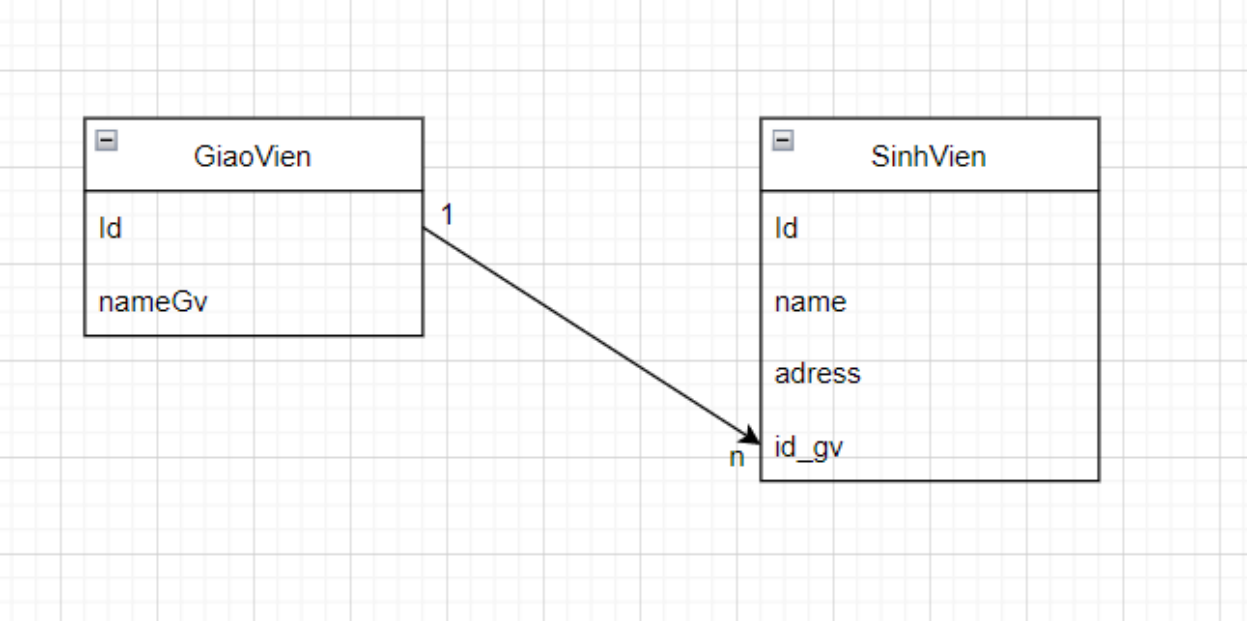
Đề bài: làm relationship 1-n bẳng giáo viên và sinh viên

