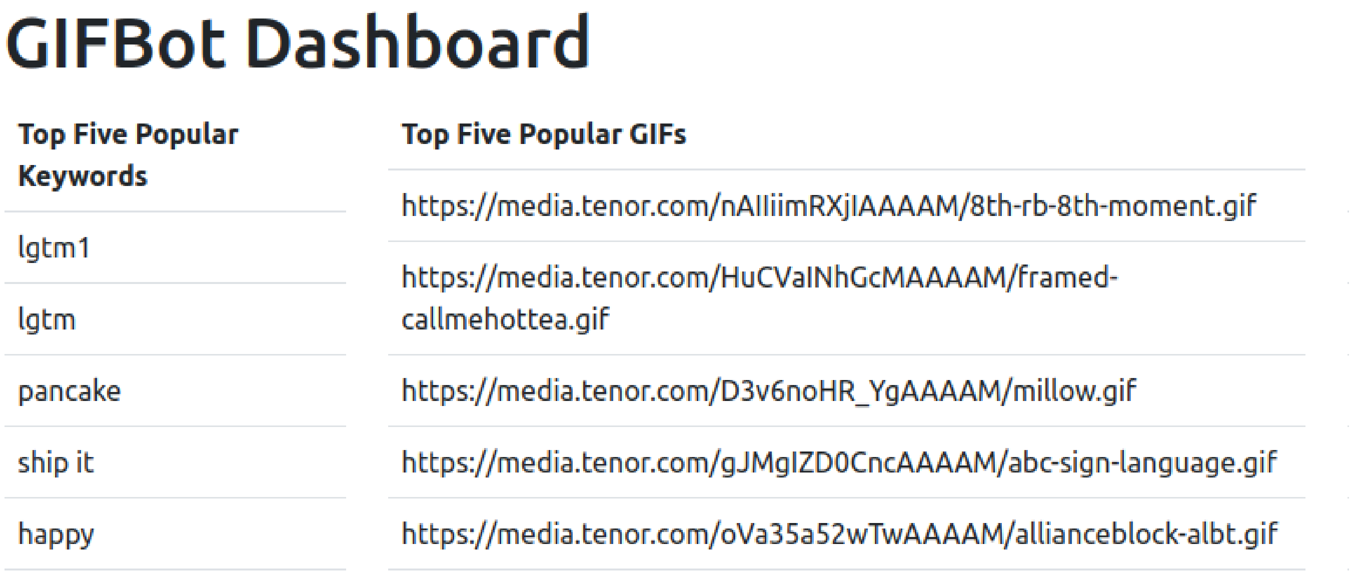
External API Latency
Average: 316.801652892562 ms
Maximum: 1644.0 ms
Minimum: 35.0 ms

Recent External API Latency(1-4) (ms)
156.0
989.0
946.0
216.0

System Log

#	Client Request Info	Client Response Info	Client Device Info	3rd Party Request Info	3rd Party Response Info
1	Document{{date=Mon Nov 21 00:00:00 UTC 2022, searchTerm=gogogo, time=Thu Jan 01 16:54:06 UTC 1970}}	Document{{date=Mon Nov 21 00:00:00 UTC 2022, responseBody=Document{{gifs=[https://media.tenor.com/EYBNOEtGi28AAAAM/go-go-cheerleader.gif, https://media.tenor.com/EKMw95fDSRUAAAAM/go-go-go-toby-ste	Document{{androidVersion=12, brand=samsung, manufacture	Document{{baseUrl=https://tenor.googleapis.com, date=Mon Nov 21 00:00:00 UTC 2022, time=T	Document{{date=Mon Nov 21 00:00:00 UTC 2022, tenorResponseBody={ "results": [{ "id": "1261092799840160623", "title": "", "media_formats": { "tinygif": { "url": "https://media.tenor.com/EYBNOEtGi28AAAAM/go-go-cheerleader.gif", "duration": 0, "preview": "", "dims": [220, 123], "size": 41050 } }, "created": 1649311921.780653, "content_description": "Go Go Cheerleader GIF", "itemurl": "https://tenor.com/view/go-go-cheerleader-priconne-princess-connect-monika-gif-25321719", "url": "https://tenor.com/bSpuZ.gif", "tags": ["Go Go", "cheerleader", "priconne", "Princess Connect", "monika", "cheer"], "flags": [], "hasaudio": false }, { "id": "1198855765771897109", "title": "", "media_formats": { "tinygif": { "url": "https://media.tenor.com/EKMw95fDSRUAAAAM/go-go-go-toby-stephen

