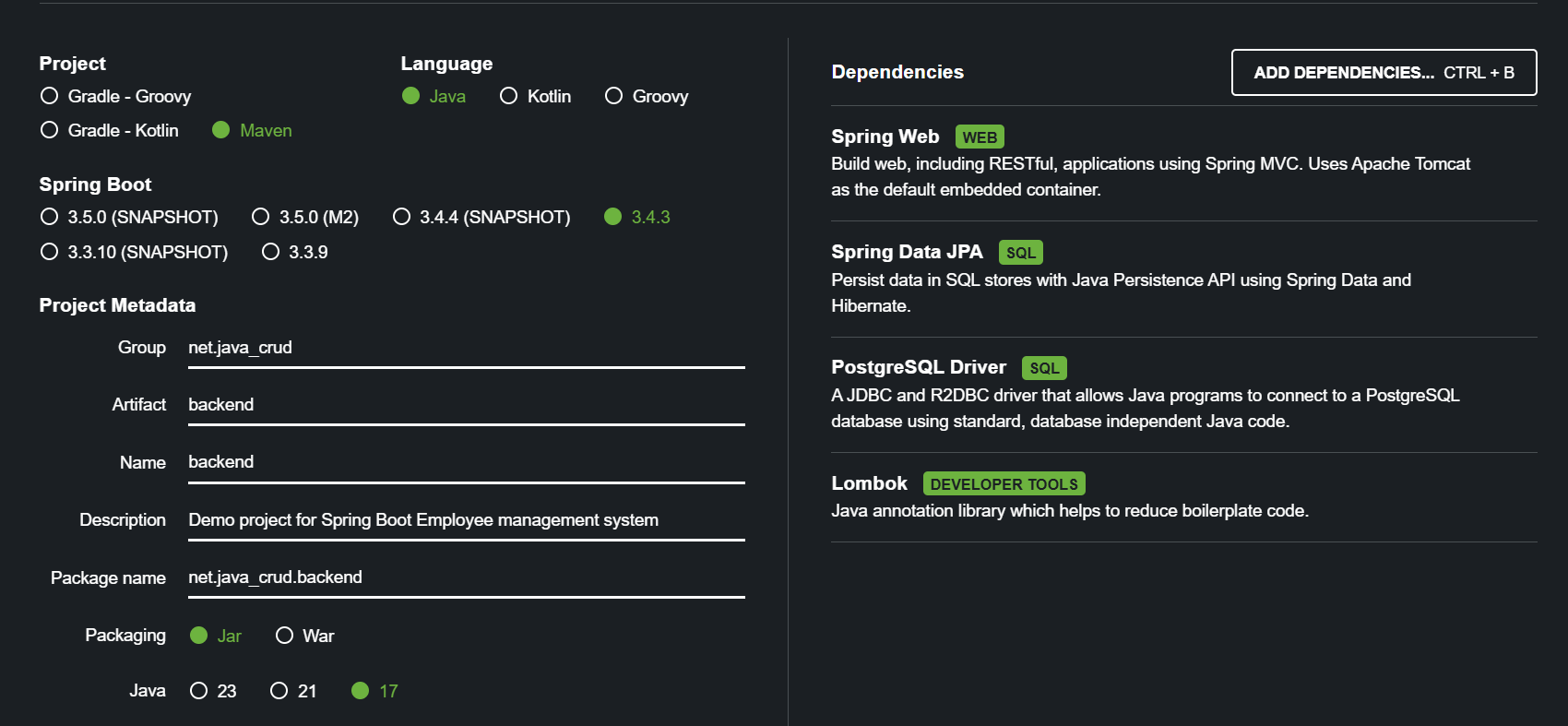
**Employee management system in java**

First you need to setup the springboot environment using <https://start.spring.io/>

And set 

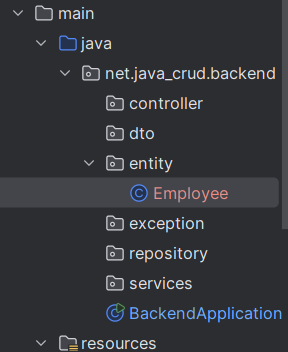
And add these dependencies as well. And generate that file and extract that,

**Connect database to the backend**

First create a database in postgresql and setup configuration in application properties file inside of the resources package.

spring.datasource.url=jdbc:postgresql://localhost:5432/ems  
spring.datasource.username=postgres  
spring.datasource.password=randila1212  
  
spring.jpa.hibernate.ddl-auto=update //automatically create database tables

and add these folder structure to the backend



package net.java\_crud.backend.entity;  
  
import jakarta.persistence.\*;  
  
import java.time.LocalDate;  
  
@Entity  
@Table(name = "employees")  
public class Employee {  
  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 private Long id;  
  
 @Column(name = "first\_name", nullable = false)  
 private String firstName;  
  
 @Column(name = "last\_name", nullable = false)  
 private String lastName;  
  
 @Column(name = "age", nullable = false)  
 private int age;  
  
 @Column(name = "email", nullable = false, unique = true)  
 private String email;  
  
 @Column(name = "join\_date")  
 private LocalDate joinDate;  
  
 // Constructors  
 public Employee() {}  
  
 public Employee(Long id, String firstName, String lastName, int age, String email, LocalDate joinDate) {  
 this.id = id;  
 this.firstName = firstName;  
 this.lastName = lastName;  
 this.age = age;  
 this.email = email;  
 this.joinDate = joinDate;  
 }  
  
 // Getters  
 public Long getId() {  
 return id;  
 }  
  
 public String getFirstName() {  
 return firstName;  
 }  
  
 public String getLastName() {  
 return lastName;  
 }  
  
 public int getAge() {  
 return age;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public LocalDate getJoinDate() {  
 return joinDate;  
 }  
  
