

## **BiLingualy - Language Translation Application**

Name: Kristen Dmello

Andrewid: kdmello

### **Description:**

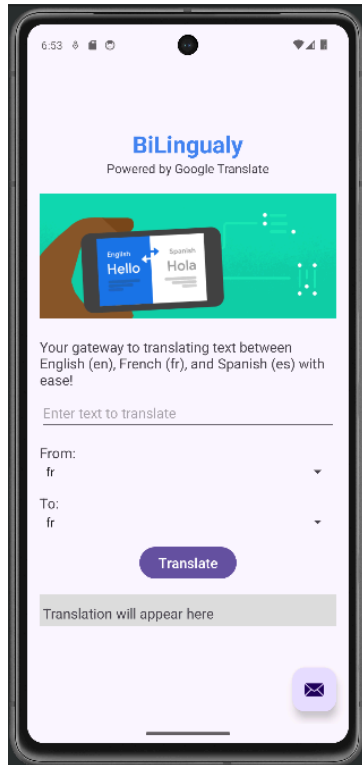
My app 'BiLingualy', serves as a powerful yet user-friendly translation tool, enabling users to seamlessly translate text between English, French, and Spanish. At its core, 'BiLingualy' harnesses the capabilities of the RapidAPI endpoint, a robust platform that facilitates the fetching of accurate JSON data for translated text. Upon inputting the text they wish to translate, users select their source and target languages from the intuitive dropdown menus labeled "From:" and "To:". With just a tap on the "Translate" button, the app efficiently processes their request, retrieves the translation in real-time, and elegantly displays the result.

### **1. Implement a Native Android Application:**

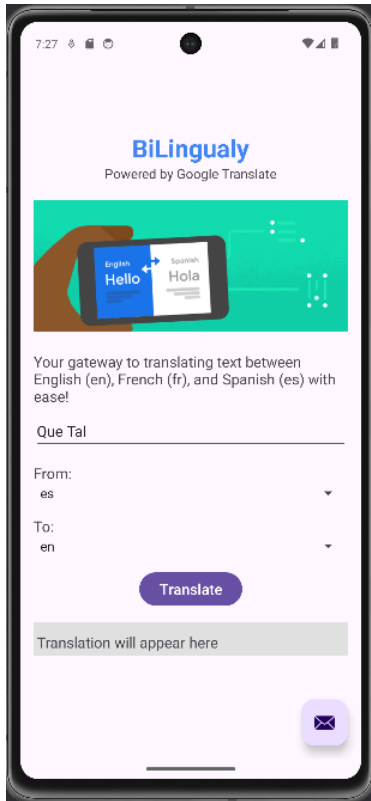
The name of my web application project is: TranslateApplication

a. Has at least three different kinds of Views in your Layout

- It has three different kinds of views in the Layout (Eg. TextView, EditText, Button, Spinner and ImageView)
- The app incorporates an EditText for input, an ImageView with a static image, Spinners for dropdowns, a Button for submission, and TextViews for displaying translations, all within a seamless Android XML layout managed by a CoordinatorLayout for smooth scrolling and interaction. This uses HTML and CSS-like styling to implement the above.
- See activity\_main.xml to understand how this has been implemented



b. Requires 3 inputs from the user: Text to translate, Source Language (in 'from' dropdown) and Target language (in 'to' dropdown)



c. Makes an HTTP request (using an appropriate HTTP method) to your web service  
My application performs an HTTP GET request using JavaScript's fetch API. The request format is:

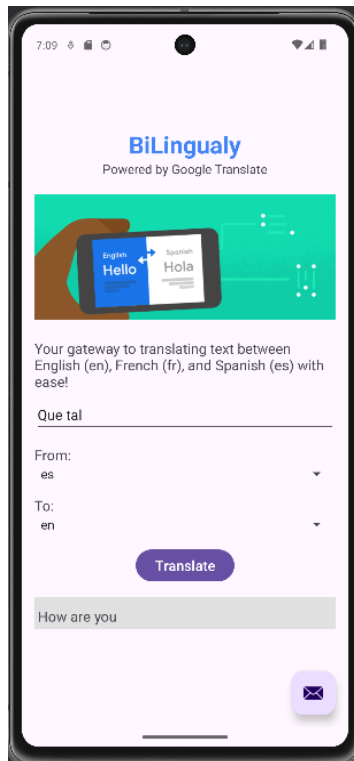
```
"https://expert-potato-wrr6rv5rvwq429vp6-8080.app.github.dev/translate?text=userSearch&source=sourceLang&target=targetLang"
```

where userSearch, sourceLang and targetLang are entered by the user.  
The application makes this request to the web service, parses the returned JSON to extract the translated text, and displays it to the user.

