

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

import com.example.demo.Modele.Api;
import com.example.demo.Modele.Product;
import com.example.demo.Service.ProductService;
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/product")
```

```
@RequiredArgsConstructor
```

```
public class ProductController {
```

```
    @Autowired
```

```
    private final ProductService productService;
```

```
    private final ArrayList<Product> product = new ArrayList<>();
```

```
    @GetMapping(".get")
```

```
    public ResponseEntity<ArrayList<Product>> getProducts(){
```

```
        return ResponseEntity.status(200).body(product);
```

```
    }
```

```

@PostMapping(".add")

public ResponseEntity addProducts(@RequestBody @Valid Product product, @NotNull Errors errors,
boolean MerchantService){

    if (errors.hasErrors()){

        return ResponseEntity.status(400).body(new
Api(errors.getFieldError().getDefaultMessage(),400));

    }

    productService.addProduct(product, MerchantService);

    return ResponseEntity.status(200).body(new Api("New Student added",200));

}

@PutMapping(".update")

public ResponseEntity updateProduct(@RequestBody @Valid Product product, Errors errors) {

    if (errors.hasErrors()) {

        return ResponseEntity.status(400).body(new Api(errors.getFieldError().getDefaultMessage(),
400));

    }

    Boolean isProductAdd=productService.updateProduct(product);

    if (! isProductAdd){

        return ResponseEntity.status(200).body(new Api("New product update", 200));

    }

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

}

}

@DeleteMapping(".delet")

public ResponseEntity deletProduct(@PathVariable String id){

    if(productService.deletProduct(id)){

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

    }

}

```

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

```
}
```

```
}
```