EXERCISE 1

//Controller Class

package com.cognizant.orm\_learn.Controller;  
import com.cognizant.orm\_learn.Service.CountryService;  
import com.cognizant.orm\_learn.Service.exception.CountrynotFoundException;  
import com.cognizant.orm\_learn.model.Country;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
import java.util.List;  
@RestController  
@RequestMapping("/countries")  
public class CountryController {  
 @Autowired  
 private CountryService countryService;  
 @GetMapping  
 public List<Country> getAllCountries() {  
 return countryService.getallcountries();  
 }  
 @GetMapping("/{code}")  
 public ResponseEntity<?> getCountryByCode(@PathVariable String code) {  
 try {  
 Country country = countryService.findCountryByCode(code);  
 return ResponseEntity.*ok*(country);  
 } catch (CountrynotFoundException e) {  
 return ResponseEntity.*status*(404).body(e.getMessage());  
 }  
 }  
 @PostMapping  
 public Country addCountry(@RequestBody Country country) {  
 System.*out*.println("Received from Postman: " + country);  
 return countryService.addCountry(country);  
 }  
 @GetMapping("/search")  
 public List<Country> searchCountries(@RequestParam String keyword) {  
 return countryService.searchCountriesByName(keyword);  
 }  
 @GetMapping("/search/sorted")  
 public List<Country> searchCountriesSorted(@RequestParam String keyword) {  
 return countryService.searchCountriesByNameSorted(keyword);  
 }  
 @GetMapping("/search/alphabet")  
 public List<Country> searchCountriesByAlphabet(@RequestParam String letter) {  
 return countryService.searchCountriesByStartingAlphabet(letter);  
 }  
}

//Model Class

package com.cognizant.orm\_learn.model;  
  
import com.fasterxml.jackson.annotation.JsonProperty;  
import jakarta.persistence.Column;  
import jakarta.persistence.Entity;  
import jakarta.persistence.Id;  
import jakarta.persistence.Table;  
  
@Entity  
@Table(name = "country")  
public class Country {  
  
 @Id  
 @Column(name = "co\_code")  
 @JsonProperty("code")  
 private String code;  
  
 @Column(name = "co\_name")  
 @JsonProperty("name")  
 private String name;  
  
 public String getCode() {  
 return code;  
 }  
  
 public void setCode(String code) {  
 this.code = code;  
 }  
  
 public String getName() {  
 return name;  
 }  
  
 public void setName(String name) {  
 this.name = name;  
 }  
  
 @Override  
 public String toString() {  
 return "Country(code=" + code + ", name=" + name + ")";  
 }  
}

//Service Class

package com.cognizant.orm\_learn.Service;  
import com.cognizant.orm\_learn.Service.exception.CountrynotFoundException;  
import com.cognizant.orm\_learn.model.Country;  
import com.cognizant.orm\_learn.repository.CountryRepository;  
import jakarta.transaction.Transactional;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import java.util.List;  
import java.util.Optional;  
@Service  
public class CountryService {  
 @Autowired  
 private CountryRepository countryRepository;  
  
 public CountryService(CountryRepository countryRepository) {  
 this.countryRepository = countryRepository;  
 }  
  
 public List<Country> getallcountries() {  
 return countryRepository.findAll();  
 }  
 @Transactional  
 public Country findCountryByCode(String countryCode) throws CountrynotFoundException {  
 Optional<Country> result = countryRepository.findById(countryCode);  
 if (!result.isPresent()) {  
 throw new CountrynotFoundException("Country with code " + countryCode + " not found.");  
 }  
 return result.get();  
 }  
 public Country addCountry(Country country) {  
 System.*out*.println("Saving: " + country);  
 return countryRepository.save(country);  
 }  
 public List<Country> searchCountriesByName(String keyword) {  
 return countryRepository.findByNameContainingIgnoreCase(keyword);  
 }  
 public List<Country> searchCountriesByNameSorted(String keyword) {  
 return countryRepository.findByNameContainingIgnoreCaseOrderByNameAsc(keyword);  
 }  
 public List<Country> searchCountriesByStartingAlphabet(String letter) {  
 return countryRepository.findByNameStartingWithIgnoreCase(letter);  
 }  
}

