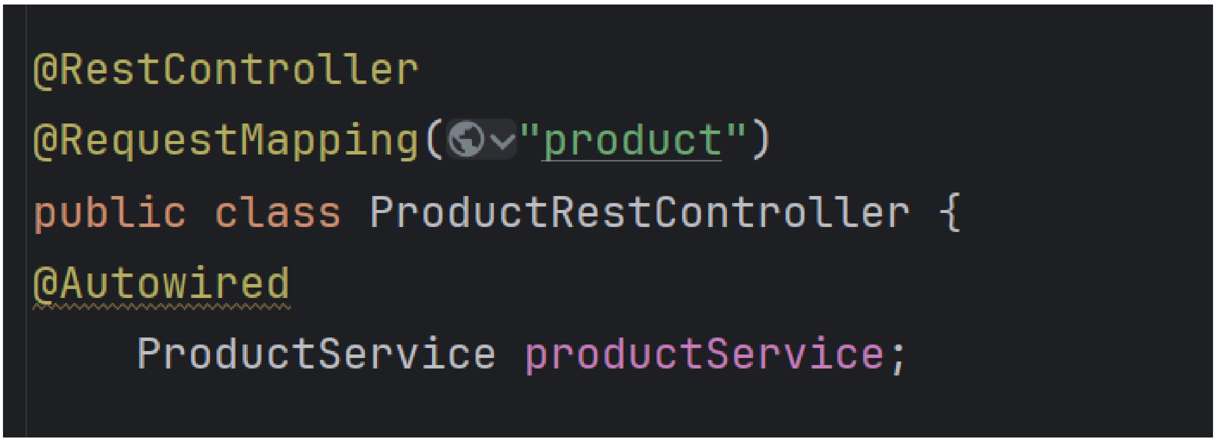
TP5 : RESTController Layer

1) Installer Postmean pour le Test :

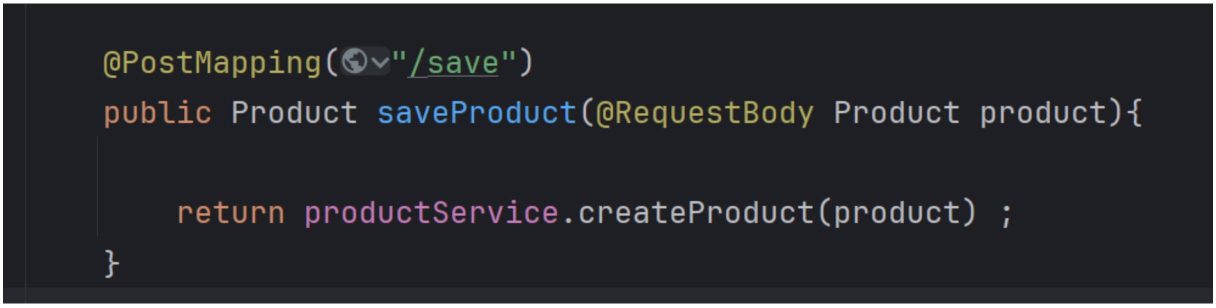
2) Créez un contrôleur REST ProductRestController :



```
@RestController
@RequestMapping("product")
public class ProductRestController {
@Autowired
    ProductService productService;
```

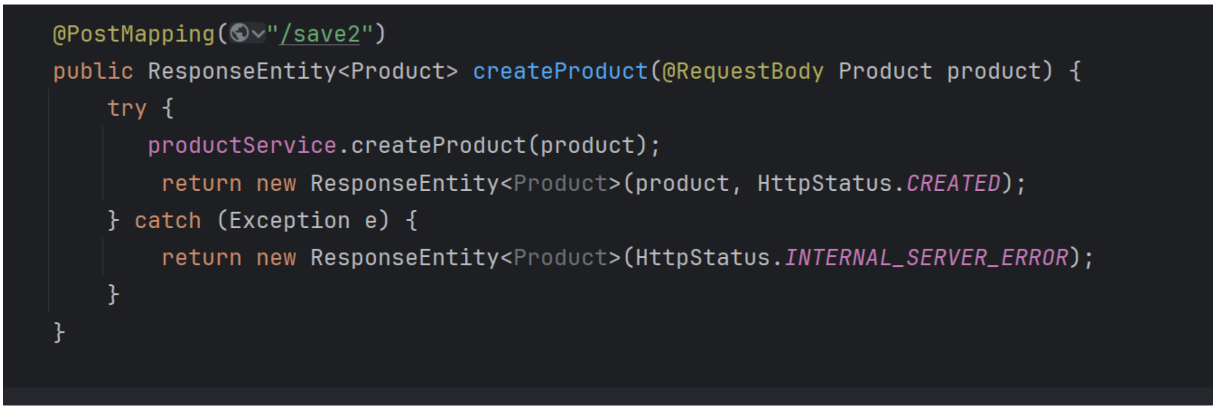
3) Ajouter les fonctions suivantes :