Thiết kế database :s

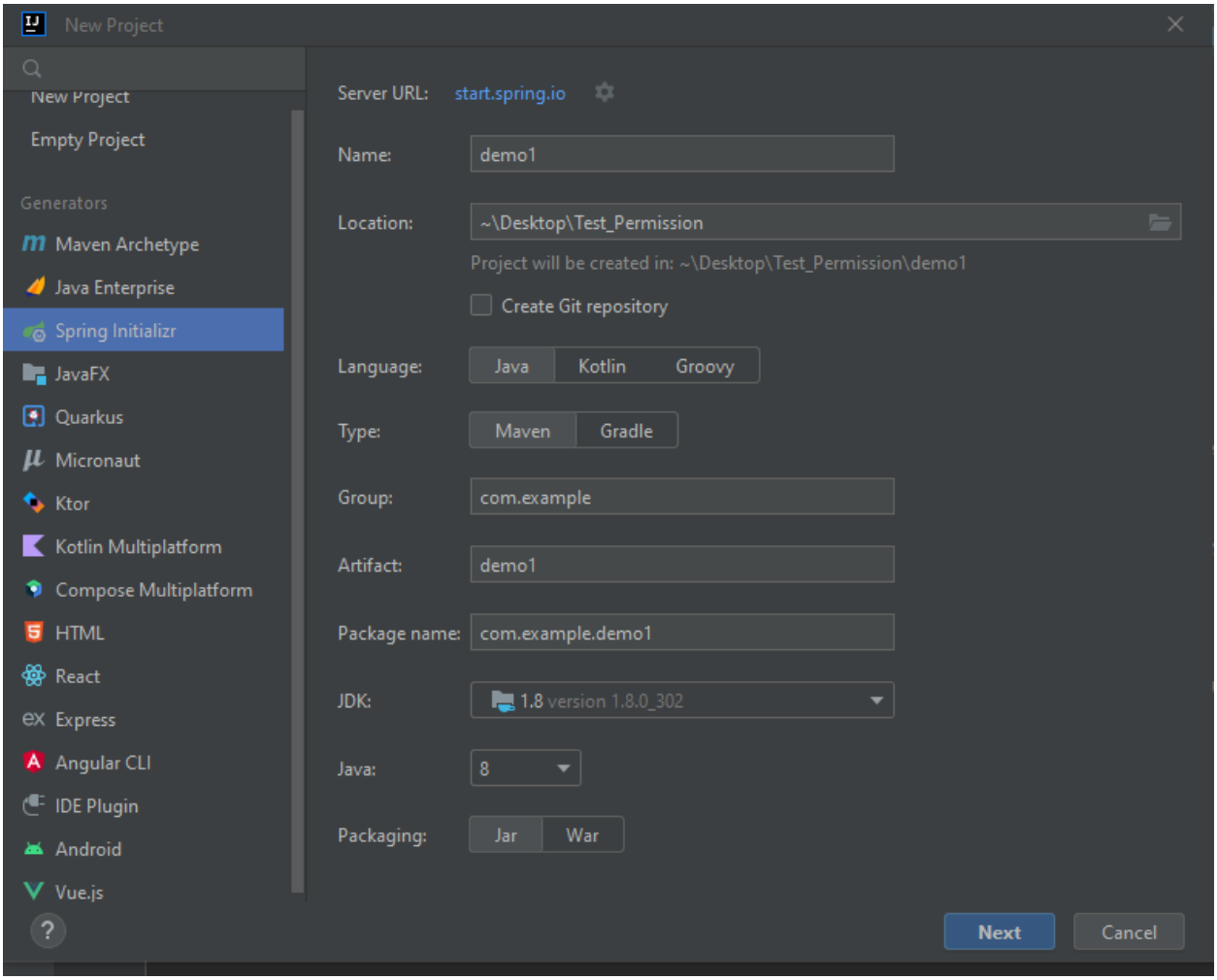
Giáo Viên (id,nameGv)

Sinh Viên(id,name,adresss,id_gv)

