Title: **UQ Lakes Station Bus Tracker Documentation**

Introduction:

This documentation provides a glimpse of the functionality and structure of the UQ Lakes Station Bus Tracker code. The code is responsible for fetching and processing bus tracking data, allowing users to search for bus routes, departure times, and live location data.

Table of Contents:

**1. Code Overview**

**2. Functions and Purpose**

**3. Usage and Flow**

**4. Data Processing and Manipulation**

**5. User Interaction and Input**

**6. Cache Management**

**7. Conclusion**

1. Code Overview:

The code is a Node.js application that fetches bus tracking data, processes it, and interacts with the user to provide information about bus routes, departure times, and live location data.

2. Functions and Purpose:

- `fetchLiveData(url)`: Fetches JSON data from a provided URL and returns the parsed JSON response.

- `readCache(filenameAppend)`: Reads JSON data from a cache file with the specified filename.

- `saveCache(filenameAppend, data)`: Saves JSON data to a cache file with the specified filename.

- `getlivedata()`: Fetches live bus tracking data and saves it as cache.

- `LiveArrivalTimeAndLocation()`: Processes cached vehicle position data to extract live arrival times and locations.

- `askToRunAgain()`: Asks the user if they want to run the main function again or exit the program.

- `join(array1, array2, joinOnField, fieldsToMerge)`: Joins two arrays based on a field and merges specific fields.

- `selectedDate()`, `selectedTime()`, `selectedRoute()`: Get user inputs for departure date, time, and bus route.

- `FilteredRoute(data\_pool, userRoute)`: Filters bus route data based on user-selected route.

- `combineLiveData(liveData, finalData)`: Combines live arrival time and location data with final data.

3. Usage and Flow:

- The main function `main()` serves as the entry point for the program.

- Static data is parsed and stored in variables for further processing.

- User inputs for date, time, and route are obtained.

- Data is joined and filtered based on date, time, and route.

- Live arrival time and location data is fetched and combined with the final data.

- The resulting data is displayed to the user.

- The user is prompted to run the main function again or exit.

4. Data Processing and Manipulation:

- Static data files (routes, trips, calendar, stop\_times, stops) are parsed and stored.

- Data is joined using common keys and filtered based on user inputs.

- Time and date manipulations are performed to filter and process data.

- Live data is fetched and combined with existing data.

5. User Interaction and Input:

- The program interacts with the user through command-line prompts.

- Users provide inputs for date, time, and bus route selection.

- Users are presented with options to run the main function again or exit.

6. Cache Management:

- Cache files are used to store fetched data for reuse.

- Cache is cleared at regular intervals using the `removeCache()` function.

7. Conclusion:

The UQ Lakes Station Bus Tracker code effectively fetches, processes, and displays bus tracking data to users. It utilizes various functions for data manipulation, user interaction, and cache management to provide a user-friendly experience for tracking bus routes and live information.

**Note:. Detailed explanations of individual functions and their parameters are available within the code comments.**