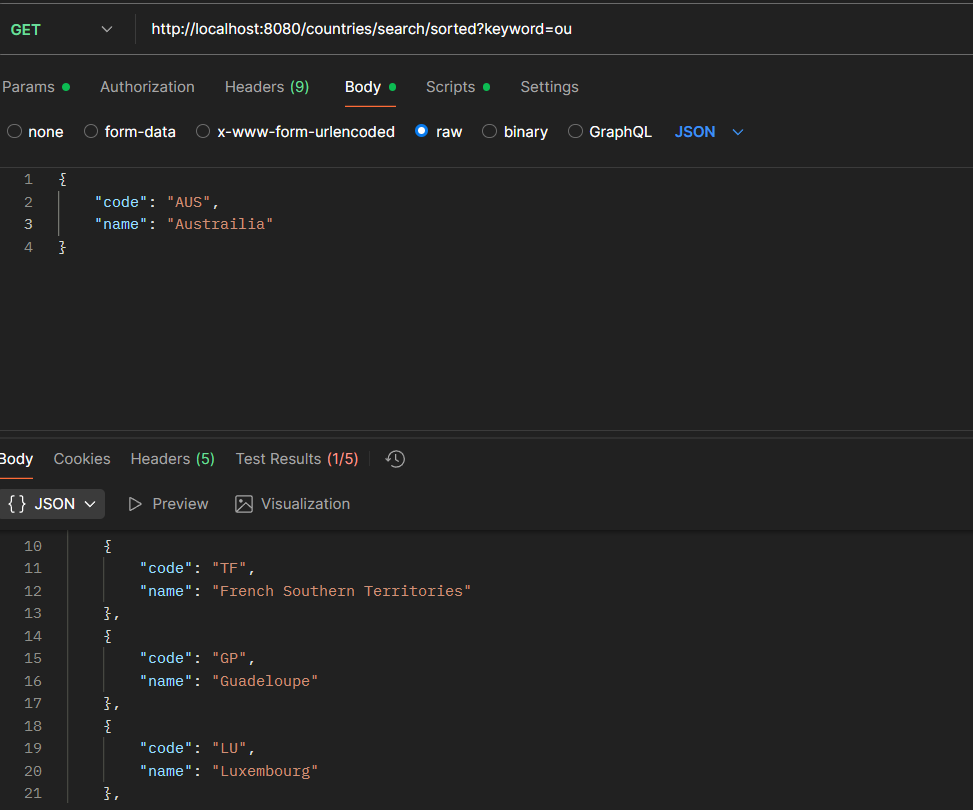
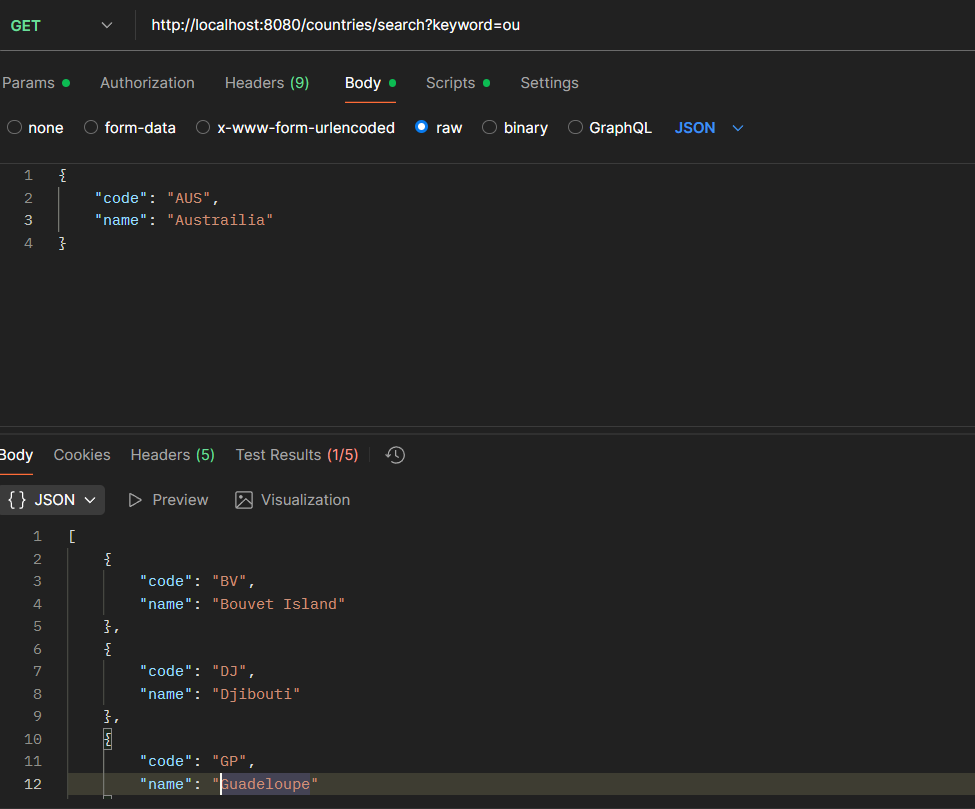
//Repository Class

