Smart Trashcan Backend (Java - Glassfish Server)

The Smart Trashcan backend is a Java-based application deployed on a Glassfish server, acting as the core service for managing and processing real-time trashcan data. This backend efficiently handles data reception, storage, and updates, enabling seamless communication between the smart trashcan hardware and the mobile application.

Key Features

1. **Real-Time Data Management**
    - Processes and stores real-time updates from the smart trashcan, such as:
        - **Trash Level**: Indicates the current fill percentage of the trashcan.
        - **Lid Status**: Tracks whether the trashcan lid is open or closed.
        - **Buzzer Status**: Indicates whether the buzzer is active.
2. **RESTful API Endpoints**
    - **POST** `/smart-trashcan`
        - Receives and logs updates from the hardware, ensuring data consistency and integrity.
        - Responds with a success or error message based on the request's validity.
    - **GET** `/smart-trashcan`
        - Serves the latest trashcan data to clients (mobile app or browser) using Server-Sent Events (SSE) for real-time updates.
3. **Real-Time Event Streaming**
    - Implements **Server-Sent Events (SSE)** to push real-time updates about the trashcan's status to connected clients.
    - Ensures immediate visibility of key metrics in the mobile app.
4. **Logging and Error Handling**
    - Logs all incoming data and events for monitoring and debugging.
    - Validates incoming requests and responds with appropriate HTTP status codes and JSON messages.
5. **Timestamped Data Storage**
    - Captures and stores the timestamp of each update to provide a complete history of trashcan activities.

Setup and Deployment

1. **Prerequisites**
    - Java Development Kit (JDK) installed.
    - Glassfish server configured and running.
2. **Deploy the Backend**
    - Compile the Java code and package it as a `.war` file.
    - Deploy the `.war` file on the Glassfish server through the administration console or CLI.
3. **API Endpoints**
    - Use tools like Postman or cURL to test the endpoints:
        - **POST**: Send JSON or form-data to `/smart-trashcan` to update trashcan data.
        - **GET**: Subscribe to `/smart-trashcan` to receive real-time updates via SSE.
4. **Integration**
    - Ensure the backend URL is configured in the React Native Expo mobile app for seamless integration.

Technologies Used

- **Java**: Core programming language for backend logic.
- **Glassfish Server**: Java EE application server for deployment.
- **Server-Sent Events (SSE)**: Enables real-time data streaming to connected clients.
- **GSON**: Converts Java objects to JSON format for client-server communication.

How It Works

1. **Hardware to Backend**
    - The smart trashcan hardware sends periodic updates (trash level, lid status, buzzer status) to the backend using HTTP POST requests.
2. **Backend Processing**
    - The backend validates the incoming data, logs it, and updates the current status.
    - Stores the latest data in memory along with a timestamp.

3. **Real-Time Updates**
    - The mobile app connects to the backend using the SSE-enabled GET endpoint to receive live updates.
    - The backend pushes the latest trashcan data to the app whenever available.

Conclusion

The Smart Trashcan backend bridges the gap between IoT hardware and the mobile interface. It leverages Java's robust ecosystem and Glassfish's reliable server capabilities to deliver a highly responsive and scalable waste management solution. This backend ensures real-time updates, smooth client-server communication, and an overall enhanced user experience.