**Hướng dẫn chi tiết để tạo project insurance-claim từ đầu đến khi hoàn chỉnh**

Dưới đây là các bước chi tiết để tạo và cấu hình một project Spring Boot từ đầu, bao gồm việc tạo các file giống như trong project insurance-claim.zip đã tải lên.

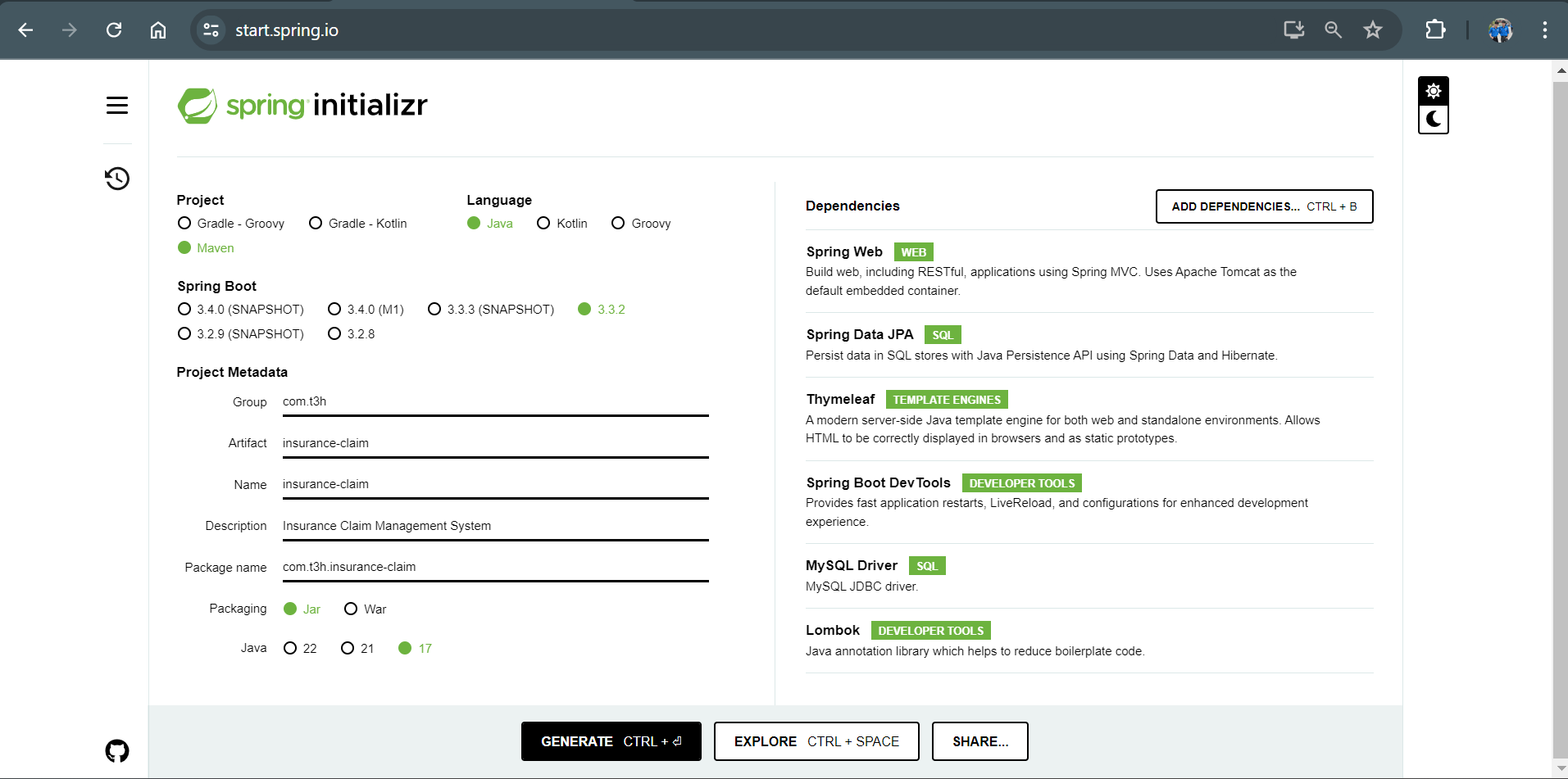
**Bước 1: Tạo project Spring Boot trên** [**https://start.spring.io/**](https://start.spring.io/)

