

E-commerce App - Usage Documentation

Technologies Used

- Backend: Java, Spring Boot, Spring Data JPA, H2/SQL
- Frontend: React, TypeScript, Axios, React Router
- Database: H2 (in-memory) or MySQL/PostgreSQL
- Build Tools: Maven (Backend), Vite/CRA (Frontend)
- API Format: REST

How to Run the Application

Backend (Spring Boot):

1. Open in IntelliJ IDEA or another IDE.
2. Configure `application.properties`.
3. Run the main class: `com.example.demo.DemoApplication`
4. App starts at <http://localhost:8080>

Frontend (React + TypeScript):

1. Navigate to frontend folder (e.g. client/)
2. Run: `npm install`
3. Start: `npm run dev`
4. Available at: <http://localhost:3000>

Available API Endpoints

`/api/users`

- GET `/api/users`
- GET `/api/users/{id}`
- POST `/api/users`
- PUT `/api/users/{id}`
- DELETE `/api/users/{id}`

`/api/products`

- GET `/api/products`
- GET `/api/products/{id}`
- POST `/api/products`
- PUT `/api/products/{id}`
- DELETE `/api/products/{id}`

`/api/cart`

- GET `/api/cart?userId=1`
- POST `/api/cart/add?userId=1&productId=2&quantity=1`
- PUT `/api/cart/update/{cartItemId}?quantity=3`
- DELETE `/api/cart/remove/{cartItemId}`

`/api/purchases`

- POST `/api/purchases/buy?userId=1`
- GET `/api/purchases/user/{userId}`

E-commerce App - Usage Documentation

Frontend Usage

- User Management: Add, edit, delete users
- Product Management: Add, edit, delete products
- Cart: Add products to cart, update/remove
- Purchase: Checkout and view history

Available Routes:

- /users
- /products
- /cart
- /purchases

Testing Tips

- Use Postman or curl for API testing
- H2 resets on restart (unless using MySQL/PostgreSQL)

Final Notes

- Ensure both frontend (localhost:3000) and backend (localhost:8080) are running
- CORS is enabled for React requests
- Error handling implemented for key flows