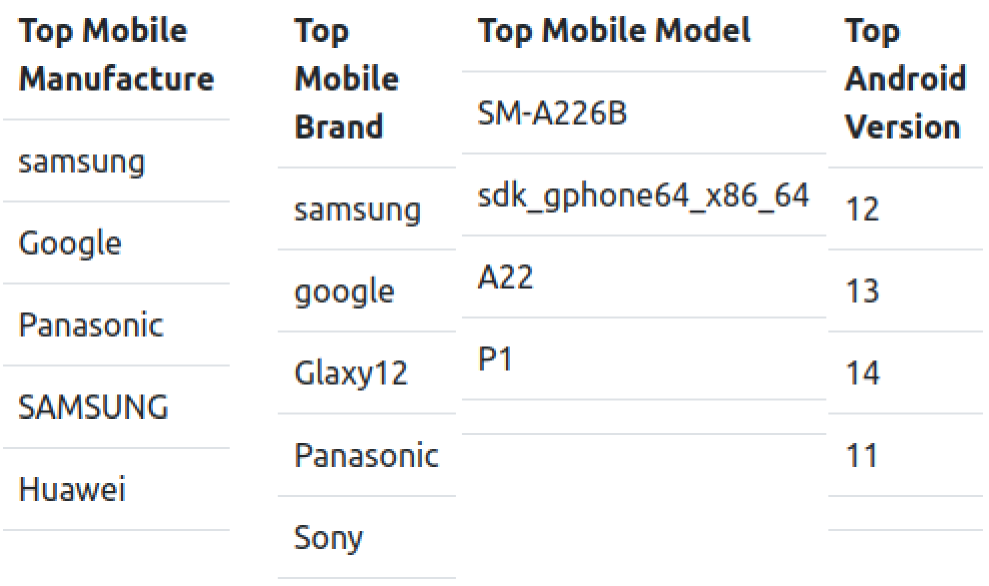
- Top Five Popular Keywords, Top Five Popular GIFs



GIFBot Dashboard

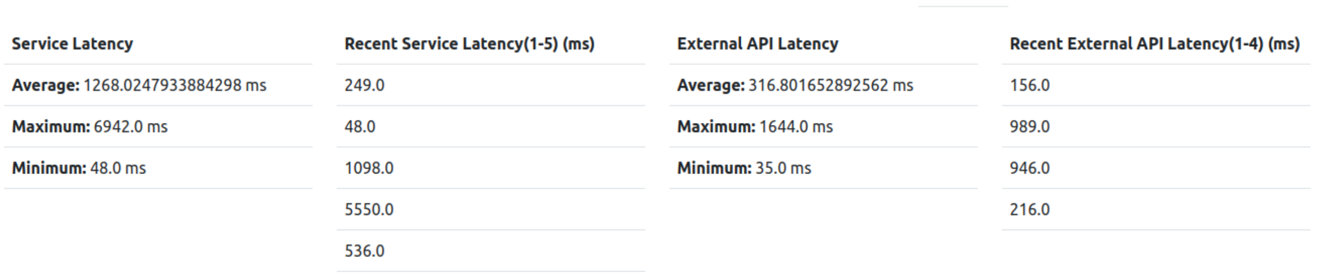
Top Five Popular Keywords	Top Five Popular GIFs
lgtm1	https://media.tenor.com/nAIliimRXjIAAAAM/8th-rb-8th-moment.gif
lgtm	https://media.tenor.com/HuCVaINhGcMAAAAM/framed-callmehottea.gif
pancake	https://media.tenor.com/D3v6noHR_YgAAAAM/millow.gif
ship it	https://media.tenor.com/gJMgIZD0CncAAAAM/abc-sign-language.gif
happy	https://media.tenor.com/oVa35a52wTwAAAAM/allianceblock-albt.gif