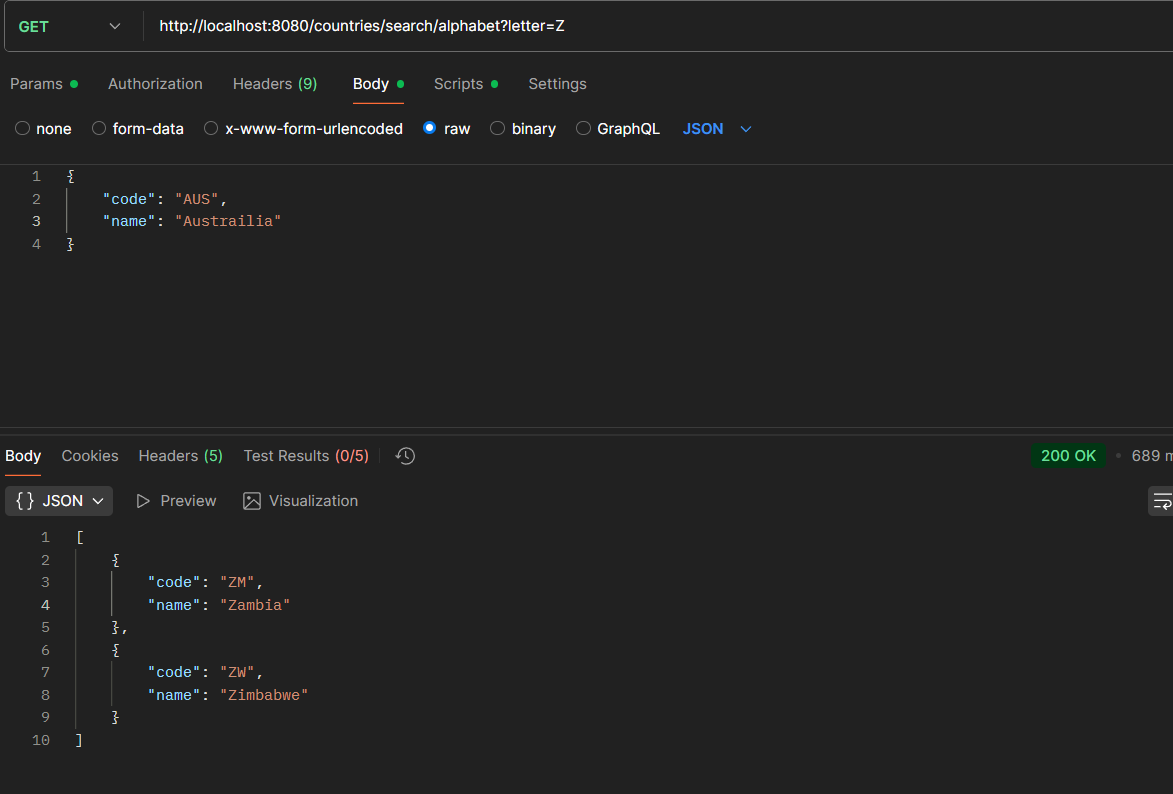
package com.cognizant.orm\_learn.repository;  
  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
import com.cognizant.orm\_learn.model.Country;  
  
import java.util.List;  
  