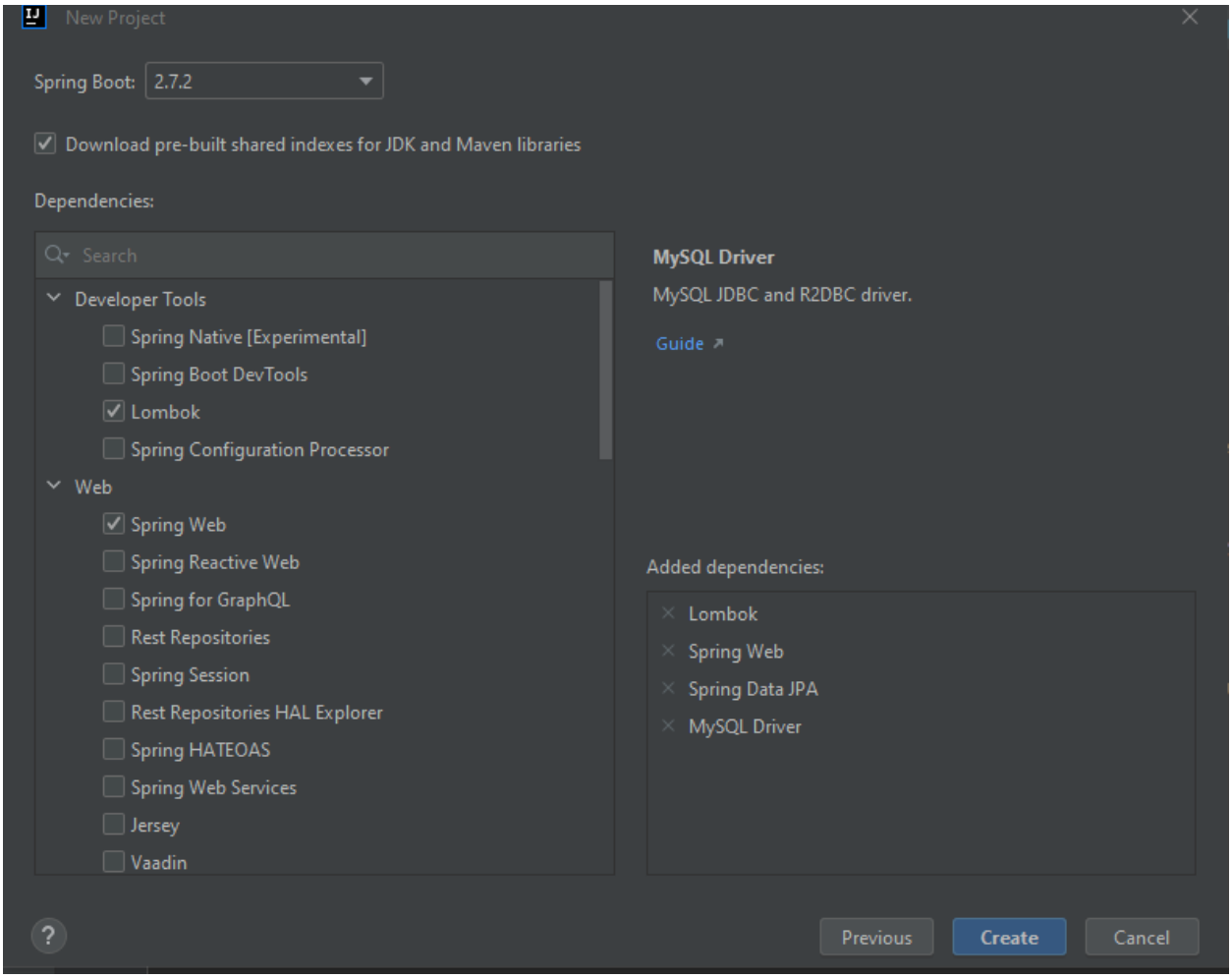


Tạo Project Maven trong intellij

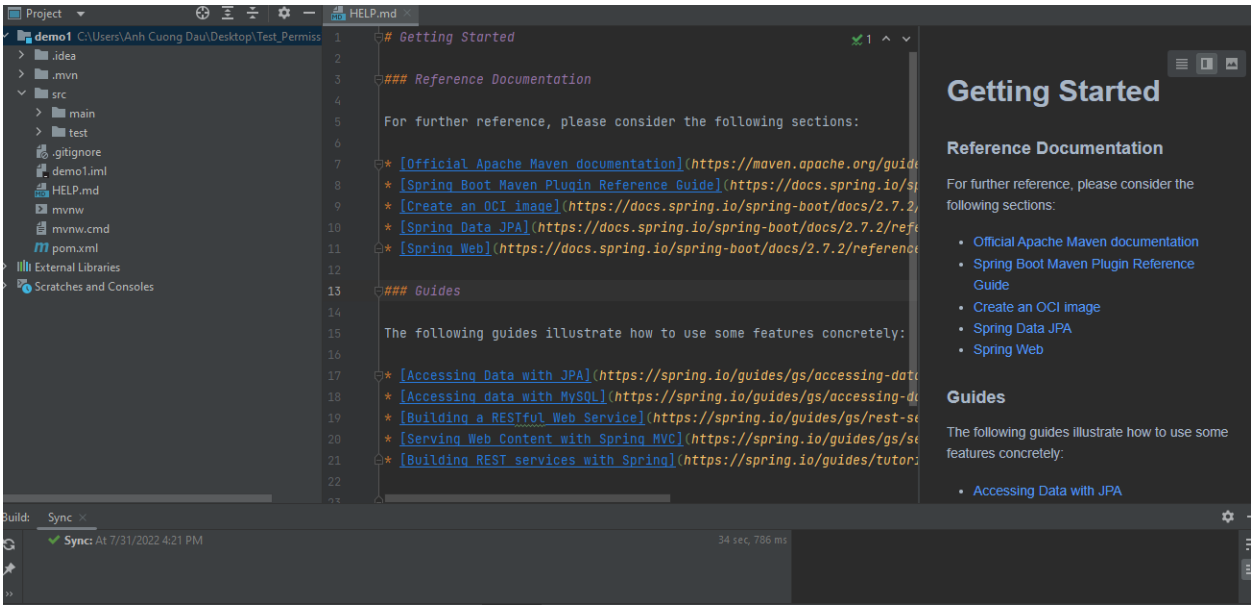
Bước 1: Mở intellij và chọn vào file -> new->project
Bước 2 : Cấu hình như ảnh