- Top Device: Top Mobile Manufacture, Top Mobile Brand, Top Mobile Model, Top Android Version



Top Mobile Manufacture	Top Mobile Brand	Top Mobile Model	Top Android Version
samsung	samsung	SM-A226B	12
Google	google	sdk_gphone64_x86_64	13
Panasonic	Glaxy12	A22	14
SAMSUNG	Panasonic	P1	11
Huawei	Sony		

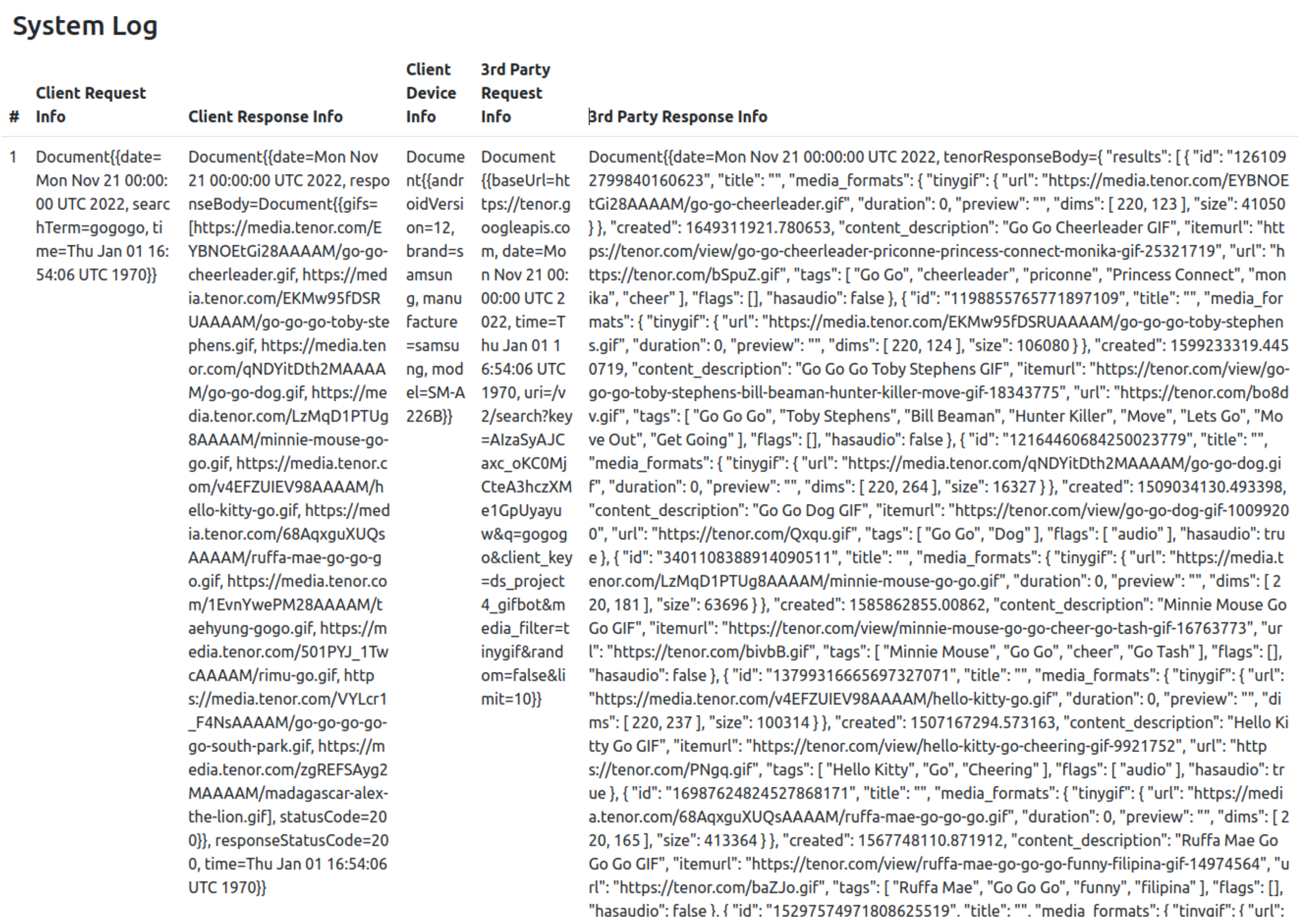
- Latency: Service Latency, Recent Service Latency(1-5) (ms), External API Latency, Recent External API Latency(1-4) (ms)



