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

import com.example.demo.Modele.Api;
import com.example.demo.Modele.MerchantStock;
import com.example.demo.Service.MerchantStockService;
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/MerchantStock")
@RequiredArgsConstructor
public class MerchantStockController {

    @Autowired
    private final MerchantStockService merchantStockService;

    private final ArrayList<MerchantStock> merchantStock = new ArrayList<>();


    @GetMapping(".get")
    public ResponseEntity<ArrayList<MerchantStock>> getMerchantStock(){
        return ResponseEntity.status(200).body(merchantStock);
    }



    @PostMapping(".add")
    public  ResponseEntity addMerchantStock(@RequestBody @Valid MerchantStock merchantStock,
@NotNull Errors errors){
        if (errors.hasErrors()){
            return ResponseEntity.status(400).body(new
Api(errors.getFieldError().getDefaultMessage(),400));
        }

        merchantStockService.addMerchantStock(merchantStock);
        return ResponseEntity.status(200).body(new Api("New merchantStock added",200));

    }

    @PutMapping(".update")
    public ResponseEntity updateMerchantStock(@RequestBody @Valid MerchantStock merchantStock,
Errors errors) {
        if (errors.hasErrors()) {
            return ResponseEntity.status(400).body(new Api(errors.getFieldError().getDefaultMessage(),
400));
        }
        Boolean isMerchantStockAdd = merchantStockService.updatMerchantStock(merchantStock);
        if (! isMerchantStockAdd){
            return ResponseEntity.status(200).body(new Api("New MerchantStock update", 200));
        }
        return ResponseEntity.status(400).body(new Api("MerchantStock not found",400));

    }
    @DeleteMapping(".delet")
    public ResponseEntity deletaddMerchantStock(@PathVariable String id){
        if(merchantStockService.deletaddMerchantStock(id)){
            return ResponseEntity.status(200).body(new Api("MerchantStock deleted",200));
        }
        return ResponseEntity.status(400).body(new Api("MerchantStock not found",400));


    }



}
```