1. **Truy cập trang web**: Mở trình duyệt và truy cập trang [start.spring.io](https://start.spring.io/).
2. **Cấu hình project**:
   * Project: Maven Project
   * Language: Java
   * Spring Boot: chọn phiên bản **3.3.2**
   * Project Metadata:
     + Group: com.t3h
     + Artifact: insurance-claim
     + Name: insurance-claim
     + Description: Insurance Claim Management System
     + Package name: com.t3h.insuranceclaim
     + Packaging: Jar
     + Java: 17
3. **Dependencies**: Chọn các dependencies cần thiết cho project, bao gồm:
   * Spring Web
   * Spring Data JPA
   * Thymeleaf
   * Spring Boot DevTools
   * MySQL Driver
   * Lombok
   * Spring Boot Starter Test

Các dependences dưới đây sẽ add sau khi tạo project

* + MapStruct
  + SpringFox
  + Servlet API

1. **Generate project**: Nhấn nút Generate để tải về file zip của project.

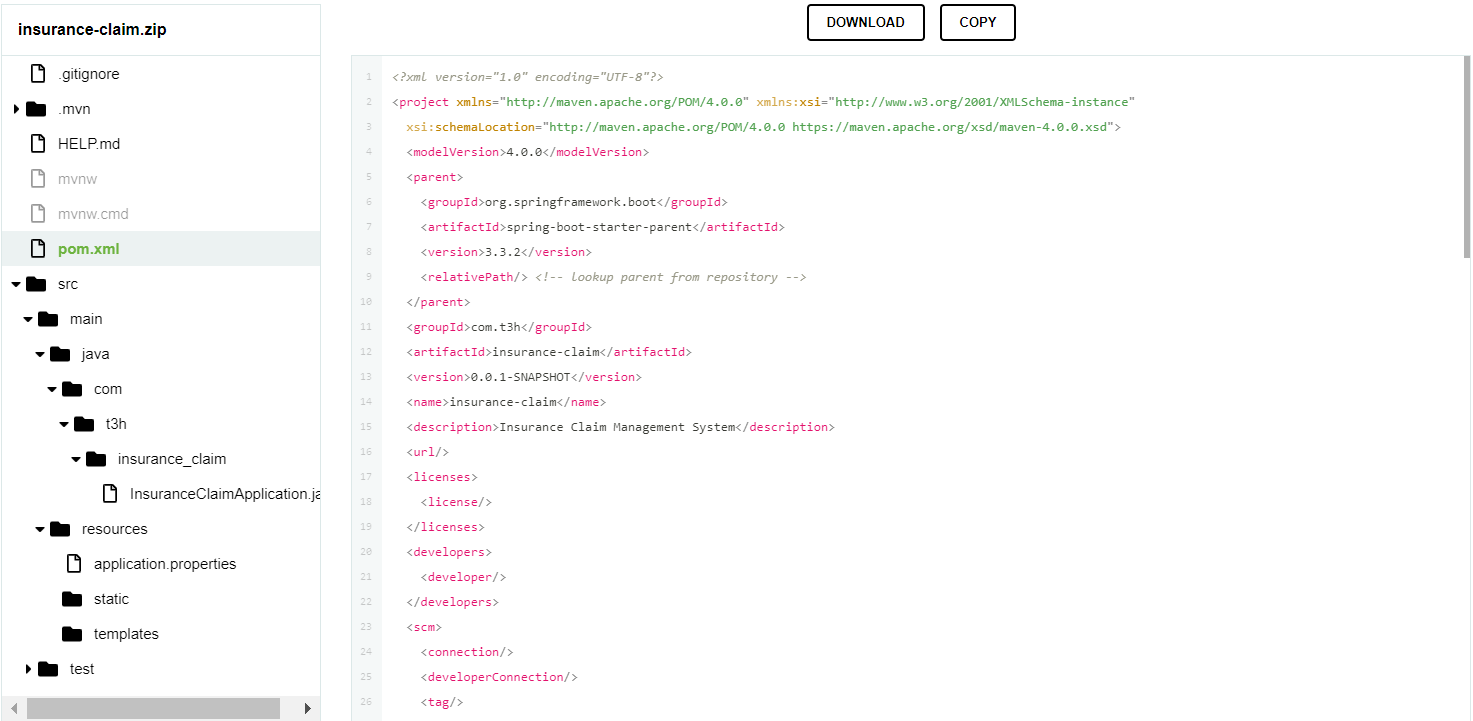
****

**Bước 2: Giải nén và mở project**

1. **Giải nén**: Giải nén file zip vừa tải về.
2. **Mở project**: Mở project bằng IDE Intellij