Service Latency	Recent Service Latency(1-5) (ms)	External API Latency	Recent External API Latency(1-4) (ms)
Average: 1268.0247933884298 ms	249.0	**Average:** 316.801652892562 ms	156.0
Maximum: 6942.0 ms	48.0	**Maximum:** 1644.0 ms	989.0
Minimum: 48.0 ms	1098.0	**Minimum:** 35.0 ms	946.0
	5550.0		216.0
	536.0		

c. The dashboard displays formatted full logs.

Below is a screenshot of the formatted full logs in the dashboard.

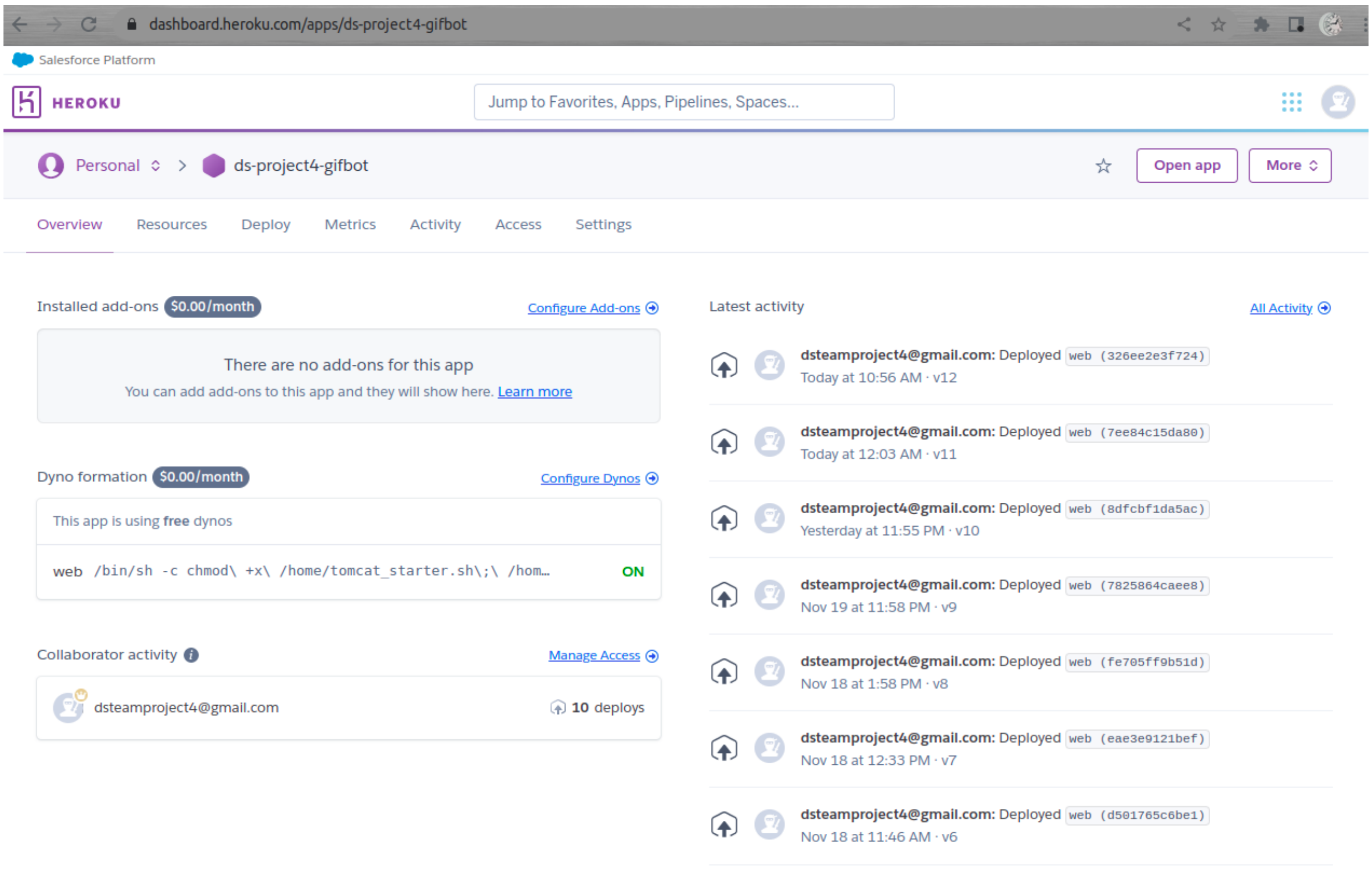


System Log

#	Client Request Info	Client Response Info	Client Device Info	3rd Party Request Info	3rd Party Response Info
1	Document{{date=Mon Nov 21 00:00:00 UTC 2022, searchTerm=gogogo, time=Thu Jan 01 16:54:06 UTC 1970}}	Document{{date=Mon Nov 21 00:00:00 UTC 2022, responseBody=Document{{gifs=[https://media.tenor.com/EYBNOEtGi28AAAAM/go-go-cheerleader.gif, https://media.tenor.com/EKMw95fDSRUAAAAM/go-go-go-toby-stephens.gif, https://media.tenor.com/qNDYitDth2MAAAAM/go-go-dog.gif, https://media.tenor.com/LzMqD1PTUg8AAAAM/minnie-mouse-go-go.gif, https://media.tenor.com/v4EFZUIEV98AAAAM/hello-kitty-go.gif, https://media.tenor.com/68AqxguXUQsAAAAM/ruffa-mae-go-go-go.gif, https://media.tenor.com/1EvnYwePM28AAAAM/taehyung-gogo.gif, https://media.tenor.com/501PYJ_1TwcAAAAM/rimu-go.gif, https://media.tenor.com/VYLcr1_F4NsAAAAM/go-go-go-go-go-south-park.gif, https://media.tenor.com/zgREFSAyg2MAAAAM/madagascar-alex-the-lion.gif], statusCode=200}}, responseStatusCode=200, time=Thu Jan 01 16:54:06 UTC 1970}}	Document{{androidVersion=12, brand=samsung, manufacture=samsung, model=SM-A226B}}	Document{{baseUrl=https://tenor.googleapis.com, date=Mon Nov 21 00:00:00 UTC 2022, time=Thu Jan 01 16:54:06 UTC 