### **AREA**

This project is built using Docker, ensuring a streamlined development and deployment process. The project structure is organized into distinct components: the backend (including database and API access), the website, and the mobile application. Each component has its own Dockerfile located within its respective directory, and there is a docker-compose.yaml file at the root of the project.

The docker-compose.yaml file orchestrates the build process by referencing the Dockerfiles in the following directories: /Backend, /FrontendAPP, and /FrontendWeb. It builds the entire project and sets up the server. You can launch it via the executable file launch\_docker.sh.

The individual Dockerfiles are responsible for constructing each component (/Backend, /FrontendAPP, and /FrontendWeb) to enable the docker-compose.yaml to successfully launch the entire project.

# **Backend Project:**

## Description

This project manages users, makes calls to third-party APIs such as Google, Spotify, OpenWeather, Twitch, Github and NewsAPI, and serves data to the frontend. Authentication is handled via JSON Web Tokens (JWT) and Supabase. Swagger is used for API documentation.

### **Table of Contents**

- Installation
- Starting the Project
- Environment Variables
- API Documentation
- Project Structure
- Useful Commands

### Installation

To install and run the Backend locally:

Clone the repository:

→ AREA (dev) git clone git@github.com:EpitechPromo2027/B-DEV-500-TLS-5-1-area-anastasia.bouby.git cd B-DEV-500-TLS-5-1-area-anastasia.bouby/Backend

Install the dependencies:

○→ Backend (dev) npm install

# Starting the project

To start the project, use the following command:

```
○ → Backend (dev) npm run dev
```

The server will run by default on port 8080.

#### **Environnement variables**

The project requires certain environment variables to function correctly.

Create a .env file at the root of the Backend directory with the following content:

```
REACT_APP_SUPABASE_URL="https://qkqqhtvrxplbxmvikcdy.supabase.co"
REACT APP SUPABASE ANON KEY="your-supabase-anon-key"
GOOGLE_API_KEY="your-google-api-key"
NEWSAPI API KEY="your-newsapi-api-key"
OPENWEATHER_KEY="your-openweather-api-key"
GOOGLE_CLIENT_ID="your-google-client-id"
GOOGLE_CLIENT_SECRET="your-google-client-secret"
SPOTIFY_CLIENT_ID="your-spotify-client-id"
SPOTIFY_CLIENT_SECRET="your-spotify-client-secret"
DISCORD_CLIENT_ID="your-discord-client-id"
DISCORD_CLIENT_SECRET="your-discord-client-secret"
DISCORD_TOKEN="your-discord-token"
GUILD_ID="1288150792301183007"
TWITCH_CLIENT_ID="your-twitch-client-id"
TWITCH_CLIENT_SECRET="your-twitch-client-secret"
GITHUB_CLIENT_ID="your-github-client-id"
GITHUB_CLIENT_SECRET="your-github-client-secret"
SECRET=AREA
```

# **API** Documentation

The complete API documentation can be accessed via Swagger. Once the server is running, navigate to <a href="http://localhost:8080/api-docs">http://localhost:8080/api-docs</a> to explore the available endpoints and their usage.

# Project structure

Here is the structure of the Backend:

```
src
  - API
      discord

    Discord.query.ts

         — Discord.ts
       github
        └─ Github.ts
       google
         Google.query.ts
        Google.ts
       spotify
        L Spotify.ts
       twitch
        └─ Twitch.ts
   config
    └─ db.ts
   index.ts
   middleware
    └─ auth.ts
   routes
         area.query.ts
        __ area.ts
      login
         — login.query.ts
— login.ts
       register
          register.query.ts
        register.ts
       services
          services.query.ts
        __ services.ts
       users
         user.query.ts
         — users.ts
       api.test.ts
       auth.test.ts
       service.test.ts
       users.test.ts
    └─ test.server.ts
```

#### **Useful Commands**

- Build the project: npm run buildStart the server: npm run start
- Run the project in development mode: npm run dev
- Lint the code: npm run lint
- Generate Swagger Documentation: Swagger documentation will be updated automatically if you follow the right format in your route comments.

## **Frontend web project:**

#### Description:

This project is the Frontend part of the AREA project. It provides a user interface to define and manage automation rules (actions and reactions), while ensuring a smooth user experience for viewing and tracking ongoing automations.

