**API Documentation For The Backend REST APIs**

* **Weather API:**
* **Endpoint:**

GET /weather/

* **Description:**

Retrieves weather data for the specified airport ID.

* **Parameters:**
* airport\_id (required): The ID of the airport for which weather data is requested.
* **Response:**
* Status Code: 200 OK
* Content Type: application/json
* **Example Response Body:**

{ "GKA": { "latitude": -6.125, "longitude": 145.375, "generationtime\_ms": 0.09799003601074219, "utc\_offset\_seconds": 0, "timezone": "GMT", "timezone\_abbreviation": "GMT", "elevation": 1585, "hourly\_units": { "time": "iso8601", "temperature\_2m": "°C", "relative\_humidity\_2m": "%", "precipitation\_probability": "%", "cloud\_cover": "%", "visibility": "m", "wind\_speed\_10m": "km/h", "wind\_direction\_10m": "°", "wind\_gusts\_10m": "km/h" }, "hourly": { "time": ["2024-05-24T00:00", "2024-05-24T01:00", ...], "temperature\_2m": [21.5, 23.2, ...], "relative\_humidity\_2m": [76, 69, ...], "precipitation\_probability": [0, 24, ...], "cloud\_cover": [71, 72, ...], "visibility": [21880, 24140, ...], "wind\_speed\_10m": [2.7, 3.1, ...], "wind\_direction\_10m": [157, 159, ...], "wind\_gusts\_10m": [6.5, 9, ...] } } }

* **Example:**

GET /weather/GKA

This will return weather data for the airport with the ID "GKA".

* **Airport API:**
* **Endpoint:**

GET /airport

* **Description:**

Retrieves a list of airports.

* **Response:**
* Status Code: 200 OK
* Content Type: application/json
* **Example Response Body:**

{ "GKA": { "Name": "Goroka Airport", "Latitude": "-6.081689835", "Longitude": "145.3919983" }, ... }

* **Example:**

GET /airports

This will return a list of sample airports.

* **Flight Route API:**
* **Endpoint:**

GET /flight\_route

* **Description:**

Retrieves a list of flight routes with final weights.

* **Response:**
* Status Code: 200 OK
* Content Type: application/json
* **Example Response Body:**

[ { "source": "ABJ", "destination": "BOY", "source\_latitude": "5.261390209", "source\_longitude": "-3.926290035", "destination\_latitude": "11.16009998", "destination\_longitude": "-4.33096981", "haversineDistance": 657.4155138661545, "weatherWeight": 0.2, "routeWeight": 394.52930831969263 }, ... ]

* **Shortest Route API:**
* **Endpoint:**

GET /flight\_route/get\_shortest\_route/:source\_airport\_id/to/:destination\_airport\_id

* **Description:**

Retrieves the shortest route between two airports.

* **Parameters:**
* source\_airport\_id (required): The ID of the source airport.
* destination\_airport\_id (required): The ID of the destination airport.
* **Response:**
* Status Code: 200 OK
* Content Type: application/json
* **Example Response Body:**

{ "route": [ { "routeNumber": 1, "source": "ABJ", "destination": "BOY", "haversineDistance": 657.415513866155, "sourceLatitude": "5.261390209", "sourceLongitude": "-3.926290035", "destinationLatitude": "11.16009998", "destinationLongitude": "-4.33096981", "weatherWeight": 0.2, "routeWeight": 394.529308319693 } ], "totalHaversineDistance": 657.415513866155 }

* **Example:**

GET /flight\_route/get\_shortest\_route/ABJ/to/BOY

This will return a list of flight routes from airport ABJ to BOY with final total Haversine distance that is needed to travel during the flight.