 // Setters  
 public void setId(Long id) {  
 this.id = id;  
 }  
  
 public void setFirstName(String firstName) {  
 this.firstName = firstName;  
 }  
  
 public void setLastName(String lastName) {  
 this.lastName = lastName;  
 }  
  
 public void setAge(int age) {  
 this.age = age;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public void setJoinDate(LocalDate joinDate) {  
 this.joinDate = joinDate;  
 }  
}

after created a interface for the employee inside of the repository

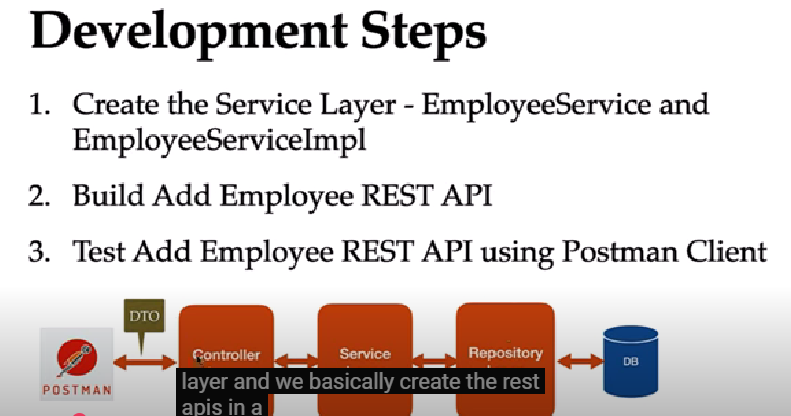
package net.java\_crud.backend.repository;  
  
import net.java\_crud.backend.entity.Employee;  
import org.springframework.data.jpa.repository.JpaRepository;  
  
public interface EmplyeeRepository extends JpaRepository<Employee,Long> {  
   
  
}

and then create Dto (data transfer object file)

package net.java\_crud.backend.dto;  
  
import java.time.LocalDate;  
  
public class EmployeeDto {  
 private long id;  
 private String firstName;  
 private String lastName;  
 private int age;  
 private String email;  
 private LocalDate joinDate;  
  
 // Constructors  
 public EmployeeDto() {}  
  
 public EmployeeDto(long id, String firstName, String lastName, int age, String email, LocalDate joinDate) {  
 this.id = id;  
 this.firstName = firstName;  
 this.lastName = lastName;  
 this.age = age;  
 this.email = email;  
 this.joinDate = joinDate;  
 }  
  
 // Getters  
 public long getId() {  
 return id;  
 }  
  
 public String getFirstName() {  
 return firstName;  
 }  
  
 public String getLastName() {  
 return lastName;  
 }  
  
 public int getAge() {  
 return age;  
 }  
  
 public String getEmail() {  
 return email;  
 }  
  
 public LocalDate getJoinDate() {  
 return joinDate;  
 }  
  
 // Setters  
 public void setId(long id) {  
 this.id = id;  
 }  
  
 public void setFirstName(String firstName) {  
 this.firstName = firstName;  
 }  
  
 public void setLastName(String lastName) {  
 this.lastName = lastName;  
 }  
  
 public void setAge(int age) {  
 this.age = age;  
 }  
  
 public void setEmail(String email) {  
 this.email = email;  
 }  
  
 public void setJoinDate(LocalDate joinDate) {  
 this.joinDate = joinDate;  
 }  
}

and then created mapper inside of the mapper

package net.java\_crud.backend.mapper;  
  
import net.java\_crud.backend.dto.EmployeeDto;  
import net.java\_crud.backend.entity.Employee;  
  
public class EmployeeMapper {  
  
 public static EmployeeDto mapToEmployeeDto(Employee emp) {  
 return new EmployeeDto(  
 emp.getId(),  
 emp.getFirstname(),  
 emp.getLastname(),  
 emp.getAge(),  
 emp.getEmail(),  
 emp.getJoinDate() // Ensure 'date' exists in Employee entity  
 );  
 }  
  
 public static Employee mapToEmployee(EmployeeDto empDto) {  
 return new Employee(  
 empDto.getId(),  
 empDto.getFirstname(),  
 empDto.getLastname(),  
 empDto.getAge(),  
 empDto.getEmail(),  
 empDto.getJoinDate() // Ensure 'date' exists in EmployeeDto  
 );  
 }  
}



package net.java\_crud.backend.services;  
  
import net.java\_crud.backend.dto.EmployeeDto;  
  
public interface EmplyeeService {  
 EmployeeDto createEmployee(EmployeeDto empdto);  
}

cteated employee services and its map to employeedto(system) to employee(database) and employee to employeeDto. And also implements Employeeservice interface as well;

package net.java\_crud.backend.services.impl;  
  
import lombok.AllArgsConstructor;  
import net.java\_crud.backend.dto.EmployeeDto;  
import net.java\_crud.backend.entity.Employee;  
import net.java\_crud.backend.repository.EmplyeeRepository;  
import net.java\_crud.backend.mapper.EmployeeMapper;  
import net.java\_crud.backend.services.EmplyeeService;  
import org.springframework.stereotype.Service;  
  
@Service  
@AllArgsConstructor  
public class EmployeeServiceImpl implements EmplyeeService {  
  
 private EmplyeeRepository employeeRepository;  
  
 @Override  
 public EmployeeDto createEmployee(EmployeeDto empdto){  
  
 Employee employee = EmployeeMapper.*mapToEmployee*(empdto);  
 Employee saveEmployee = employeeRepository.save(employee);  
 return EmployeeMapper.*mapToEmployeeDto*(saveEmployee);  
 }  
}