#### Table of Contents:

- Installation
- Starting the Project
- Project Structure
- Useful Commands

#### Installation:

To install and run the frontend web locally:

1. Prerequisites

Before getting started, make sure you have the following installed on your machine:

- Node.js (version 14.18+ or 16+ recommended)
- npm (comes with Node.js) or yarn.

2. Clone the repository:

```
    AREA (dev) git clone git@github.com:EpitechPromo2027/B-DEV-500-TLS-5-1-area-anastasia.bouby.git cd B-DEV-500-TLS-5-1-area-anastasia.bouby/FrontendWeb
    Install the dependencies:
```

```
o→ FrontendWeb (dev) npm install
or
o→ FrontendWeb (dev) npm install yarn
```

#### Starting the project

1. To start the project, use the following command:

```
or

→ FrontendWeb (dev) npm run dev

or

→ FrontendWeb (dev) yarn dev
```

The server will run by default on port 8081.

2. If you want build the project and after start it independently:

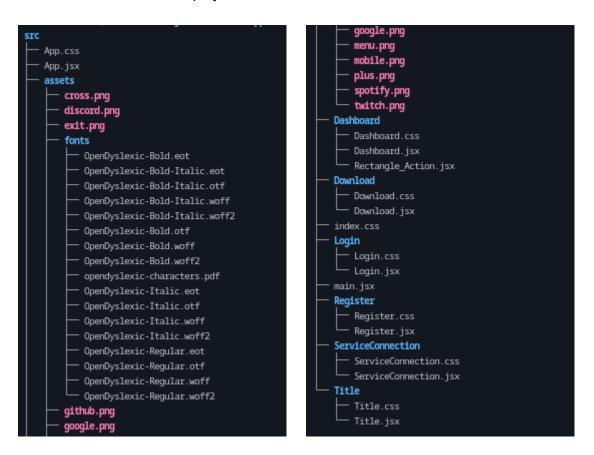
```
ProntendWeb (dev) npm run build
or

FrontendWeb (dev) yarn build
and:

FrontendWeb (dev) npm run preview
or
```

#### Project structure

Here is the structure of the project:



#### **Useful Commands**

- Build the project: npm run build
- Start the server: npm run start
- Run the project in development mode: npm run dev
- Lint the code: npm run lint

# **Frontend App Project:**

## Description

This project is the Frontend part of the AREA project. It provides a user interface to define and manage automation rules (actions and reactions), while ensuring a smooth user experience for viewing and tracking ongoing automations.

### **Table of Contents**

- Installation
- Starting the Project
- Project Structure
- Useful Commands

### Installation

To install and run the frontend web locally:

1. Prerequisites

Before getting started, make sure you have the following installed on your machine:

- Node.js (version 14.18+ or 16+ recommended)
- npm (comes with Node.js) or yarn.
- 2. Clone the repository:

```
→ AREA (dev) git clone git@github.com:EpitechPromo2027/B-DEV-500-TLS-5-1-area-anastasia.bouby.git cd B-DEV-500-TLS-5-1-area-anastasia.bouby/FrontendApp
```

- 3. Install the dependencies:
  - → FrontendApp (dev) npm install

# Starting the project

1. To start the project and open it in android studio, use the following command:

```
○ → FrontendApp (dev) ionic capacitor build android
```

2. To start the project and open it in the web browser, use the following command:

```
○ → FrontendApp (dev) ionic serve
```

The server will run by default on port 8100.