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

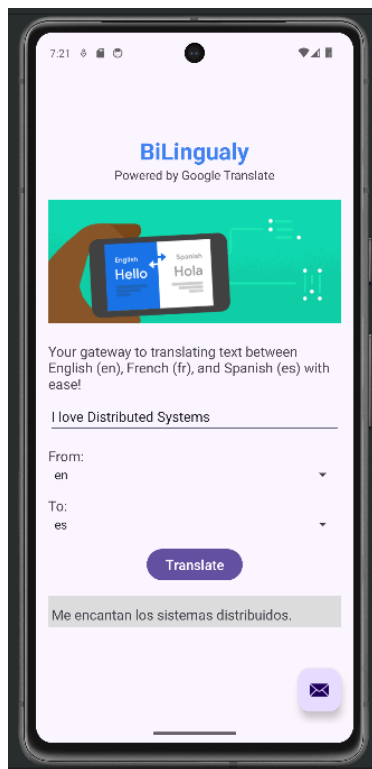
```
responseContent : "{\"data\":{\"translations\":[{\"translatedText\":\"how are you\"}]}}"
```

e. Displays new information to the user



f. Is repeatable

The application can be used:



## 2. Implement a web service

The URL of my web service deployed to Codespaces is:

<https://expert-potato-wrr6rv5rvwg429vp6-8080.app.github.dev/dashboard>

*(Note: In case it does not work, we might have to create a new codespace and update the url in the android app. This issue has occurred twice in the past.)*

a. Implement a simple (can be a single path) API.

Controller: TranslationServlet.java and DashboardServlet.java *(also includes model logic)*

View: index.jsp, dashboard.jsp

b. Receives an HTTP request from the native Android application

TranslationServlet.java receives the HTTP GET request with the parameters "text", "sourceLang", and "targetLang", and conveys this string to the API to fetch the translated text.

c. Executes business logic appropriate to your application. This includes fetching XML or JSON information from some 3rd party API and processing the response.

It makes an HTTP request to: <https://google-translate1.p.rapidapi.com/language/translate/v2> and appends the below information to the request. The encodedParams are the three user inputs.

Request format:

```
HttpRequest request = HttpRequest.newBuilder()
    .uri(URI.create(TRANSLATE_URL))
    .header(name: "content-type", value: "application/x-www-form-urlencoded")
    .header(name: "Accept-Encoding", value: "application/gzip")
    .header(name: "X-RapidAPI-Key", API_KEY)
    .header(name: "X-RapidAPI-Host", HOST)
    .POST(HttpRequest.BodyPublishers.ofString(encodedParams.toString()))
    .build();
```

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 JSON formatted response.

```
responseContent : "{\"data\":{\"translations\":[{\"translatedText\":\"how are you\"}]}}"
```

Corresponding logic in Android:

```
String translatedText =
jsonObject.getJSONObject("data").getJSONArray("translations").getJSONObject(0).
getString("translatedText");
```

### **3. Handled error conditions in my app**

#### **4. Logging: Below is the information that is being logged by the application:**

- Request Content: This is the text submitted for translation by the user. Helps in analyzing which features or aspects of the service are most used or if certain phrases are commonly translated, which can inform business and development strategies.
- User Agent: This identifies the client software that made the request, which in this case appears to be an Android application running on a 64-bit emulator or device. Can help determine the type of devices and software versions that are accessing the service, which can guide optimization for the most common clients.
- IP Address: This records the IP address from which the request was made. The address 0:0:0:0:0:0:0:1 is the IPv6 loopback address, indicating a request made from the same machine that hosts the server (often used in testing). IP addresses can often be mapped to geographic locations, allowing for analysis of service usage by region.
- Request Timestamp: The date and time when the request was received by the server. Helps in identifying peak usage times which can be valuable for resource allocation and scaling decisions. Provides a timeline of service usage which is critical for resolving issues that are reported after the fact.
- Response Timestamp: The date and time when the response was sent back to the client. Helps in verifying compliance with SLAs that may require responses within a certain timeframe.
- Duration: The time taken to process the request and generate a response, measured in milliseconds. Measures how long the server takes to process requests, which is vital for identifying performance bottlenecks. Long durations could indicate problems that affect the user experience, and thus are important for ensuring high-quality service.

