E User Guide

E.1 System Requirements

E.1.1 Server and Web Application

The software was developed using the **Ubuntu 20.04.2.0 LTS (Focal Fossa)** distribution of the GNU/Linux operating system. It is advised to run the Server and Web Application modules on **GNU/Linux distributions based on Debian**. Still, the use of the Docker installation allows them to be run on other operating systems. The reader should however be aware that they have not been tested and no claims of compatibility have been made.

The Web Application has been developed and tested for **Google Chrome 89.0**. It is suggested the reader uses this browser.

E.1.2 Mobile Application

The Mobile Application runs solely on Android devices. Furthermore, such devices need to have the following characteristics:

- **Android Version 10** or above (other versions might work but have not been tested)
- Working **WiFi and Location functions** accessible through the built-in APIs
- Access to battery optimisation settings to **disable Doze and App Standby**

E.2 Installing and Running

This document will walk you through installing and running the various software modules for this work.

It assumes that you have a working version of **Docker** and **Docker Compose** running on your machine. To learn more, please consult Docker's Installation Guide [1] and Docker Compose's Installation Guide [2].

Moreover, it assumes you are running on a **GNU/Linux distribution based on Debian**. While this is not a strict requirement, it is advised.

[1] Available at `https://docs.docker.com/engine/install/` as of the 1st of April, 2021

[2] Available at `https://docs.docker.com/compose/install/` as of 1st of April, 2021

E.2.1 Installing Server and Web Application

The Server module will contain a REST API, available at `http://localhost:4683` by default. If you want to change this port, you need to edit its reference in the `Dockerfile` and `docker-compose.yml`, as well as updating the `.env` file. You can find these files under the *server* folder. Similarly, the Web Application (named *client*), will be available at `http://localhost:3000`.

Before starting, make sure no processes are running on at ports `3000`, `4683`, and `27017`. This includes stopping any existing mongod services that might be running on port `27017`[3].

If you have installed `npm` on your machine, you can kill all processes running on these ports by using the following:

```
npx kill-port 3000 4683 27017
```

Docker Installation

The installation is fairly straightforward. Assuming you have installed Docker an Docker Compose, run the following command from the root folder of the project.

The Server runs on the PM2[4] process manager. By default it will use **all** the cores available to it. To change this option, modify the `instances` property in the `ecosystem.config.js` file to the number of cores you want (e.g. `instances : 1`).

```
docker-compose up
```

The process might take a while. Please wait for the client to be running.

Non-Docker Installation

If you cannot install Docker or Docker Compose, or prefer not to, installation guides are available in the project files at `server/README.md` and `client/README.md`.

E.2.2 Installing the Android Application

The Android Application can be installed using the `app.apk` package found under the *android-app* folder, which can be manually installed on an Android Device. An article from **thecustomdroid** [5] walks through the process in detail.

[3] On Ubuntu 20.04, you can use `sudo systemctl stop mongod`

[4] Package available at `https://www.npmjs.com/package/pm2` as of 1st of April 2021

[5] Available at `https://www.thecustomdroid.com/how-to-install-apk-on-android/` as of 1st of April 2021

Non-apk Installation

In order to install the Android Application without the use of an `apk` file, you'll need to install Android Studio [6] and follow the steps described in the official documentation [7] .

E.2.3 Caching Function

The caching function is a simple script that is meant to run using the Lambda service offered by Amazon Web Services. As such, it is not meant to be run. However, if you want to use it to start the caching operation of map data, then do the following:

1. Make sure the REST API is running on port **4683**.
2. Install dependencies:

```
npm install
```

3. Then edit `index.js` and add the following line at the end of the file :

```
startRequest()
```

4. Finally, run the script by using

```
node index
```

This will send two `PATCH` requests to the REST API, which will start the caching process.

E.3 Initial Configuration

While the Dockerised install contains a configuration that will work for a demo, it can be modified. Moreover, the Mobile Application needs to be configured to work properly on startup. This section will walk through the process of doing so.

E.3.1 Server

The Docker installation contains a pre-packaged configuration in the form of an .env file. While it is useful to demo and run the application, it is important to change the values for security reasons for extensive use. An overview of the used .env values is available at Table E.1.

