```java
//PetStoreApplication

package pet.store;

import org.springframework.boot.SpringApplication;
import org.springframework.boot.autoconfigure.SpringBootApplication;

@SpringBootApplication
public class PetStoreApplication {

    public static void main(String[] args) {
        SpringApplication.run(PetStoreApplication.class, args);

    }

}




//PetStoreController

package pet.store.controller;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.PutMapping;
import org.springframework.web.bind.annotation.RequestBody;
import org.springframework.web.bind.annotation.ResponseStatus;
import org.springframework.web.bind.annotation.RestController;

import lombok.extern.slf4j.Slf4j;
import pet.store.service.PetStoreService;

//telling springboot that this is a rest controller
@RestController
@Slf4j
public class PetStoreController {
    @Autowired
    private PetStoreService petStoreService;

    @PostMapping("/pet_store")
    @ResponseStatus(code = HttpStatus.CREATED)
    public PetStoreData insertPetStore(@RequestBody PetStoreData petStoreData) {
        log.info("Inserting pet store {}", petStoreData);

        return petStoreService.savePetStore(petStoreData);
    }

    @PutMapping("/pet_store/{petStoreId}")
    public PetStoreData updatePetStore(@PathVariable Long petStoreId,
        @RequestBody PetStoreData petStoreData) {
        petStoreData.setPetStoreId(petStoreId);
        log.info("Updating pet store {}", petStoreData);
        return petStoreService.savePetStore(petStoreData);
    }

}


//PetStoreData

package pet.store.controller;

import java.util.HashSet;
import java.util.Set;

import org.springframework.cglib.core.LocalVariablesSorter;

import lombok.Data;
import lombok.NoArgsConstructor;
import pet.store.entity.Customer;
import pet.store.entity.Employee;
import pet.store.entity.PetStore;
```

```
@Data
@NoArgsConstructor
public class PetStoreData {
    //creating the variables for this class
    private Long petStoreId;
    private String petStoreName;
    private String petStoreAddress;
    private String petStoreCity;
    private String petStoreState;
    private String petStoreZip;
    private String petStorePhone;
    private Set<PetStoreCustomer> customers = new HashSet<PetStoreCustomer>();
    private Set<PetStoreEmployee> employees = new HashSet<PetStoreEmployee>();

    public PetStoreData(PetStore petStore) {
        //filling the variables with data
        petStoreId = petStore.getPetStoreId();
        petStoreName = petStore.getPetStoreName();
        petStoreAddress = petStore.getPetStoreAddress();
        petStoreCity = petStore.getPetStoreCity();
        petStoreState = petStore.getPetStoreState();
        petStoreZip = petStore.getPetStoreZip();
        petStorePhone = petStore.getPetStorePhone();

        for (Customer customer : petStore.getCustomers()) {
            customers.add(new PetStoreCustomer(customer));

            for (Employee employee : petStore.getEmployees()) {
                employees.add(new PetStoreEmployee(employee));

            }
        }
    }

    @Data
    @NoArgsConstructor
    //Associating the petstorecustomers with petstore
    public static class PetStoreCustomer {
        private Long customerId;
        private String customerFirstName;
        private String customerLastName;
        private String customerEmail;

        private PetStoreCustomer(Customer customer) {
            customerId = customer.getCustomerId();
            customerFirstName = customer.getCustomerFirstName();
            customerLastName = customer.getCustomerLastName();
            customerEmail = customer.getCustomerEmail();
        }
    }

    @Data
    @NoArgsConstructor
    //Associating the petstoreemployees with petstore
    public static class PetStoreEmployee {
        private Long petStoreEmployeeId;
        private String petStoreEmployeeFirstName;
        private String petStoreEmployeeLastName;
        private String petStoreEmployeePhone;
        private String petStoreEmployeeJobTitle;

        private PetStoreEmployee(Employee employee) {
            petStoreEmployeeId = employee.getEmployeeId();
            petStoreEmployeeFirstName = employee.getEmployeeFirstName();
            petStoreEmployeeLastName = employee.getEmployeeLastName();
            petStoreEmployeePhone = employee.getEmployeePhone();
            petStoreEmployeeJobTitle = employee.getEmployeeJobTitle();
        }
    }
}



//GlobalErrorHandler

package pet.store.controller.error;
```

```java
import java.util.HashMap;
import java.util.Map;
import java.util.NoSuchElementException;

import org.springframework.http.HttpStatus;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.bind.annotation.ResponseStatus;
import org.springframework.web.bind.annotation.RestControllerAdvice;

import lombok.extern.slf4j.Slf4j;

@RestControllerAdvice
@Slf4j
public class GlobalErrorHandler {

    @ExceptionHandler(NoSuchElementException.class)
    @ResponseStatus(code = HttpStatus.NOT_FOUND)
    private Map<String, String> handleNoSuchErrorException(NoSuchElementException exception) {
        log.info("NoSuchElementException occured: {}", exception.getMessage());

        Map<String, String> errorResponse = new HashMap<String, String>();
        errorResponse.put("message", exception.toString());

        return errorResponse;
    }
}



//PetStoreDao

package pet.store.dao;

import org.springframework.data.jpa.repository.JpaRepository;
import pet.store.entity.PetStore;

public interface PetStoreDao extends JpaRepository<PetStore, Long> {

}


//PetStoreService

package pet.store.service;

import java.util.NoSuchElementException;
import java.util.Objects;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import pet.store.controller.PetStoreData;
import pet.store.dao.PetStoreDao;
import pet.store.entity.PetStore;

@Service
public class PetStoreService {
    // adding autowired so that spring can inject the DAO object
    @Autowired
    private PetStoreDao petStoreDao;

    @Transactional(readOnly = false)
    public PetStoreData savePetStore(PetStoreData petStoreData) {
        PetStore petStore = findOrCreatePetStore(petStoreData.getPetStoreId());

        copyPetStoreFields(petStore, petStoreData);

        return new PetStoreData(petStoreDao.save(petStore));
    }

    private void copyPetStoreFields(PetStore petStore, PetStoreData petStoreData) {
        petStore.setPetStoreId(petStoreData.getPetStoreId());
        petStore.setPetStoreName(petStoreData.getPetStoreName());
        petStore.setPetStoreAddress(petStoreData.getPetStoreAddress());
        petStore.setPetStoreCity(petStoreData.getPetStoreCity());
        petStore.setPetStoreState(petStoreData.getPetStoreState());
        petStore.setPetStoreZip(petStoreData.getPetStoreZip());
        petStore.setPetStorePhone(petStoreData.getPetStorePhone());

    }

    private PetStore findOrCreatePetStore(Long petStoreId) {
        PetStore petStore;
        if (Objects.isNull(petStoreId)) {
            petStore = new PetStore();
        } else {
            petStore = findPetStoreById(petStoreId);
        }
        return petStore;
    }

    private PetStore findPetStoreById(Long petStoreId) {
        return petStoreDao.findById(petStoreId)
                .orElseThrow(() -> new NoSuchElementException("Pet Store with ID= " + petStoreId + " was not found.
    }
}



//application.yaml

Spring:
  datasource:
    username: pet_store
    password: pet_store
    url: jdbc:mysql://localhost:3306/pet_store

  jpa:
    hibernate:
      #the "create" option will wipe out pre-existing data and recreate the tables
      #the "update" will keep existing data and update the tables with new information
      ddl-auto: update
    show-sql: true



///////////////////////////////////////////////////////////////////
//YOUTUBE: https://youtu.be/KNYCo8w1aiA
//GITHUB : https://github.com/fmd5045/Week13-15-PetStore-Project
//////////////////////////////////////////////////////////////////
```