# Project structure

Here is the structure of the project:

```
login.service.ts
                                                              register
 app.canActivate.ts
                                                                  register.service.ts
 app.component.html
                                                              token
 app.component.scss

    token.service.ts

 app.component.spec.ts
                                                              utils
 app.component.ts
                                                              utils.service.ts
 app.module.ts
                                                      assets
 app.redirectGard.ts
                                                        discord.png
app-routing.module.ts
                                                          fonts
 dashboard

    OpenDyslexic-Bold.eot

    dashboard.module.ts

    OpenDyslexic-Bold-Italic.eot

    dashboard.page.html

    OpenDyslexic-Bold-Italic.otf

    dashboard.page.scss

                                                             OpenDyslexic-Bold-Italic.woff

    dashboard.page.spec.ts

                                                             OpenDyslexic-Bold-Italic.woff2

    dashboard.page.ts

                                                             OpenDyslexic-Bold.otf
   — dashboard-routing.modules.ts
                                                             OpenDyslexic-Bold.woff
- empty.component.ts
                                                             OpenDyslexic-Bold.woff2
 login
                                                             opendyslexic-characters.pdf

    login.module.ts

                                                             OpenDyslexic-Italic.eot

    login.page.html

                                                             OpenDyslexic-Italic.otf

    login.page.scss

                                                             OpenDyslexic-Italic.woff
   — login.page.spec.ts
                                                             OpenDyslexic-Italic.woff2
    - login.page.ts
                                                             OpenDyslexic-Regular.eot

    login-routing.module.ts

 register
                                                             OpenDyslexic-Regular.otf
                                                             OpenDyslexic-Regular.woff
   register.module.ts
                                                              OpenDyslexic-Regular.woff2

    register.page.html

                                                          github.png

    register.page.scss

                                                          google.png
   register.page.spec.ts
                                                          spotify.png
    register.page.ts
   register-routing.module.ts
                                                          twitch.png
                                                      environments
 service
                                                         environment.prod.ts

    service.module.ts

                                                          environment.ts

    service.page.html

                                                     global.scss

    service.page.scss

    service.page.spec.ts

                                                      index.html
     service.page.ts
                                                      main.ts
                                                     polyfills.ts

    service-routing.module.ts

 services
                                                      theme
      area.service.ts
                                                         variables.scss
                                                      utils
     localStorage
                                                        - ApiService.ts
      localStorage.service.ts
                                                      zone-flags.ts
```

#### **Useful Commands**

- Build the project: npm run build
- Start the server: npm run start
- Run the project in android studio: ionic capacitor build android
- Run the project in the web browser: ionic serve
- Lint the code: npm run lint

# **Service Project:**

The Service project is responsible for managing all actions and reactions within the application.

#### **Table of Contents**

- Installation
- Starting the Project
- Project Structure
- Useful Commands

## Installation

To install and run the service locally:

1. Prerequisites

Before getting started, make sure you have the following installed on your machine:

- Node.js (version 14.18+ or 16+ recommended)
- npm (comes with Node.js) or yarn.
- 2. Clone the repository:

```
→ AREA (dev) git clone git@github.com:EpitechPromo2027/B-DEV-500-TLS-5-1-area-anastasia.bouby.git cd B-DEV-500-TLS-5-1-area-anastasia.bouby/Service
```

3. Install the dependencies:

```
Service (dev) npm install
```

# Starting the project

1. To start the project, use the following command:

### **Environnement variables**

The project requires certain environment variables to function correctly.

Create a .env file at the root of the Service directory with the following content:

```
REACT_APP_SUPABASE_URL="https://qkqqhtvrxplbxmvikcdy.supabase.co"
REACT_APP_SUPABASE_ANON_KEY="your-supabase-anon-key"
GOOGLE_API_KEY="your-google-api-key"
NEWSAPI_API_KEY="your-newsapi-api-key"
OPENWEATHER_KEY="your-openweather-api-key"
GOOGLE_CLIENT_ID="your-google-client-id"
GOOGLE_CLIENT_SECRET="your-google-client-secret"
SPOTIFY_CLIENT_ID="your-spotify-client-id"
SPOTIFY_CLIENT_SECRET="your-spotify-client-secret"
DISCORD_CLIENT_ID="your-discord-client-id"
DISCORD_CLIENT_SECRET="your-discord-client-secret"
DISCORD_TOKEN="your-discord-token"
GUILD_ID="1288150792301183007"
TWITCH_CLIENT_ID="your-twitch-client-id"
TWITCH_CLIENT_SECRET="your-twitch-client-secret"
GITHUB_CLIENT_ID="your-github-client-id"
GITHUB_CLIENT_SECRET="your-github-client-secret"
SECRET=AREA
```

# Project structure

Here is the structure of the project:

```
src
  API
      - Discord
        discord.query.ts
         — discord.ts
       Github
         github.query.ts
         — github.ts
       gmail
         — Gmail.query.ts
         — Gmail.ts
       news
         News.query.ts
         — News.ts
       openWeather

    openWeather.query.ts

          - openWeather.ts
       spotify
         spotify.query.tsspotify.ts
       twitch
        twitch.query.ts
         — twitch.ts
   area
     area.service.ts

    service.action.ts

      service.reaction.ts
   config
    └─ db.ts
   DB
         area.query.ts
         — area.ts
       tokens
         — token.query.ts
         token.ts
   index.ts
   manageFS
    └─ manageFile.ts
```

#### **Useful Commands**

- Build the project: npm run build
- Run the project in development mode: npm run dev