Documentação: API de Autenticação em Java (Spring Boot)

# 1️⃣ Pré-requisitos

Antes de iniciar, você precisa ter instalado no seu computador:  
1. Java JDK 21 – Verifique com: java -version  
2. Maven – Verifique com: mvn -version  
3. Visual Studio Code com extensões: Java Extension Pack e Spring Boot Extension Pack  
4. Git (opcional, se for clonar projetos).

# 2️⃣ Abrir o projeto no VS Code

1. Extraia o arquivo ZIP do projeto: api-authentication-java.zip.  
2. Abra o VS Code.  
3. Vá em File → Open Folder e selecione a pasta api-authentication-java.  
4. Aguarde o VS Code reconhecer o projeto Maven e baixar dependências.

# 3️⃣ Estrutura do projeto

api-authentication-java/  
├── pom.xml  
└── src  
 └── main  
 ├── java/com/example/api  
 │ ├── ApiApplication.java  
 │ ├── controller  
 │ │ ├── AuthController.java  
 │ │ ├── LeadsController.java  
 │ │ ├── NewsController.java  
 │ │ ├── ProfileController.java  
 │ │ └── SalesController.java  
 │ ├── model/User.java  
 │ ├── service/AuthService.java  
 │ └── security/JwtUtil.java  
 └── resources/application.properties

# 4️⃣ Executar a API

No VS Code, abra o terminal integrado e execute:  
mvn spring-boot:run  
  
Se tudo estiver correto, verá algo como:  
Tomcat started on port(s): 8080  
Started ApiApplication in 2.345 seconds  
  
A API estará disponível em: http://localhost:8080

# 5️⃣ Endpoints da API

## 🔹 Autenticação

POST /api/login  
Body JSON:  
{  
 "username": "admin",  
 "password": "1234"  
}  
Retorna token JWT válido por 1 hora.

## 🔹 Leads

POST /api/leads/create – Criar lead  
DELETE /api/leads/delete – Deletar lead

## 🔹 Notícias

GET /api/news – Listar notícias

## 🔹 Perfil

POST /api/profile/create – Criar perfil  
DELETE /api/profile/delete – Deletar perfil  
PUT /api/profile/update – Atualizar perfil

## 🔹 Vendas

GET /api/sales/highlights – Destaques de vendas  
GET /api/sales/month – Vendas do mês  
GET /api/sales/stars – Vendedores estrela  
GET /api/sales/year – Vendas do ano

# 6️⃣ Testando a API

Você pode usar Postman ou Insomnia, ou cURL no terminal:  
  
Exemplo login:  
curl -X POST http://localhost:8080/api/login -H "Content-Type: application/json" -d '{"username":"admin","password":"1234"}'  
  
Exemplo acessar leads:  
curl -X POST http://localhost:8080/api/leads/create

# 7️⃣ Observações

Projeto é exemplo simples, pronto para testes.  
Todos os endpoints retornam mensagem dummy, mas a estrutura já permite expansão.  
JWT já implementado, mas sem filtro de autenticação nos endpoints (pode ser adicionado depois).