**Bước 3: Cấu trúc thư mục của project**

Khi mở project, bạn sẽ thấy cấu trúc thư mục như sau:



**Bước 4: Cấu hình file application.properties**

**Trước đó hãy tạo sẵn database insurance-claim trong database mysql**

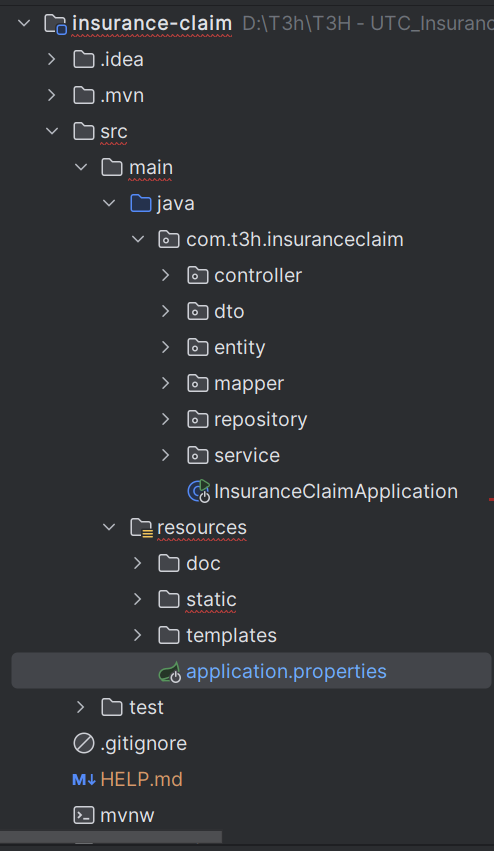
Mở file src/main/resources/application.properties và cấu hình cơ sở dữ liệu như sau:

spring.application.name=insurance-claim  
# MySQL configurations  
spring.datasource.url=jdbc:mysql://localhost:3306/insuranceclaim?allowPublicKeyRetrieval=true&useSSL=false  
spring.datasource.username=root  
#password sẽ tùy thuộc vào password bạn đặt cho mysql server, nêsu ai dùng mysql phpmyadmin có thể bỏ trống nếu ko có   
spring.datasource.password=root  
# JPA configurations  
spring.jpa.hibernate.ddl-auto=update  
spring.jpa.show-sql=true  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQL8Dialect  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.datasource.hikari.maximum-pool-size=10

**Bước 5: Tạo các package và classes cần thiết**

Tạo các package và class như sau:

1. **Entities**: Tạo package entity
2. **Repositories**: Tạo package repository
3. **Services**: Tạo package service
4. **Controllers**: Tạo package controller
5. **Mapper:** Tạo package mapper
6. **Dto:** Tạo package dto



**Tạo các file class**

**Entities: Tạo class Claim,** **ClaimDocument, ClaimStatus, Customer**

1. Tạo package com.t3h.insuranceclaim.entity.
2. Tạo file class Claim.java trong package entity.
3. Các file class khác tạo tương tự và copy từ source code mẫu ra  
   

Code ví dụ:  
  
package com.t3h.insuranceclaim.entity;  
  
import jakarta.persistence.\*;  
import lombok.Data;  
  
import java.time.LocalDate;  
import java.time.LocalDateTime;  
  
@Entity  
@Data  
public class Claim {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long id;  
  
 @ManyToOne  
 @JoinColumn(name = "customer\_id")  
 private Customer customer;  
  
 @ManyToOne  
 @JoinColumn(name = "product\_id")  
 private InsuranceProduct insuranceProduct;  
  
 @ManyToOne  
 @JoinColumn(name = "status\_id")  
 private ClaimStatus claimStatus;  
  
 private LocalDate claimDate;  
 private String description;  
 private double amount;  
 private LocalDateTime createdDate;  
 private String createdBy;  
 private LocalDateTime lastModifiedDate;  
 private String lastModifiedBy;  
 private Boolean deleted;  
 // Getters and Setters  
}