➢ AjouterProduct:



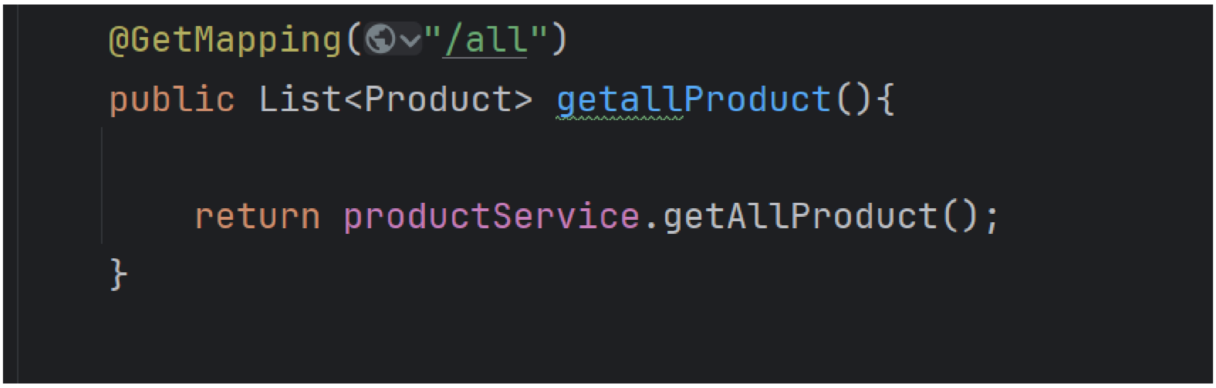
```
    @PostMapping("/save")
    public Product saveProduct(@RequestBody Product product){

        return productService.createProduct(product) ;
    }
```

Ou Bien



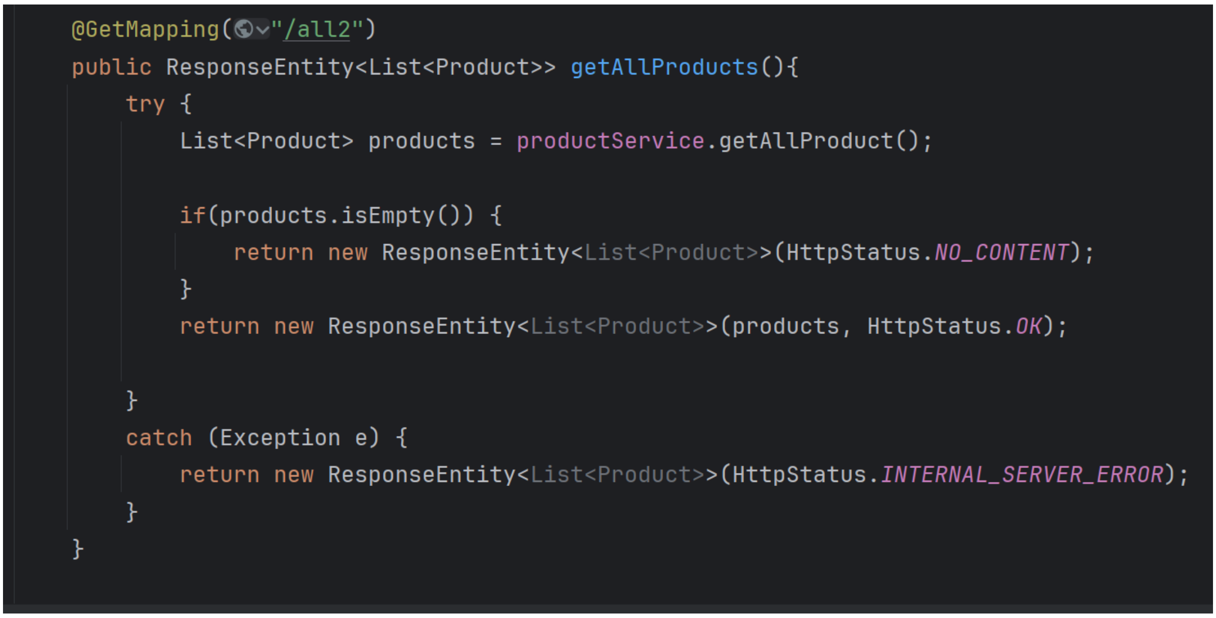
```
@PostMapping("/save2")
public ResponseEntity<Product> createProduct(@RequestBody Product product) {
    try {
        productService.createProduct(product);
         return new ResponseEntity<Product>(product, HttpStatus.CREATED);
    } catch (Exception e) {
         return new ResponseEntity<Product>(HttpStatus.INTERNAL_SERVER_ERROR);
    }
}
```