[6] Avalable at `https://developer.android.com/studio/install` as of 1st of April 2021

[7] Available at `https://developer.android.com/studio/run` as of 1st of April 2021

Name	Description
PORT	The server's port
DOMAIN	The server's domain
API_PREFIX	The base URL for the API
DOCKER_ENV	true if the Server has to run in a Docker container
DOCKER_DATABASE_URL	The MongoDB URL used when DOCKER_ENV is "true"
DATABASE_URL	The MongoDB URL used when not DOCKER_ENV isn't "true"
DB_NAME	The name of the database
TOKEN_KEY	The key used to generate JWTs
TOKEN_EXPIRATION_TIME	The time of expiry for JWTs
DEV_UPLOAD_TOKEN	A special token that bypasses token validation
CACHE_ON_STARTUP	Load cached features and heat-map data when first starting
GEOJSON_URL	URL of the GEOJson feature collection used to cache data

Table E.1: The .env configuration variables available for the server module

E.3.2 Web Application

The Docker installation contains a pre-packaged configuration in the form of an .env file. While it is useful to demo and run the application, it is important to change the values for security reasons for extensive use. An overview of the used .env values is available at Table E.2.

Name	Description
REACT_APP_MODE	production if running the app in production, dev if otherwise
REACT_APP_SERVER_URL_PRODUCTION	The URL of the API when REACT_APP_MODE=production
REACT_APP_SERVER_URL	The URL of the API when REACT_APP_MODE=dev

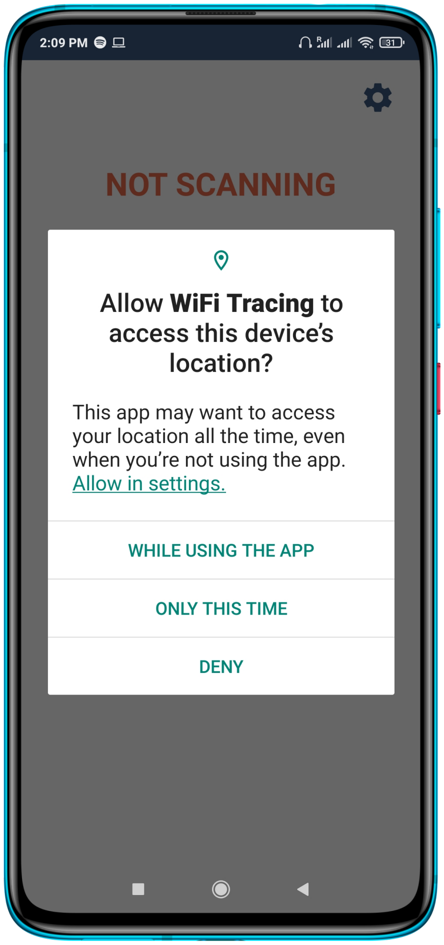
Table E.2: The .env configuration variables available for the client module

E.3.3 Mobile Application

Enabling Permissions

Upon starting the application, you will be shown several information screens. After reading through all of them you will be prompted to enable location permissions for the application. Please choose to **Allow all the time**. If you choose to only enable it **While using the app**, background collection of data will not work as designed. An example is available at Figure E.1, where the User will need to tap on "*Allow in settings*" in order to enable the necessary permissions.

The device will also need to have **Location and WiFi services enabled** in order to work scan for WiFi signals.



Allow **WiFi Tracing** to access this device's location?

This app may want to access your location all the time, even when you're not using the app. Allow in settings.

Figure E.1: Popup shown if the User hasn't given location access to the app.