1970, uri=/v2/search?key=AIzaSyAJCaxc_oKC0MjCteA3hczXMe1GpUyayuw&q=gogogo&client_key=ds_project4_gifbot&media_filter=tinygif&random=false&limit=10}}	Document{{date=Mon Nov 21 00:00:00 UTC 2022, tenorResponseBody={ "results": [{ "id": "1261092799840160623", "title": "", "media_formats": { "tinygif": { "url": "https://media.tenor.com/EYBNOEtGi28AAAAM/go-go-cheerleader.gif", "duration": 0, "preview": "", "dims": [220, 123], "size": 41050 } }, "created": 1649311921.780653, "content_description": "Go Go Cheerleader GIF", "itemurl": "https://tenor.com/view/go-go-cheerleader-priconne-princess-connect-monika-gif-25321719", "url": "https://tenor.com/bSpuZ.gif", "tags": ["Go Go", "cheerleader", "priconne", "Princess Connect", "monika", "cheer"], "flags": [], "hasaudio": false }, { "id": "1198855765771897109", "title": "", "media_formats": { "tinygif": { "url": "https://media.tenor.com/EKMw95fDSRUAAAAM/go-go-go-toby-stephens.gif", "duration": 0, "preview": "", "dims": [220, 124], "size": 106080 } }, "created": 1599233319.4450719, "content_description": "Go Go Go Toby Stephens GIF", "itemurl": "https://tenor.com/view/go-go-go-toby-stephens-bill-beaman-hunter-killer-move-gif-18343775", "url": "https://tenor.com/bo8dv.gif", "tags": ["Go Go Go", "Toby Stephens", "Bill Beaman", "Hunter Killer", "Move", "Lets Go", "Move Out", "Get Going"], "flags": [], "hasaudio": false }, { "id": "12164460684250023779", "title": "", "media_formats": { "tinygif": { "url": "https://media.tenor.com/qNDYitDth2MAAAAM/go-go-dog.gif", "duration": 0, "preview": "", "dims": [220, 264], "size": 16327 } }, "created": 1509034130.493398, "content_description": "Go Go Dog GIF", "itemurl": "https://tenor.com/view/go-go-dog-gif-10099200", "url": "https://tenor.com/Qxqu.gif", "tags": ["Go