**Repositories: Tạo interface ClaimDocumentRepository.java, ClaimRepository.java, ClaimStatusRepository.java, CustomerRepository.java, InsuranceProductRepository.java**

1. Tạo package com.t3h.insuranceclaim.repository.
2. Tạo file ClaimDocumentRepository.java trong package repository.
3. Các file khác cũng tạo tương tự

Code ví dụ:

package com.t3h.insuranceclaim.repository;  
  
import com.t3h.insuranceclaim.entity.Claim;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface ClaimRepository extends JpaRepository<Claim, Long> {  
}

**Services: Tạo class ClaimService**

1. Tạo package com.t3h.insuranceclaim.service.
2. Tạo file InsuranceClaimService.java trong package service.

java

Sao chép mã

package com.example.insuranceclaim.service;

import com.example.insuranceclaim.entity.InsuranceClaim;

import com.example.insuranceclaim.repository.InsuranceClaimRepository;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Service;

import java.util.List;

@Service

public class InsuranceClaimService {

@Autowired

private InsuranceClaimRepository repository;

public List<InsuranceClaim> findAll() {

return repository.findAll();

}

public InsuranceClaim save(InsuranceClaim claim) {

return repository.save(claim);

}

public void deleteById(Long id) {

repository.deleteById(id);

}

}

**Controllers: Tạo class InsuranceClaimController**

1. Tạo package com.example.insuranceclaim.controller.
2. Tạo file InsuranceClaimController.java trong package controller.

java

Sao chép mã

package com.example.insuranceclaim.controller;

import com.example.insuranceclaim.entity.InsuranceClaim;

import com.example.insuranceclaim.service.InsuranceClaimService;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.web.bind.annotation.\*;

import java.util.List;

@RestController

@RequestMapping("/api/claims")

public class InsuranceClaimController {

@Autowired

private InsuranceClaimService service;

@GetMapping

public List<InsuranceClaim> getAllClaims() {

return service.findAll();

}

@PostMapping

public InsuranceClaim createClaim(@RequestBody InsuranceClaim claim) {

return service.save(claim);

}

@DeleteMapping("/{id}")

public void deleteClaim(@PathVariable Long id) {

service.deleteById(id);

}

}

**Bước 6: Tạo giao diện với Thymeleaf**

1. **Tạo file HTML trong thư mục templates**
   * Tạo file index.html trong src/main/resources/templates.
   * Tạo các thư mục con và file HTML khác trong src/main/resources/templates như admin, guest, common,...

**index.html**

html

Sao chép mã

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Insurance Claim</title>

</head>

<body>

<h1>Welcome to the Insurance Claim Management System</h1>

<a th:href="@{/admin/claims}">Manage Claims</a>

</body>

</html>

**admin/claim/claim-manager.html**

html

Sao chép mã

<!DOCTYPE html>

<html xmlns:th="http://www.thymeleaf.org">

<head>

<title>Claim Manager</title>

</head>

<body>

<h1>Claim Manager</h1>

<table>

<tr>

<th>ID</th>

<th>Claimant Name</th>

<th>Policy Number</th>

<th>Description</th>

<th>Amount</th>

</tr>

<tr th:each="claim : ${claims}">

<td th:text="${claim.id}"></td>

<td th:text="${claim.claimantName}"></td>

<td th:text="${claim.policyNumber}"></td>

<td th:text="${claim.claimDescription}"></td>

<td th:text="${claim.claimAmount}"></td>

</tr>

</table>

</body>

</html>

**Bước 7: Chạy ứng dụng**

1. **Run**: Chạy class InsuranceClaimApplication bằng cách nhấn chuột phải và chọn Run.
2. **Test API**: Sử dụng Postman hoặc công cụ khác để test các API của bạn.

**Bước 8: Kiểm tra và chỉnh sửa project trong file insurance-claim.zip**

1. **Giải nén**: Giải nén file insurance-claim.zip và kiểm tra mã nguồn hiện tại.
2. **Đối chiếu**: Đối chiếu mã nguồn hiện tại với các bước đã hướng dẫn để đảm bảo rằng tất cả các phần đã được cài đặt và cấu hình đúng cách.