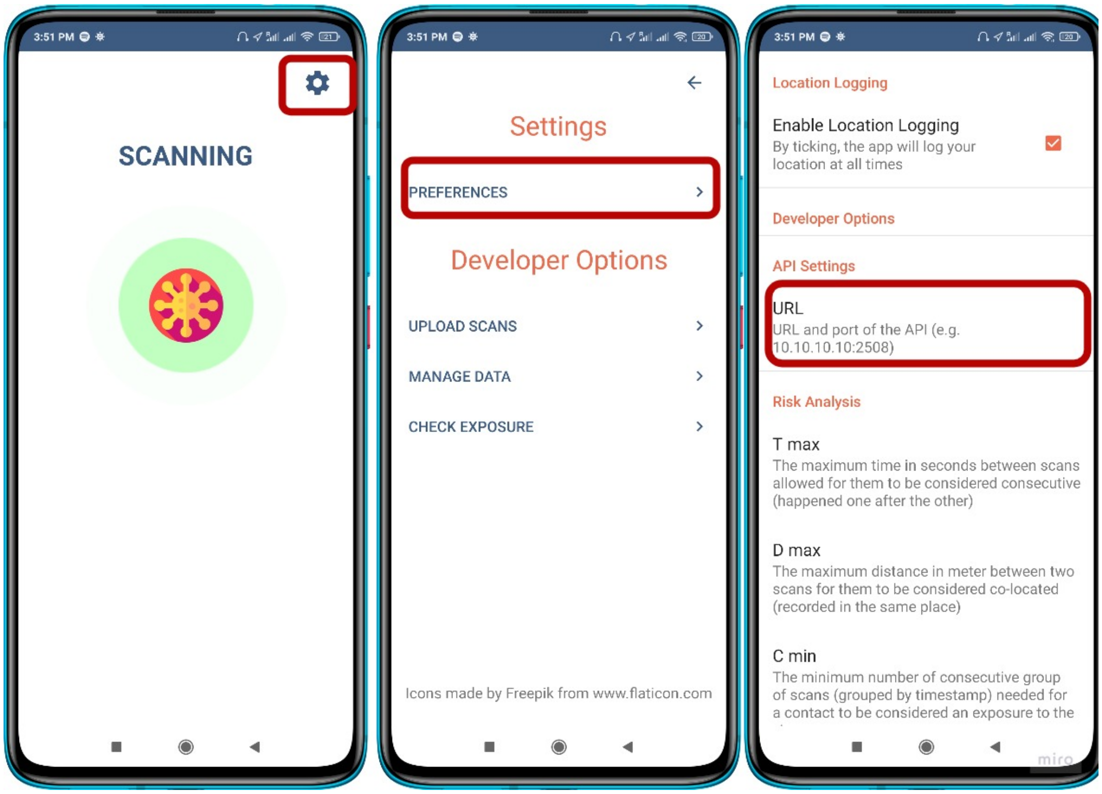
Setting Preferences

The version of the application available for install contains special Developer menus. They can be found by clicking on the gear icon on the main page.

Here, preferences can be set. You should **update the URL value** with the URL and port of the Server application as shown in Figure E.3. It is found under **Settings > Preferences > API Settings > URL**. Once set up, operations of scan uploads and risk analysis will work as designed.

Similarly, under the **Settings > Preferences > Risk Analysis** menu you will be able to **modify the constants used to perform Risk Analysis**.

It is suggested that you also **enable location logging** under **Settings > Preferences > Location Logging > Enable Location Logging**. This will allow you to upload location data to the Exposure Ingestion Service and see it visualised in the Web Application.



Location Logging

Enable Location Logging ☑
By ticking, the app will log your location at all times

Developer Options

API Settings

URL
URL and port of the API (e.g. 10.10.10.10:2508)

Risk Analysis

T max
The maximum time in seconds between scans allowed for them to be considered consecutive (happened one after the other)

D max
The maximum distance in meter between two scans for them to be considered co-located (recorded in the same place)

C min
The minimum number of consecutive group of scans (grouped by timestamp) needed for a contact to be considered an exposure to the

Figure E.2: Steps to follow to edit the URL and port of the Exposure Ingestion Service / REST API

E.4 Functionality & User Interface

This section will list the core functionalities available through the User Interfaces developed for each module.

