

Setup Guide - Amazon-like E-commerce Application

PostgreSQL Installation

Your system doesn't have PostgreSQL installed. Here's how to install it:

macOS (using Homebrew)

```
# Install PostgreSQL
brew install postgresql@16

# Start PostgreSQL service
brew services start postgresql@16

# Add PostgreSQL to PATH (add to ~/.zshrc)
echo 'export PATH="/opt/homebrew/opt/postgresql@16/bin:$PATH"' >>
~/.zshrc
source ~/.zshrc

# Create the database
createdb ecommerce

# Verify installation
psql --version
```

macOS (using Postgres.app)

1. Download from: <https://postgresapp.com/>
2. Install and run Postgres.app
3. Click "Initialize" to create a new server
4. Open Terminal and add to PATH:

```
echo 'export
PATH="/Applications/Postgres.app/Contents/Versions/latest/bin:$PATH
"' >> ~/.zshrc
source ~/.zshrc
```

5. Create database:

```
createdb ecommerce
```

Alternative: Use Docker

If you prefer Docker, you can run PostgreSQL in a container:

```
# Start PostgreSQL in Docker
docker run --name postgres-ecommerce \
-e POSTGRES_PASSWORD=postgres \
-e POSTGRES_DB=ecommerce \
-p 5432:5432 \
-d postgres:16

# Verify it's running
docker ps
```

After PostgreSQL Installation

1. Verify PostgreSQL is running:

```
# Check if service is running
brew services list | grep postgresql
# OR for Docker
docker ps | grep postgres
```

2. Create the database (if not already created):

```
# Using psql command
createdb ecommerce

# OR connect to PostgreSQL and create manually
psql -U postgres
CREATE DATABASE ecommerce;
\q
```

3. Update credentials (if needed):

- Edit [src/main/resources/application.properties](#)
- Update username/password to match your PostgreSQL setup

Running the Application

Once PostgreSQL is set up:

```
# Build the project
./gradlew clean build

# Run the application
./gradlew bootRun
```

The application will:

- Automatically create database tables
- Populate with 12 sample products (with fake ASINs)
- Create 3 test users
- Start on <http://localhost:8081>

Quick Start Commands

```
# 1. Install PostgreSQL (macOS with Homebrew)
brew install postgresql@16
brew services start postgresql@16

# 2. Create database
createdb ecommerce

# 3. Run the application
./gradlew bootRun

# 4. Open in browser
open http://localhost:8081
```

Troubleshooting

"Connection refused" error

- PostgreSQL is not running
- Solution: [brew services start postgresql@16](#)

"database does not exist"

- Database not created
- Solution: [createdb ecommerce](#)

Port 5432 already in use

- Another PostgreSQL instance is running
- Solution: Stop other instances or change port in application.properties

Build fails

- Dependencies not downloaded
- Solution: [./gradlew clean build --refresh-dependencies](#)

What You'll See

Once running, the application provides:

1. Web Interface (<http://localhost:8081>)

- Product catalog with 12 items
- Categories: Electronics, Books, Home & Kitchen, Sports, Fashion, Beauty
- Each product has a fake ASIN (Amazon Standard Identification Number)
- Search functionality
- Add to cart buttons

2. REST API (<http://localhost:8081/api/products>)

- JSON responses for all products
- Filter by category, ASIN, or search keywords

3. Sample Data

- 3 test users (john.doe@, jane.smith@, bob.jones@)
- 12 products with realistic ASINs
- Stock quantities
- Product images (external URLs)

Next Steps

After the application is running, you can:

1. Browse products at <http://localhost:8081>

2. Test the REST API with curl or Postman:

```
# Get all products
curl http://localhost:8081/api/products

# Search products
curl http://localhost:8081/api/products/search?keyword=echo

# Get by category
curl http://localhost:8081/api/products/category/Electronics
```

3. Modify the data in `DataInitializer.java` to add your own products

4. Extend functionality with shopping cart, checkout, user authentication, etc.

Database Schema

The application creates these tables:

- `users` - User accounts
- `products` - Product catalog
- `orders` - Customer orders
- `order_items` - Order line items