  
@Repository  
public interface CountryRepository extends JpaRepository<Country, String> {  
 List<Country> findByNameContainingIgnoreCase(String name);  
 List<Country> findByNameContainingIgnoreCaseOrderByNameAsc(String name);  
 List<Country> findByNameStartingWithIgnoreCase(String prefix);  
  
}

//Main Class

package com.cognizant.orm\_learn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class OrmLearnApplication {  
  
 public static void main(String[] args) {  
 SpringApplication.*run*(OrmLearnApplication.class, args);  
 System.*out*.println("Spring Boot Country Service App is running...");  
 }  
}

//output





///Exercise

//Contreoller Class

package com.cognizant.orm\_learn.Controller;  
  
import com.cognizant.orm\_learn.Service.DepartmentService;  
import com.cognizant.orm\_learn.model.Department;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
import java.util.List;  
  
@RestController  
@RequestMapping("/api/departments")  
public class DepartmentController {  
  
 @Autowired  
 private DepartmentService departmentService;  
  
 @GetMapping  
 public ResponseEntity<List<Department>> getAllDepartments() {  
 return ResponseEntity.*ok*(departmentService.getAllDepartments());  
 }  
 @GetMapping("/{id}")  
 public ResponseEntity<Department> getDepartmentById(@PathVariable int id) {  
 Department department = departmentService.get(id);  
 return department != null ? ResponseEntity.*ok*(department) : ResponseEntity.*notFound*().build();  
 }  
  
 @PostMapping  
 public ResponseEntity<Department> createDepartment(@RequestBody Department department) {  
 Department createdDepartment = departmentService.save(department);  
 return ResponseEntity.*status*(HttpStatus.*CREATED*).body(createdDepartment);  
 }  
  
 @PutMapping("/{id}")  
 public ResponseEntity<Department> updateDepartment(@PathVariable int id, @RequestBody Department department) {  
 Department updatedDepartment = departmentService.update(id, department);  
 return updatedDepartment != null ? ResponseEntity.*ok*(updatedDepartment) : ResponseEntity.*notFound*().build();  
 }  
  
 @DeleteMapping("/{id}")  
 public ResponseEntity<Void> deleteDepartment(@PathVariable int id) {  
 departmentService.delete(id);  
 return ResponseEntity.*noContent*().build();  
 }  
}

//

package com.cognizant.orm\_learn.Controller;  
  
import com.cognizant.orm\_learn.Service.EmployeeService;  
import com.cognizant.orm\_learn.model.Employee;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
import java.util.List;  
  
@RestController  
@RequestMapping("/api/employees")  
public class EmployeeController {  
  
 @Autowired  
 private EmployeeService employeeService;  
  
 @GetMapping  
 public ResponseEntity<List<Employee>> getAllEmployees() {  
 return ResponseEntity.*ok*(employeeService.getAllEmployees());  
 }  
  
 @GetMapping("/{id}")  
 public ResponseEntity<Employee> getEmployeeById(@PathVariable int id) {  
 Employee employee = employeeService.get(id);  
 return employee != null ? ResponseEntity.*ok*(employee) : ResponseEntity.*notFound*().build();  
 }  
  
 @PostMapping  
 public ResponseEntity<Employee> createEmployee(@RequestBody Employee employee) {  
 Employee createdEmployee = employeeService.save(employee);  
 return ResponseEntity.*status*(HttpStatus.*CREATED*).body(createdEmployee);  
 }  
  
 @PutMapping("/{id}")  
 public ResponseEntity<Employee> updateEmployee(@PathVariable int id, @RequestBody Employee employee) {  
 Employee updatedEmployee = employeeService.update(id, employee);  
 return updatedEmployee != null ? ResponseEntity.*ok*(updatedEmployee) : ResponseEntity.*notFound*().build();  
 }  
  
 @DeleteMapping("/{id}")  
 public ResponseEntity<Void> deleteEmployee(@PathVariable int id) {  
 employeeService.delete(id);  
 return ResponseEntity.*noContent*().build();  
 }  
}

//

package com.cognizant.orm\_learn.Controller;  
  
import com.cognizant.orm\_learn.Service.SkillService;  
import com.cognizant.orm\_learn.model.Skill;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.http.HttpStatus;  
import org.springframework.http.ResponseEntity;  
import org.springframework.web.bind.annotation.\*;  
import java.util.List;  
  