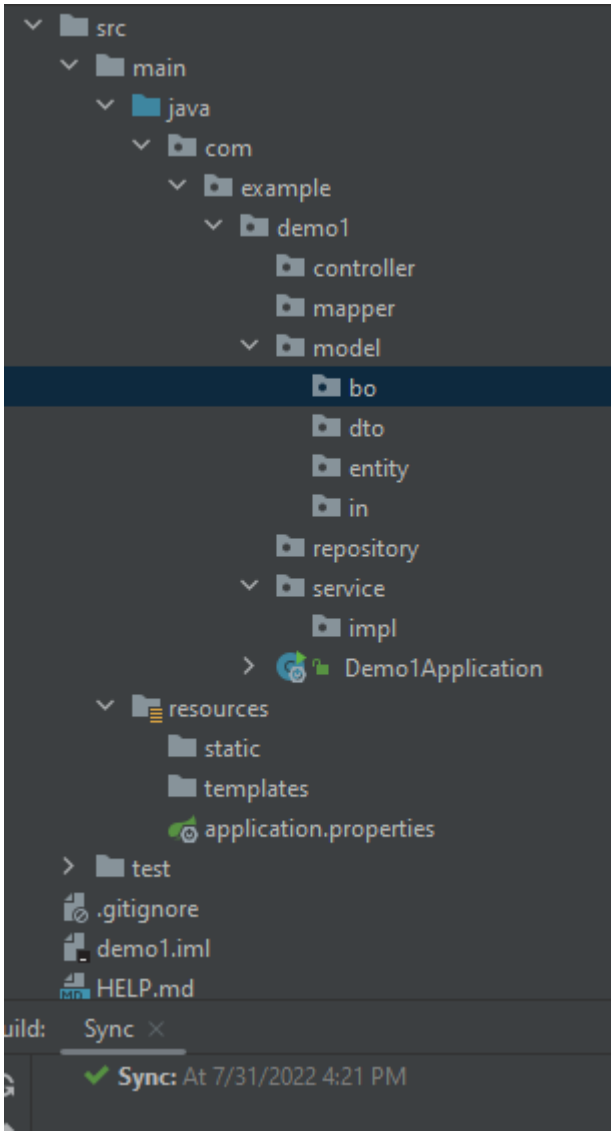
B3 : next và add 4 dependency : Lombok , spring web , spring data JPA , mysql drive



B4 : Nhấn Create là chúng ta đã có project java maven



Tạo xong project java maven rồi chúng ta tạo các file package controller , model (bo,entity,dto,in), mapper,repository,service(impl)



Tiếp theo chúng ta cấu hình file application.properties

```
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
spring.datasource.url=jdbc:mysql://localhost:3306/testRealtionF3?createDatabaseIfNotExist=true
spring.datasource.username=root
spring.datasource.password=123456
spring.jpa.hibernate.ddl-auto=update
spring.jpa.show-sql=true
```

Vào Package bo tạo class ResponPage để trả về cho Fontend messenger và dữ liệu 1 Object là data

```java
package com.example.demotts_java.model.bo;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

10 usages
@Data
@AllArgsConstructor
@NoArgsConstructor
public class ResponPage {
    private String messenger;
    private Object data;
}
```

Vẫn Package bo tạo class Respon để trả về cho fontend 1 trạng thái (true,false)và 1 messenegr

```java
package com.example.demotts_java.model.bo;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

22 usages
@Data
@AllArgsConstructor
@NoArgsConstructor
public class Respon {
    private boolean status;
    private String messenger;
}
```

Vào package entity tạo 1 class GiaoVien có id và nameGv

```java
package com.example.demotts_java.model.dto;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

6 usages
@Data
@AllArgsConstructor
@NoArgsConstructor
public class GiaoVienDto {
    private Integer Id;
    private String nameGv;
}
```

Vào package dto tạo 1 class GiaoViendto có id và nameGv

```java
package com.example.demotts_java.model.dto;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

6 usages
@Data
@AllArgsConstructor
@NoArgsConstructor
public class GiaoVienDto {
    private Integer Id;
    private String nameGv;
}
```

Vào package in(định nghĩa dữ liệu đầu vào) tạo 1 class GiaoVienIn có id và nameGv

```java
package com.example.demotts_java.model.in;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;
11 usages
@Data
@AllArgsConstructor
@NoArgsConstructor
public class GiaoVienIn {
        private Integer Id;
        private String nameGv;
}
```

