```java
package com.example.demo.Controller;

import com.example.demo.Modele.Api;
import com.example.demo.Modele.Merchant;
import com.example.demo.Service.MerchantService;
import lombok.RequiredArgsConstructor;
import org.jetbrains.annotations.NotNull;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.validation.Errors;
import org.springframework.web.bind.annotation.*;

import javax.validation.Valid;
import java.util.ArrayList;

@RestController
@RequestMapping("Api/v1/Merchant")
@RequiredArgsConstructor
public class MerchantController {

    private final ArrayList<Merchant> merchant = new ArrayList<>();
    @Autowired
    private final MerchantService merchantService;




    @GetMapping(".get")
    public ResponseEntity<ArrayList<Merchant>> getMerchant(){
```

```java
        return ResponseEntity.status(200).body(merchant);

    }



    @PostMapping(".add")
    public  ResponseEntity addMerchant(@RequestBody @Valid Merchant merchant, @NotNull Errors errors){
        if (errors.hasErrors()){
            return ResponseEntity.status(400).body(new
Api(errors.getFieldError().getDefaultMessage(),400));

        }


        merchantService.addMerchant(merchant);
        return ResponseEntity.status(200).body(new Api("New Merchant added",200));


    }
    @PutMapping(".update")
    public ResponseEntity updateMerchant(@RequestBody @Valid Merchant merchant, Errors errors) {
        if (errors.hasErrors()) {
            return ResponseEntity.status(400).body(new Api(errors.getFieldError().getDefaultMessage(),
400));
        }
        Boolean isMerchantAdd=merchantService.updateMerchant(merchant);
        if (! isMerchantAdd){
            return ResponseEntity.status(200).body(new Api("New Merchant update", 200));
        }
        return ResponseEntity.status(400).body(new Api("Merchant not found",400));


    }
    @DeleteMapping(".delet")
```

```java
    public ResponseEntity deletMerchant(@PathVariable String id){

        if(merchantService.deletMerchant(id)){

            return ResponseEntity.status(200).body(new Api("Merchant deleted",200));

        }

        return ResponseEntity.status(400).body(new Api("Merchant not found",400));


    }
}
```