Go", "Dog"], "flags": ["audio"], "hasaudio": true }, { "id": "3401108388914090511", "title": "", "media_formats": { "tinygif": { "url": "https://media.tenor.com/LzMqD1PTUg8AAAAM/minnie-mouse-go-go.gif", "duration": 0, "preview": "", "dims": [220, 181], "size": 63696 } }, "created": 1585862855.00862, "content_description": "Minnie Mouse Go Go GIF", "itemurl": "https://tenor.com/view/minnie-mouse-go-go-cheer-go-tash-gif-16763773", "url": "https://tenor.com/bivbB.gif", "tags": ["Minnie Mouse", "Go Go", "cheer", "Go Tash"], "flags": [], "hasaudio": false }, { "id": "13799316665697327071", "title": "", "media_formats": { "tinygif": { "url": "https://media.tenor.com/v4EFZUIEV98AAAAM/hello-kitty-go.gif", "duration": 0, "preview": "", "dims": [220, 237], "size": 100314 } }, "created": 1507167294.573163, "content_description": "Hello Kitty Go GIF", "itemurl": "https://tenor.com/view/hello-kitty-go-cheering-gif-9921752", "url": "https://tenor.com/PNgq.gif", "tags": ["Hello Kitty", "Go", "Cheering"], "flags": ["audio"], "hasaudio": true }, { "id": "16987624824527868171", "title": "", "media_formats": { "tinygif": { "url": "https://media.tenor.com/68AqxguXUQsAAAAM/ruffa-mae-go-go-go.gif", "duration": 0, "preview": "", "dims": [220, 165], "size": 413364 } }, "created": 1567748110.871912, "content_description": "Ruffa Mae Go Go Go GIF", "itemurl": "https://tenor.com/view/ruffa-mae-go-go-go-funny-filipina-gif-14974564", "url": "https://tenor.com/baZJo.gif", "tags": ["Ruffa Mae", "Go Go Go", "funny", "filipina"], "flags": [], "hasaudio": false }, { "id": "15297574971808625519", "title": "", "media_formats": { "tinygif": { "url":

7. Deploy the web service to Heroku

Below is a screenshot of the application deployment on the Heroku.