Vào package mapper tạo 1 class GiaoVienMapper để chuyển dữ liệu đầu vào từ in sang entity để lưu vào database và lấy dữ liệu ra từ entity sang dto để trả Fontend

```java
import com.example.demotts_java.model.dto.GiaoVienDto;
import com.example.demotts_java.model.entity.GiaoVien;
import com.example.demotts_java.model.in.GiaoVienIn;

3 usages
public class GiaoVienMapper {
    1 usage
    public static GiaoVien mapIn(GiaoVienIn giaoVienIn){
        GiaoVien giaoVien = new GiaoVien();
        giaoVien.setId(giaoVienIn.getId());
        giaoVien.setNameGv(giaoVienIn.getNameGv());
        return giaoVien;
    }
    1 usage
    public static GiaoVienDto mapEntity(GiaoVien giaoVien)
        GiaoVienDto giaoVienDto = new GiaoVienDto();
        giaoVienDto.setId(giaoVien.getId());
        giaoVienDto.setNameGv(giaoVien.getNameGv());
        return giaoVienDto;
    }
}
```

Vào package repository tạo 1 class GiaoVienRepository (thao tác với bảng giao vien trong database)

```java
package com.example.demotts_java.repository;

import com.example.demotts_java.model.entity.GiaoVien;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.stereotype.Repository;

4 usages
@Repository
public interface GiaoVienRepository extends JpaRepository<GiaoVien,Integer> {
    6 usages
    GiaoVien getById(Integer Id);
}
```

Vào package service tạo 1 interface GiaoVienService để khởi tạo các phường thức trừu tượng

```java
package com.example.demotts_java.service;

import com.example.demotts_java.model.bo.Respon;
import com.example.demotts_java.model.bo.ResponPage;
import com.example.demotts_java.model.in.GiaoVienIn;
import org.springframework.stereotype.Service;

4 usages  1 implementation
@Service
public interface GiaoVienService {
    1 usage  1 implementation
    ResponPage getAllGV();
    1 usage  1 implementation
    Respon createGV(GiaoVienIn giaoVienIn);
    1 usage  1 implementation
    Respon updateGV(Integer Id,GiaoVienIn giaoVienIn);
    1 usage  1 implementation
    Respon deleteGv(Integer Id);
}
```

Vào package impl tạo 1 class GiaoVienServiceimpl implement GiaoVienService để ghi đè lên các phương thức mà inface đã định nghĩa từ trước

```java
package com.example.demotts_java.service.impl;

import com.example.demotts_java.mapper.GiaoVienMapper;
import com.example.demotts_java.model.bo.Respon;
import com.example.demotts_java.model.bo.ResponPage;
import com.example.demotts_java.model.dto.GiaoVienDto;
import com.example.demotts_java.model.entity.GiaoVien;
import com.example.demotts_java.model.entity.sinhVien;
import com.example.demotts_java.model.in.GiaoVienIn;
import com.example.demotts_java.repository.GiaoVienRepository;
import com.example.demotts_java.repository.sinhVienRepository;
import com.example.demotts_java.service.GiaoVienService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Component;

import java.util.List;
import java.util.stream.Collectors;

@Component
public class GiaoVienImpl implements GiaoVienService {

    6 usages
    @Autowired
    private GiaoVienRepository giaoVienRepository;

    1 usage
    @Override
```