@RestController  
@RequestMapping("/api/skills")  
public class SkillController {  
  
 @Autowired  
 private SkillService skillService;  
  
 @GetMapping  
 public ResponseEntity<List<Skill>> getAllSkills() {  
 return ResponseEntity.*ok*(skillService.getAllSkills());  
 }  
  
 @GetMapping("/{id}")  
 public ResponseEntity<Skill> getSkillById(@PathVariable int id) {  
 Skill skill = skillService.get(id);  
 return skill != null ? ResponseEntity.*ok*(skill) : ResponseEntity.*notFound*().build();  
 }  
  
 @PostMapping  
 public ResponseEntity<Skill> createSkill(@RequestBody Skill skill) {  
 Skill createdSkill = skillService.save(skill);  
 return ResponseEntity.*status*(HttpStatus.*CREATED*).body(createdSkill);  
 }  
}

//Model Classes

package com.cognizant.orm\_learn.model;  
  
import jakarta.persistence.\*;  
import lombok.Data;  
import com.fasterxml.jackson.annotation.JsonIgnore;  
import java.util.HashSet;  
import java.util.Set;  
  
@Entity  
@Table(name = "department")  
@Data  
public class Department {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 @Column(name = "dp\_id")  
 private int id;  
  
 @Column(name = "dp\_name", nullable = false, length = 45)  
 private String name;  
  
 @OneToMany(mappedBy = "department", fetch = FetchType.*LAZY*, cascade = CascadeType.*ALL*)  
 @JsonIgnore  
 private Set<Employee> employees = new HashSet<>();  
}

//

package com.cognizant.orm\_learn.model;  
  
import jakarta.persistence.\*;  
import lombok.Data;  
import com.fasterxml.jackson.annotation.JsonIgnore;  
import java.math.BigDecimal;  
import java.util.Date;  
import java.util.HashSet;  
import java.util.Set;  
  
@Entity  
@Table(name = "employee")  
@Data  
public class Employee {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 @Column(name = "em\_id")  
 private int id;  
  
 @Column(name = "em\_name", nullable = false, length = 45)  
 private String name;  
  
 @Column(name = "em\_salary", nullable = false, precision = 10, scale = 2)  
 private BigDecimal salary;  
  
 @Column(name = "em\_permanent", nullable = false)  
 private boolean permanent = false;  
  
 @Column(name = "em\_date\_of\_birth", nullable = false)  
 @Temporal(TemporalType.*DATE*)  
 private Date dateOfBirth;  
  
 @ManyToOne(fetch = FetchType.*EAGER*)  
 @JoinColumn(name = "em\_dp\_id", nullable = false)  
 private Department department;  
  
 @ManyToMany(fetch = FetchType.*EAGER*)  
 @JoinTable(  
 name = "employee\_skill",  
 joinColumns = @JoinColumn(name = "es\_em\_id"),  
 inverseJoinColumns = @JoinColumn(name = "es\_sk\_id")  
 )  
 @JsonIgnore  
 private Set<Skill> skills = new HashSet<>();  
}

//

package com.cognizant.orm\_learn.model;  
  
import jakarta.persistence.\*;  
import lombok.Data;  
import com.fasterxml.jackson.annotation.JsonIgnore;  
import java.util.HashSet;  
import java.util.Set;  
  
@Entity  
@Table(name = "skill")  
@Data  
public class Skill {  
 @Id  
 @GeneratedValue(strategy = GenerationType.*IDENTITY*)  
 @Column(name = "sk\_id")  
 private int id;  
  
 @Column(name = "sk\_name", nullable = false, length = 45, unique = true)  
 private String name;  
  
 @ManyToMany(mappedBy = "skills", fetch = FetchType.*LAZY*)  
 @JsonIgnore  
 private Set<Employee> employees = new HashSet<>();  
}

//Repository Classes

package com.cognizant.orm\_learn.repository;  
  
import com.cognizant.orm\_learn.model.Department;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface DepartmentRepository extends JpaRepository<Department, Integer> {  
}

