Internet Cafe Spring Boot Application

Overview

This project is a backend system for managing an internet cafe. It allows registration of computers, management of user sessions, billing, and reporting. It features a web dashboard, REST API, and optional agent software for desktop session tracking.

Main Features

- Register computers and assign them to branches.
- Start/stop user sessions and track usage time.
- Generate daily billing reports.
- JWT-based authentication with roles: ADMIN, AGENT, USER.
- Web dashboard for live monitoring and management.
- Optional agent for desktop lock/unlock detection.

Main Modules/Packages

- **auth**: Authentication controllers and login logic.
- **authority**: User roles and authorities.
- **branch**: Branch entity and repository.
- **computer**: Computer entity, controller, and service for managing computers.
- **session**: Session entity, controller, and service for managing user sessions.
- **report**: Billing and reporting logic.
- **security**: JWT, security configuration, and filters.
- **service**: Core services (billing, agent, websocket, etc.).
- **users**: User management, controllers, and repositories.
- **utils**: Utility classes and helpers.
- **exception**: Custom exception handling.

Prerequisites & Dependencies

- Java 17+
- Maven (or use the included `mvnw` wrapper)
- Database: Microsoft SQL Server (default, configurable in `application.yaml`)
- Spring Boot 4.0.0 and related dependencies (see `pom.xml`)

Configuration

- Main config: `src/main/resources/application.yaml`
- Set database connection, server port (default: 8052), and JWT secret here.

Running the Application

1. **Build:**

```
./mvnw -DskipTests clean package
```

2. **Run:**

```
./mvnw spring-boot:run
# or
java -jar target/internet-cafe-0.0.1-SNAPSHOT.jar
```

3. **Test:**

```
./mvnw test
```

Accessing the Application

- **Dashboard UI:**
  http://localhost:8052/dashboard
  Live monitoring of computers, sessions, and revenue.
- **Login Page:**
  http://localhost:8052/login
  Default admin:
  - Username: `admin`
  - Password: `admin`

User Manual

1. Logging In

- Go to `/login`, enter your credentials.
- On first run, use the default admin credentials.

2. Using the Dashboard

- View all registered computers and their statuses.
- See running sessions and live revenue.
- Start/stop sessions, lock/unlock computers (if authorized).

3. API Usage

- Authenticate:
  `POST /api/auth/login` with JSON body `{ "username": "...", "password": "..." }`
- Use the returned JWT for subsequent API requests in the `Authorization` header.

4. Managing Computers & Sessions

- Register a computer:
  `POST /computers` (ADMIN/AGENT)
- Start a session:
  `POST /sessions/{computerId}/start`
- Stop a session:
  `POST /sessions/{computerId}/stop-running`

5. Billing Reports

- Get daily report:
  `GET /api/v1/billing/reports/daily?from=YYYY-MM-DD&to=YYYY-MM-DD` (ADMIN)

6. Agent Setup (Optional)

- **Windows:**
  Place the JAR in `C:\internet-cafe`, add a startup script to run minimized.
- **Linux:**
  Place the JAR in `/opt/internet-cafe/`, ensure `dbus-monitor` is installed, add to startup applications.

Troubleshooting

- Check `application.yaml` for correct database and JWT settings.
- Ensure Java 17+ is installed.
- For agent issues, verify required dependencies (`dbus-monitor` on Linux, `LogonUI.exe` on Windows).

This documentation provides a clear overview, module summary, and user manual for both developers and end users. If you need more detailed API documentation or developer setup instructions, let me know!