

Insurance Purchase API - README

Project Overview

The Insurance Purchase API is a Spring Boot REST API that allows users to:

- List available insurances
- Purchase an insurance policy
- Download the policy document

Tech Stack

- Backend: Java 17, Spring Boot 3
 - Database: MySQL
 - ORM: Hibernate
 - Build Tool: Maven
 - Testing: JUnit 5, Mockito

Project Structure

```
insurance-api/  
  src/main/java/com/insurance/api/  
    controller/      # REST API Controllers  
    service/         # Business Logic  
    repository/      # Database Repositories  
    entity/          # JPA Entities  
    dto/             # Data Transfer Objects  
    exception/       # Custom Exceptions  
  src/test/java/com/insurance/api/ # Unit Tests  
  src/main/resources/  
    application.properties # Configuration File  
    pom.xml                # Maven Dependencies  
    README.md              # Documentation
```

API Endpoints

Method	Endpoint	Description
GET	/api/insurances	Get all insurances
POST	/api/purchase	Purchase insurance
GET	/api/purchase/download/{id}	Download policy PDF

Setup Instructions

1. Clone the Repository:

```
```bash  
git clone https://github.com/your-repo/insurance-api.git
cd insurance-api
```
```

2. Configure the Database:

Insurance Purchase API - README

```
``properties
spring.datasource.url=jdbc:mysql://localhost:3306/insurance_db
spring.datasource.username=root
spring.datasource.password=root
``
```

3. Build and Run the Application:

```
``bash
mvn clean install
java -jar target/insurance-api-0.0.1-SNAPSHOT.jar
``
```

Deployment Guide

1. Set Up AWS EC2 Instance:

- Launch an Ubuntu EC2 instance on AWS.
- Open Security Group: Allow ports 22 (SSH), 8080 (API), 3306 (MySQL).
- Connect to EC2:

```
``bash
ssh -i your-key.pem ubuntu@your-ec2-public-ip
``
```

2. Install Dependencies:

```
``bash
sudo apt update
sudo apt install openjdk-17-jdk mysql-server nginx git -y
``
```

3. Deploy the Application:

```
``bash
git clone https://github.com/your-repo/insurance-api.git
cd insurance-api
mvn clean install
java -jar target/insurance-api-0.0.1-SNAPSHOT.jar
``
```

4. Configure Nginx (Optional):

```
``bash
sudo nano /etc/nginx/sites-available/default
``
```

Add:

```
``nginx
server {
    listen 80;
    server_name your-ec2-public-ip;
    location / {
        proxy_pass http://localhost:8080;
    }
}
```

Insurance Purchase API - README

...

Restart Nginx:

```
```bash
sudo systemctl restart nginx
```
```

Unit Test Cases

Example: InsuranceServiceTest.java

```
```java
import static org.junit.jupiter.api.Assertions.*;
import static org.mockito.Mockito.*;
import org.junit.jupiter.api.BeforeEach;
import org.junit.jupiter.api.Test;
import org.mockito.InjectMocks;
import org.mockito.Mock;
import org.mockito.MockitoAnnotations;
import java.util.Collections;
import java.util.List;

public class InsuranceServiceTest {

 @Mock
 private InsuranceRepository insuranceRepository;

 @InjectMocks
 private InsuranceService insuranceService;

 @BeforeEach
 void setUp() {
 MockitoAnnotations.openMocks(this);
 }

 @Test
 public void testGetAllInsurances() {
 Insurance insurance = new Insurance(1L, "Health Insurance", "Medical Coverage", 5000.0);
 when(insuranceRepository.findAll()).thenReturn(Collections.singletonList(insurance));
 List<Insurance> insurances = insuranceService.getAllInsurances();
 assertEquals(1, insurances.size());
 }
}
```
```

cURL Requests

1. Fetch All Insurances:

```
```bash
curl -X GET http://your-ec2-public-ip/api/insurances
```
```

Insurance Purchase API - README

2. Purchase Insurance:

```
```bash
curl -X POST http://your-ec2-public-ip/api/purchase \
 -H "Content-Type: application/json" \
 -d '{"insuranceId":1, "userId":101}'
```
```

3. Download Policy Document:

```
```bash
curl -X GET http://your-ec2-public-ip/api/purchase/download/1 -o policy.pdf
```
```

Conclusion

This guide covers the complete setup, deployment, API documentation, unit testing, and best coding practices for the Insurance Purchase API.