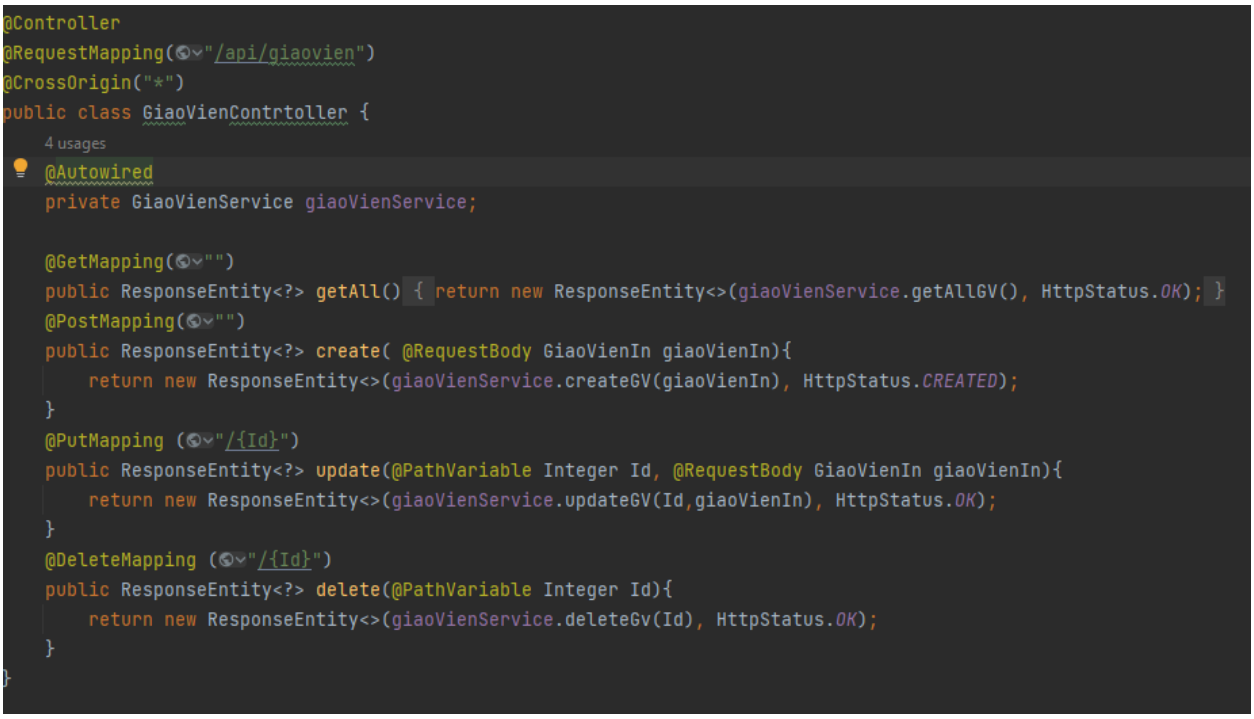
```java
@Override
public ResponPage getAllGV() {
    List<GiaoVien> list = giaoVienRepository.findAll();
    List<GiaoVienDto> listDto = list.stream().map(GiaoVienMapper::mapEntity).collect(Collectors.toList());
    return new ResponPage( messenger: "success", listDto);
}

1 usage
@Override
public Respon createGV(GiaoVienIn giaoVienIn) {
    GiaoVien giaoVien = GiaoVienMapper.mapIn(giaoVienIn);
    giaoVienRepository.save(giaoVien);
    return new Respon( status: true,  messenger: "add sucess");
}

1 usage
@Override
public Respon updateGV(Integer Id, GiaoVienIn giaoVienIn) {
    GiaoVien giaoVien = giaoVienRepository.getById(Id);
    giaoVien.setId(Id);
    giaoVien.setNameGv(giaoVienIn.getNameGv());
    giaoVienRepository.save(giaoVien);
    return new Respon( status: true,  messenger: "update success");
}
```

```java
1 usage
@Override
public Respon deleteGv(Integer Id) {
    GiaoVien giaoVien = giaoVienRepository.getById(Id);
    giaoVienRepository.delete(giaoVien);
    return new Respon( status: true, messenger: "delete success");
}
```

Vào package controller tạo 1 class GiaoVienController để lấy dữ liệu từ fontend truyền vào và return(trả ra dữ liệu cho fontend)



```java
@Controller
@RequestMapping("/api/giaovien")
@CrossOrigin("*")
public class GiaoVienContrtoller {
    4 usages
    @Autowired
    private GiaoVienService giaoVienService;

    @GetMapping("")
    public ResponseEntity<?> getAll() { return new ResponseEntity<>(giaoVienService.getAllGV(), HttpStatus.OK); }
    @PostMapping("")
    public ResponseEntity<?> create( @RequestBody GiaoVienIn giaoVienIn){
        return new ResponseEntity<>(giaoVienService.createGV(giaoVienIn), HttpStatus.CREATED);
    }
    @PutMapping ("/{Id}")
    public ResponseEntity<?> update(@PathVariable Integer Id, @RequestBody GiaoVienIn giaoVienIn){
        return new ResponseEntity<>(giaoVienService.updateGV(Id,giaoVienIn), HttpStatus.OK);
    }
    @DeleteMapping ("/{Id}")
    public ResponseEntity<?> delete(@PathVariable Integer Id){
        return new ResponseEntity<>(giaoVienService.deleteGv(Id), HttpStatus.OK);
    }
}
```