Installed add-ons $0.00/month — Configure Add-ons

There are no add-ons for this app
You can add add-ons to this app and they will show here. Learn more

Dyno formation $0.00/month — Configure Dynos

This app is using **free** dynos

```
web  /bin/sh -c chmod\ +x\ /home/tomcat_starter.sh\;\ /hom...   ON
```

Collaborator activity — Manage Access

dsteamproject4@gmail.com — 10 deploys

Latest activity — All Activity

dsteamproject4@gmail.com: Deployed `web (326ee2e3f724)`
Today at 10:56 AM · v12

dsteamproject4@gmail.com: Deployed `web (7ee84c15da80)`
Today at 12:03 AM · v11

dsteamproject4@gmail.com: Deployed `web (8dfcbf1da5ac)`
Yesterday at 11:55 PM · v10

dsteamproject4@gmail.com: Deployed `web (7825864caee8)`
Nov 19 at 11:58 PM · v9

dsteamproject4@gmail.com: Deployed `web (fe705ff9b51d)`
Nov 18 at 1:58 PM · v8

dsteamproject4@gmail.com: Deployed `web (eae3e9121bef)`
Nov 18 at 12:33 PM · v7

dsteamproject4@gmail.com: Deployed `web (d501765c6be1)`
Nov 18 at 11:46 AM · v6

In this project, we use the Github to do the code version control and team collabration. For the Heroku deployment, the project uses the Github Actions to do the continuous delivery. For each merge to the main branch, the deployment will be triggered and run automatically.

- Dockerfile for Heroku

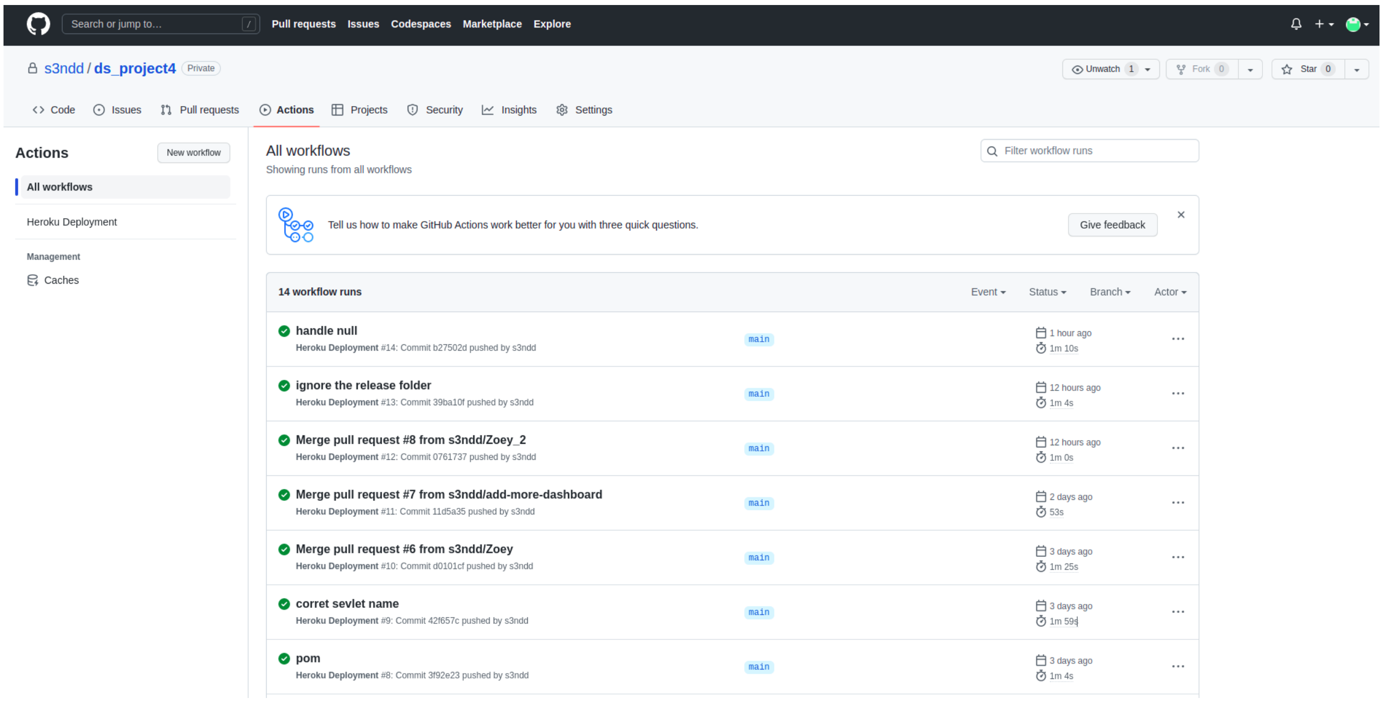
```
# Use Tomcat 10, which supports Servlets 5
FROM tomcat:10.1.0-M5-jdk16-openjdk-slim-bullseye

# This provides better support for running the JVM in a container
ENV JAVA_OPTS="-Xmx300m"

# Expose port 8080 when running on localhost
EXPOSE 8080

# Copy in our ROOT.war to the right place in the container
COPY target/Project4Task2WebService-1.0-SNAPSHOT.war
/usr/local/tomcat/webapps/

# LOCALHOST:  Run catalina in the container
# Comment the next line out if running on Heroku
#CMD ["catalina.sh", "run"]

# HEROKU: Run catalina in a container on Heroku
# Copy tomcat_starter.sh to the container; will executed when deployed
# Comment the next two lines if running on Localhost
COPY heroku/tomcat_starter.sh /home/
CMD chmod +x /home/tomcat_starter.sh; /home/tomcat_starter.sh
```

