TradeWithUs Java Backend

This directory contains the Java Spring Boot implementation of the TradeWithUs B2B platform backend. It handles user authentication, profile management, and product catalog functionality using Spring Boot, Spring Security, and MongoDB.

Tech Stack

- **Framework:** Spring Boot 3
- **Language:** Java 17
- **Database:** MongoDB
- **Authentication:** JWT (JSON Web Tokens)
- **Build Tool:** Maven

Features

- **User Authentication:** JWT-based authentication with signup, login, and profile retrieval
- **Profile Management:** CRUD operations for trader profiles
- **Product Catalog:** CRUD operations for product listings
- **Data Validation:** Request validation using Bean Validation
- **Security:** Role-based access control with Spring Security
- **Database Seeding:** Automatic seeding of initial data

Project Structure

```
backendJava/
├── src/
│   ├── main/
│   │   ├── java/
│   │   │   └── com/
│   │   │       └── tradewithus/
│   │   │           └── backend/
│   │   │               ├── controller/     # REST controllers
│   │   │               ├── model/          # Domain models/entities
│   │   │               ├── repository/     # MongoDB repositories
│   │   │               ├── security/       # Security config and JWT handling
│   │   │               ├── seeder/         # Database seeders
│   │   │               ├── service/        # Business logic
│   │   │               └── TradeWithUsApplication.java  # Main application
│   │   └── resources/
│   │       └── application.properties      # Configuration properties
│   └── test/                               # Test classes
├── pom.xml                                 # Maven configuration
└── README.md                               # This file
```

Setup and Running

Prerequisites

- JDK 17 or later
- Maven (3.8+)
- MongoDB (running instance or Atlas connection)

Installation

1. **Clone the repository** (if not already done)
2. **Navigate to the Java backend directory:**

```
cd backendJava
```

3. **Install dependencies and build:**

```
mvn clean install
```

Running the Application

```
mvn spring-boot:run
```

The API will be available at `http://localhost:8000`.

API Endpoints

Authentication

- **POST /auth/signup** - Register a new user
- **POST /auth/login** - Log in and get JWT token
- **GET /auth/me** - Get current user info (requires token)
- **DELETE /auth/user/{userId}** - Delete a user (requires token)

Products

- **GET /product/all** - Get all products
- **GET /product/{productId}** - Get product by ID
- **POST /product** - Create product (requires token)
- **PUT /product/{productId}** - Update product (requires token)
- **DELETE /product/{productId}** - Delete product (requires token)

Profiles

- **GET /profile/all** - Get all profiles
- **GET /profile/{profileId}** - Get profile by ID

- **POST /profile** - Create profile (requires token)
- **PUT /profile/{profileId}** - Update profile (requires token)
- **DELETE /profile/{profileId}** - Delete profile (requires token)

Health

- **GET /health** - Server health check

Development

Configuration

Edit `src/main/resources/application.properties` to configure:

- MongoDB connection
- JWT secret and expiration
- Server port
- Logging levels

Database Seeding

The application automatically seeds the database with sample data on startup if no data exists. This is handled by the `DatabaseSeeder` class.

Testing

Run tests using Maven:

```
mvn test
```