Xong CRUD Giáo viên

Tiếp theo chúng ta CRUD sinh viên

Vào entity khởi tạo bảng table sinh viên có id , tên , địa chỉ và khóa phụ

```java
@Entity
@Data
@AllArgsConstructor
@NoArgsConstructor
@Table(name = "sinhvien")
public class sinhVien {
    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private Integer Id;
    @Column
    private String name;
    @Column
    private String adress;

    1 usage
    @ManyToOne(cascade = CascadeType.MERGE)
    @JoinColumn(name = "id_gv")
    private GiaoVien giaoVien;

}
```

Vào in khởi tạo giá trị đầu vào của sinh viên

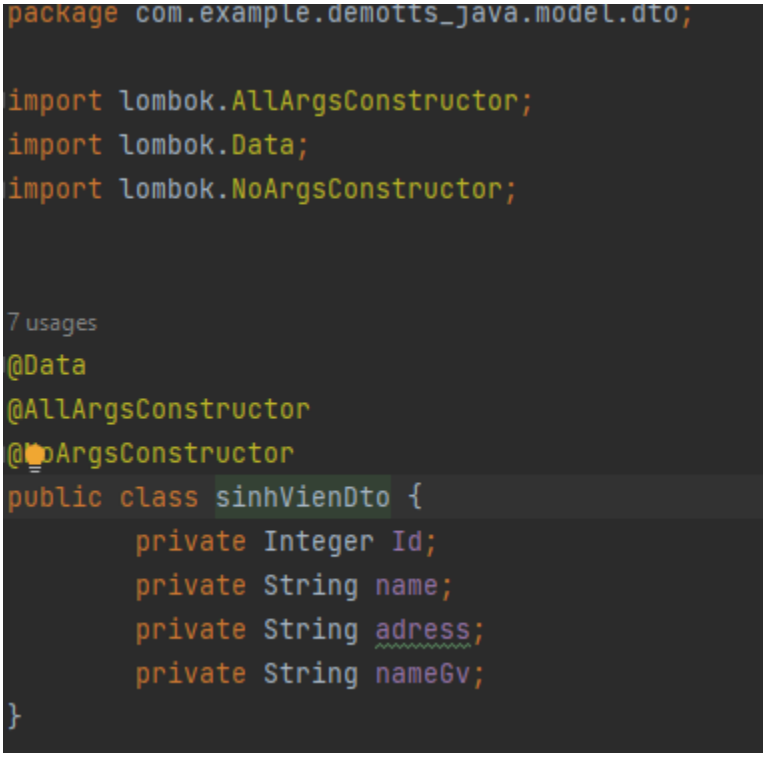
```java
package com.example.demotts_java.model.in;


import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;

import javax.persistence.Column;

12 usages
@Data
@AllArgsConstructor
@NoArgsConstructor
public class sinhVienIn {
    private Integer Id;
    private String name;
    private String adress;
    private Integer idGv;
}
```

Vào dto khởi tạo giá trị đầu ra của sinh viên khi get dữ liệu



```java
package com.example.demotts_java.model.dto;

import lombok.AllArgsConstructor;
import lombok.Data;
import lombok.NoArgsConstructor;


7 usages
@Data
@AllArgsConstructor
@NoArgsConstructor
public class sinhVienDto {
        private Integer Id;
        private String name;
        private String adress;
        private String nameGv;
}
```

Vào package mapper khởi tạo sinhvienMapper để chuyển dữ liệu đầu vào từ in sang entity để lưu vào database và lấy dữ liệu ra từ entity sang dto để trả Fontend

```
public class sinhVienMapper {
    1 usage
    public static sinhVien map(sinhVienIn sinhVienIn){
        sinhVien sinhVien = new sinhVien();
        sinhVien.setId(sinhVienIn.getId());
        sinhVien.setName(sinhVienIn.getName());
        sinhVien.setAdress(sinhVienIn.getAdress());
        return sinhVien;
    }
    1 usage
    public static sinhVienDto mapASD(sinhVien sinhVien){
        sinhVienDto sinhVienDto = new sinhVienDto();
        sinhVienDto.setId(sinhVien.getId());
        sinhVienDto.setName(sinhVien.getName());
        sinhVienDto.setAdress(sinhVien.getAdress());
        if(sinhVien.getGiaoVien()==null){
            sinhVienDto.setNameGv(null);
        }else {
            sinhVienDto.setNameGv(sinhVien.getGiaoVien().getNameGv());
        }
        return sinhVienDto;
    }
}
```