E.4.1 Server

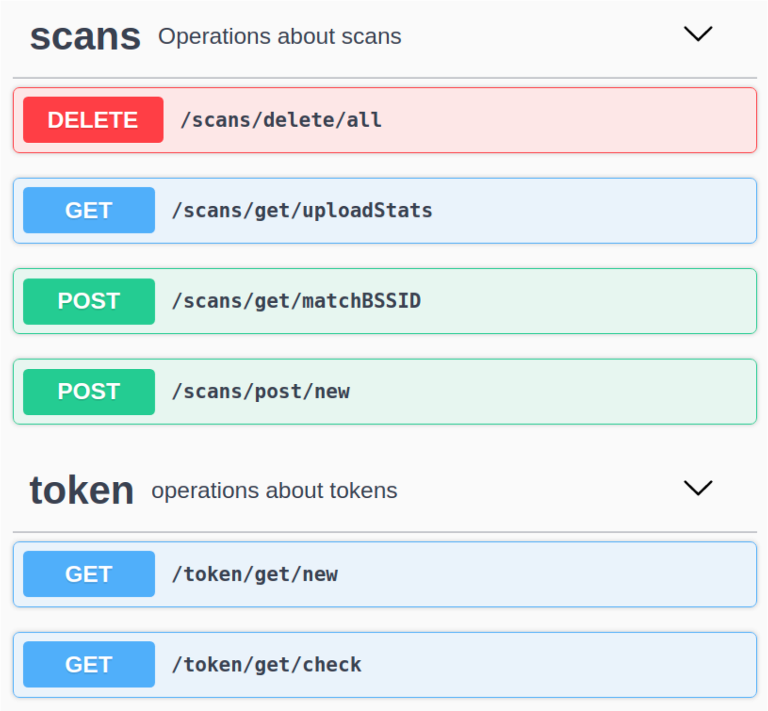
The Server does not contain any User Interface, as its main role is that of an API. However, Users that wish can access an interactive documentation HTML page generate through the Swagger library [8].

In order to access it, simply open a web browser at `http://<DOMAIN>:<PORT>/api-docs` [9].

You will be able to test the supported endpoints by clicking on the **Try it out** button. Make sure that the Server is running and the database is connected.

[8] More information available at `https://swagger.io/`

[9] For the default dockerised configuration, go to `http://localhost:4683/api-docs/`



scans Operations about scans

DELETE /scans/delete/all

GET /scans/get/uploadStats

POST /scans/get/matchBSSID

POST /scans/post/new

token operations about tokens

GET /token/get/new

GET /token/get/check

Figure E.3: A close look at some of the endpoints shown when viewing the REST API docs. Clicking on one of them will show more information.

E.4.2 Web Application

By default, the Web Application will be available at `http://localhost:3000`. All pages can be reached by using the navigation menu on the left side of the layout.

Map

This page contains an interactive map. After the loading is complete, it's possible to move around the map by holding and dragging the mouse cursor.

Layout Options

On the top-right corner of the map it's possible to modify the following setting:

- Choose the theme of the map. Light or Dark.
- Toggle the Heat-map on.
- Toggle the subdivision polygons that show contagion summaries.

If you want to visualise data about the new scans you've uploaded, make sure you've refreshed the cached data by either using the caching function, or by sending a PATCH request to the caching endpoints.

Division Details

You can hover over subdivisions to see a Popup showing details about the cached contagion data for that specific area. If the subdivisions are not visible, check the *Show Data* tick-box in the layer options.

E.4.3 Mobile Application

Main Activity

The Main Activity shows the current status of scanning. Opening the Main Activity will create a sticky notification that shows the current status of the scanning.

If the location or WiFi services are turned off, a different notification will be shown, prompting you to turn them on.

You can enter the **Settings Activity** by tapping on the cog icon on the top-right corner of the page.

Settings Activity

By tapping on the **Preferences** button, you can update the preferences of the application, including enabling location logging and modifying API and Risk Analysis settings.

Upload Scans

Upload Scans allows you to test the upload of Access Points data to the Exposure Ingestion Service. You can choose to also send location data by checking the Switch component on screen. To upload the currently stored scans and WiFi locations, simply press **Upload Scans**.

It also shows two metrics: *Scans*, the number of scans performed, and *AP Scans*, the number of individual scans of each Access Point.

Manage Data.

From this screen, you can manage Access Point data stored remotely on the Server's database, as well as the scan data stored on the device. Tapping **Delete Collection** will delete the *scans* collection in the back-end. Tapping **Delete Data** will delete the locally stored data.

Check Exposure

This page will allow you to test the Risk Analysis performed by the application. Clicking on **Check Exposure** will simulate the period exposure check performed by the application every 12 hours. To change the constants used for the Risk Analysis, please go to **Preferences > Developer Options**.