- Code for Github workflow

```
name: Heroku Deployment
on:
  push:
    branches: [ main ]
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Git Checkout
        uses: actions/checkout@v1
      - name: Set up JDK 11
        uses: actions/setup-java@v1
        with:
```

```
     java-version: 11
   - name: Build with Maven
    run: mvn -B package --file Project4Task2WebService/pom.xml
   - name: Heroku Container Registry login
    env:
     HEROKU_API_KEY: ${{ secrets.HEROKU_API_KEY }}
    working-directory: Project4Task2WebService/heroku
    run: heroku container:login
   - name: Build and push
    env:
     HEROKU_API_KEY: ${{ secrets.HEROKU_API_KEY }}
    working-directory: Project4Task2WebService/heroku
    run: heroku container:push -a ${{ secrets.HEROKU_APP_NAME }} web --
context-path=..
   - name: Release
    env:
     HEROKU_API_KEY: ${{ secrets.HEROKU_API_KEY }}
    working-directory: Project4Task2WebService/heroku
    run: heroku container:release -a ${{ secrets.HEROKU_APP_NAME }} web
```

- Screenshot for Github Actions



All workflows
Showing runs from all workflows

Tell us how to make GitHub Actions work better for you with three quick questions.

14 workflow runs

handle null
Heroku Deployment #14: Commit b27502d pushed by s3ndd — main — 1 hour ago — 1m 10s

ignore the release folder
Heroku Deployment #13: Commit 39ba10f pushed by s3ndd — main — 12 hours ago — 1m 4s

Merge pull request #8 from s3ndd/Zoey_2
Heroku Deployment #12: Commit 0761737 pushed by s3ndd — main — 12 hours ago — 1m 0s

Merge pull request #7 from s3ndd/add-more-dashboard
Heroku Deployment #11: Commit 11d5a35 pushed by s3ndd — main — 2 days ago — 53s

Merge pull request #6 from s3ndd/Zoey
Heroku Deployment #10: Commit d0101cf pushed by s3ndd — main — 3 days ago — 1m 25s

corret sevlet name
Heroku Deployment #9: Commit 42f657c pushed by s3ndd — main — 3 days ago — 1m 59s

pom
Heroku Deployment #8: Commit 3f92e23 pushed by s3ndd — main — 3 days ago — 1m 4s