Và tiếp theo chúng ta vào package repository để tạo interface sinhVienRepository để chọc vào database lấy dữ liệu cần thiết từ yêu cầu của fontend

```java
import com.example.demotts_java.model.entity.sinhVien;
import org.springframework.data.jpa.repository.JpaRepository;
import org.springframework.data.jpa.repository.Query;
import org.springframework.stereotype.Repository;

3 usages
@Repository
public interface sinhVienRepository extends JpaRepository<sinhVien, Integer> {
    6 usages
    sinhVien getById(Integer Id);
}
```

Vào package service tạo 1 interface là sinhVienService để khởi tạo các phương thức trừu tượng

```java
import com.example.demotts_java.model.bo.Respon;
import com.example.demotts_java.model.bo.ResponPage;
import com.example.demotts_java.model.dto.sinhVienDto;
import com.example.demotts_java.model.entity.sinhVien;
import com.example.demotts_java.model.in.sinhVienIn;
import org.springframework.stereotype.Service;

import java.util.List;

5 usages  1 implementation
@Service
public interface sinhVienService {
    1 usage  1 implementation
    ResponPage getAll();
    1 usage  1 implementation
    Respon create(sinhVienIn sinhVienIn);
    1 usage  1 implementation
    Respon update(Integer Id ,sinhVienIn sinhVienIn);
    1 usage  1 implementation
    Respon delete(Integer Id);
}
```

Vào package impl tạo 1 class là sinhVienServiceimpl và kế thừa phương thức sinhVienService để khai triển phương thức mà bên interface định nghĩa

```java
@Component
public class sinhVienImpl implements sinhVienService  {

    6 usages
    @Autowired
    private sinhVienRepository sinhVienRepository;

    2 usages
    @Autowired
    private GiaoVienRepository giaoVienRepository;
    1 usage
    @Override
    public ResponPage getAll() {
        List<sinhVien> list = sinhVienRepository.findAll();
        List<sinhVienDto> listDto = list.stream().map(sinhVienMapper::mapASD).collect(Collectors.toList());
        return new ResponPage( messenger: "Success",listDto);
    }

    1 usage
    @Override
    public Respon create(sinhVienIn sinhVienIn) {
        sinhVien sinhVien = sinhVienMapper.map(sinhVienIn);
        GiaoVien giaoVien = giaoVienRepository.getById(sinhVienIn.getIdGv());
        sinhVien.setGiaoVien(giaoVien);
        sinhVienRepository.save(sinhVien);
        return new Respon( status: true, messenger: "add thành công Students");
    }
```

```
1 usage
@Override
public Respon update(Integer Id, sinhVienIn sinhVienIn) {
    sinhVien sinhVien = sinhVienRepository.getById(Id);
    GiaoVien giaoVien = giaoVienRepository.getById(sinhVienIn.getIdGv());
    sinhVien.setGiaoVien(giaoVien);
    sinhVien.setName(sinhVienIn.getName());
    sinhVien.setAdress(sinhVienIn.getAdress());
    sinhVienRepository.save(sinhVien);
    return new Respon( status: true, messenger: "update thành công Students");
}

1 usage
@Override
public Respon delete(Integer Id) {
    sinhVien sinhVien = sinhVienRepository.getById(Id);
    sinhVienRepository.delete(sinhVien);
    return new Respon( status: true, messenger: "delete thành công Students");
}
```

Cuối cùng ta vào file Package controller khởi tạo 1 class siinhVienController để lấy dữ liệu từ fontend truyền vào và return(trả ra dữ liệu cho fontend)

```java
@Controller
@RequestMapping("/api")
@CrossOrigin("*")
public class sinhVienController {
    4 usages
    @Autowired
    private sinhVienService sinhVienService;

    @GetMapping("")
    public ResponseEntity<?> getAll() { return new ResponseEntity<>(sinhVienService.getAll(), HttpStatus.OK); }
    @PostMapping("")
    public ResponseEntity<?> create(@RequestBody sinhVienIn sinhVienIn){
        return new ResponseEntity<>(sinhVienService.create(sinhVienIn), HttpStatus.CREATED);
    }
    @PutMapping ("/{Id}")
    public ResponseEntity<?> update(@PathVariable Integer Id, @RequestBody sinhVienIn sinhVienIn){
        return new ResponseEntity<>(sinhVienService.update(Id,sinhVienIn), HttpStatus.OK);
    }
    @DeleteMapping ("/{Id}")
    public ResponseEntity<?> delete(@PathVariable Integer Id){
        return new ResponseEntity<>(sinhVienService.delete(Id), HttpStatus.OK);
    }
}
```