#### **5. Store the log information in a database**

Snapshot of database in MongoDB:

KRISTEN'S ORG - 2024-04-01 > PROJECT 0 > DATABASES

**ClusterO** VERSION 7.0.8 REGION AWS N. Virginia (us-east-1)

Overview Real Time Metrics **Collections** Atlas Search Profiler Performance Advisor Online Archive

DATABASES: 2 COLLECTIONS: 7 VISUALIZE YOUR DATA REFRESH

+ Create Database

Search Namespaces

sample\_mflix

userTranslateInputDb

Translate

**userTranslateInputDb.Translate**

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 10.65KB TOTAL DOCUMENTS: 28 INDEXES TOTAL SIZE: 36KB

Find Indexes Schema Anti-Patterns Aggregation Search Indexes

INSERT DOCUMENT

Filter Type a query: { field: 'value' } Reset Apply Options

```

_id: ObjectId('6610d4ea6cba3532109e808d')
requestContent: "my favourite course is DS"
responseContent: "{\"data\":{\"translations\":[{\"translatedText\":\"mi curso favorito es DS\"}]}"
ipAddress: "0:0:0:0:0:0:1"
requestTimestamp: "2024-04-06 04:51:54"
responseTimestamp: "2024-04-06 04:51:54"

```

```

// Initialize MongoDB client
String connectionString =
"mongodb+srv://kdmello:1nlJrNsxSw9BgvOk@cluster0.eqb6ctl.mongodb.net/?retryWrites=true&w=majority&appName=ClusterO";
mongoClient = MongoClient.create(connectionString);
database = mongoClient.getDatabase("userTranslateInputDb");
collection = database.getCollection("Translate");

```

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

<https://expert-potato-wrr6rv5rvwq429vp6-8080.app.github.dev/dashboard>

1. Total Number of Requests: The app has processed 27 individual translation requests since the tracking began. Tracks user engagement with the app, indicating its popularity and usage patterns—essential for capacity planning and scaling.
2. Total Time App Has Run (ms): The cumulative time the app has been operational and processing requests is 70,123,000 milliseconds. Measures the app's workload over time, helping assess server utilization and operational costs.
3. Most Frequent Request: The phrase "I love Distributed Systems" is the text that users have requested to be translated most often. Identifies trends in user behavior, which can inform updates and feature development to enhance user satisfaction.

## App Analytics

Total Number of Requests	27
Total Time App Has Run (ms)	70123000
Most Frequent Request	I love Distributed Systems

## Translation Service Logs

Request Content	User Agent	IP Address	Request Timestamp	Response Timestamp	Duration
my favourite course is DS	Dalvik/2.1.0 (Linux; U; Android 14; sdk_gphone64_x86_64 Build/UE1A.230829.036.A1)	0:0:0:0:0:0:0:	2024-04-06 04:51:54	2024-04-06 04:51:54	512
I am working on a project	Dalvik/2.1.0 (Linux; U; Android 14; sdk_gphone64_x86_64 Build/UE1A.230829.036.A1)	0:0:0:0:0:0:1	2024-04-06 06:16:19	2024-04-06 06:16:20	837
Que bien que te gusto	Dalvik/2.1.0 (Linux; U; Android 14; sdk_gphone64_x86_64 Build/UE1A.230829.036.A1)	0:0:0:0:0:0:0:	2024-04-06 06:29:05	2024-04-06 06:29:05	172
Que bien	Dalvik/2.1.0 (Linux; U; Android 14; sdk_gphone64_x86_64 Build/UE1A.230829.036.A1)	0:0:0:0:0:0:1	2024-04-06 06:29:29	2024-04-06 06:29:29	148
No fumador por favor	Dalvik/2.1.0 (Linux; U; Android 14; sdk_gphone64_x86_64 Build/UE1A.230829.036.A1)	0:0:0:0:0:0:0:	2024-04-06 06:29:44	2024-04-06 06:29:45	198
el nino es muy pico	Dalvik/2.1.0 (Linux; U; Android 14; sdk_gphone64_x86_64 Build/UE1A.230829.036.A1)	0:0:0:0:0:0:1	2024-04-06 06:30:12	2024-04-06 06:30:12	170
que tal	Dalvik/2.1.0 (Linux; U; Android 14; sdk_gphone64_x86_64 Build/UE1A.230829.036.A1)	0:0:0:0:0:0:0:	2024-04-06 06:31:14	2024-04-06 06:31:14	154
La vie est belle	Dalvik/2.1.0 (Linux; U; Android 14; sdk_gphone64_x86_64 Build/UE1A.230829.036.A1)	0:0:0:0:0:0:1	2024-04-06 06:31:55	2024-04-06 06:31:55	156
Le monde est petit	Dalvik/2.1.0 (Linux; U; Android 14; sdk_gphone64_x86_64 Build/UE1A.230829.036.A1)	0:0:0:0:0:0:1	2024-04-06 06:36:42	2024-04-06 06:36:43	204
Le monde est petit	Dalvik/2.1.0 (Linux; U; Android 14; sdk_gphone64_x86_64 Build/UE1A.230829.036.A1)	0:0:0:0:0:0:1	2024-04-06 06:36:55	2024-04-06 06:36:55	135