//

package com.cognizant.orm\_learn.repository;  
  
import com.cognizant.orm\_learn.model.Employee;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface EmployeeRepository extends JpaRepository<Employee, Integer> {  
}

// package com.cognizant.orm\_learn.repository;  
  
import com.cognizant.orm\_learn.model.Skill;  
import org.springframework.data.jpa.repository.JpaRepository;  
import org.springframework.stereotype.Repository;  
  
@Repository  
public interface SkillRepository extends JpaRepository<Skill, Integer> {  
}

//Service Classes

package com.cognizant.orm\_learn.Service;  
  
import com.cognizant.orm\_learn.model.Department;  
import com.cognizant.orm\_learn.repository.DepartmentRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import org.springframework.transaction.annotation.Transactional;  
import java.util.List;  
  
@Service  
public class DepartmentService {  
  
 @Autowired  
 private DepartmentRepository departmentRepository;  
  
 @Transactional(readOnly = true)  
 public Department get(int id) {  
 return departmentRepository.findById(id).orElse(null);  
 }  
  
 @Transactional  
 public Department save(Department department) {  
 return departmentRepository.save(department);  
 }  
  
 @Transactional(readOnly = true)  
 public List<Department> getAllDepartments() {  
 return departmentRepository.findAll();  
 }  
  
 @Transactional  
 public Department update(int id, Department departmentDetails) {  
 Department department = get(id);  
 if (department != null) {  
 department.setName(departmentDetails.getName());  
 return save(department);  
 }  
 return null;  
 }  
  
 @Transactional  
 public void delete(int id) {  
 departmentRepository.deleteById(id);  
 }  
}

////

package com.cognizant.orm\_learn.Service;  
  
import com.cognizant.orm\_learn.model.Employee;  
import com.cognizant.orm\_learn.repository.EmployeeRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import org.springframework.transaction.annotation.Transactional;  
import java.util.List;  
  
@Service  
public class EmployeeService {  
  
 @Autowired  
 private EmployeeRepository employeeRepository;  
  
 @Transactional(readOnly = true)  
 public Employee get(int id) {  
 return employeeRepository.findById(id).orElse(null);  
 }  
  
 @Transactional  
 public Employee save(Employee employee) {  
 return employeeRepository.save(employee);  
 }  
  
 @Transactional(readOnly = true)  
 public List<Employee> getAllEmployees() {  
 return employeeRepository.findAll();  
 }  
  
 @Transactional  
 public Employee update(int id, Employee employeeDetails) {  
 Employee employee = get(id);  
 if (employee != null) {  
 employee.setName(employeeDetails.getName());  
 employee.setSalary(employeeDetails.getSalary());  
 employee.setPermanent(employeeDetails.isPermanent());  
 employee.setDateOfBirth(employeeDetails.getDateOfBirth());  
 employee.setDepartment(employeeDetails.getDepartment());  
 return save(employee);  
 }  
 return null;  
 }  
  
 @Transactional  
 public void delete(int id) {  
 employeeRepository.deleteById(id);  
 }  
}

//////

package com.cognizant.orm\_learn.Service;  
  
import com.cognizant.orm\_learn.model.Skill;  
import com.cognizant.orm\_learn.repository.SkillRepository;  
import org.springframework.beans.factory.annotation.Autowired;  
import org.springframework.stereotype.Service;  
import org.springframework.transaction.annotation.Transactional;  
import java.util.List;  
  
@Service  
public class SkillService {  
  
 @Autowired  
 private SkillRepository skillRepository;  
  
 @Transactional(readOnly = true)  
 public Skill get(int id) {  
 return skillRepository.findById(id).orElse(null);  
 }  
  
 @Transactional  
 public Skill save(Skill skill) {  
 return skillRepository.save(skill);  
 }  
  
 @Transactional(readOnly = true)  
 public List<Skill> getAllSkills() {  
 return skillRepository.findAll();  
 }  
}

////Main Class

package com.cognizant.orm\_learn;  
  
import org.springframework.boot.SpringApplication;  
import org.springframework.boot.autoconfigure.SpringBootApplication;  
  
@SpringBootApplication  
public class OrmLearnApplication2 {  
 public static void main(String[] args) {  
 SpringApplication.*run*(OrmLearnApplication2.class, args);  
 }  
}

///output