➢ ListProduct :



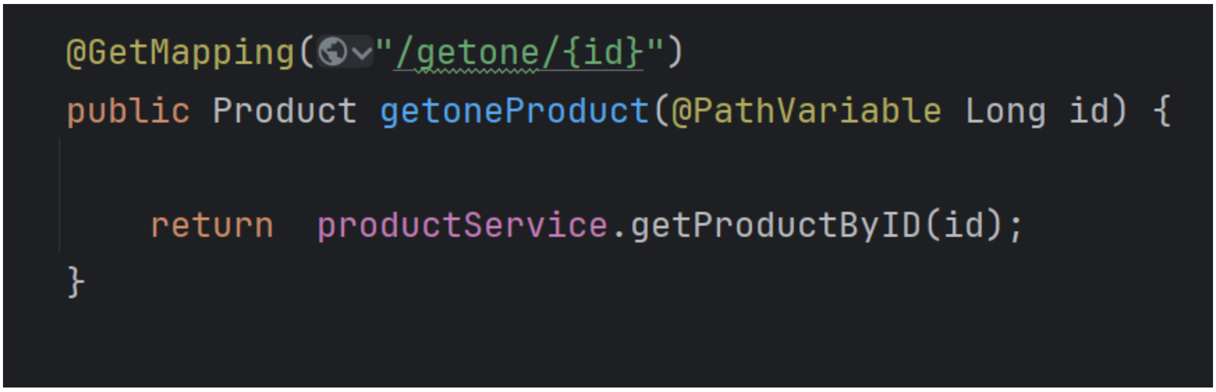
```java
@GetMapping("/all")
public List<Product> getallProduct(){

    return productService.getAllProduct();
}
```

Ou bien



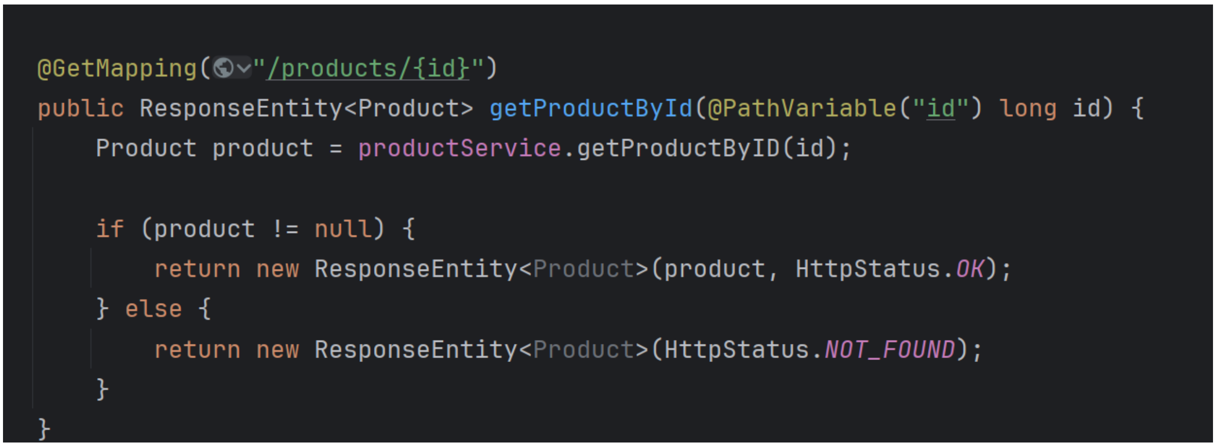
```java
@GetMapping("/all2")
public ResponseEntity<List<Product>> getAllProducts(){
    try {
        List<Product> products = productService.getAllProduct();

        if(products.isEmpty()) {
            return new ResponseEntity<List<Product>>(HttpStatus.NO_CONTENT);
        }
        return new ResponseEntity<List<Product>>(products, HttpStatus.OK);

    }
    catch (Exception e) {
        return new ResponseEntity<List<Product>>(HttpStatus.INTERNAL_SERVER_ERROR);
    }
}
```

- GetProductById :



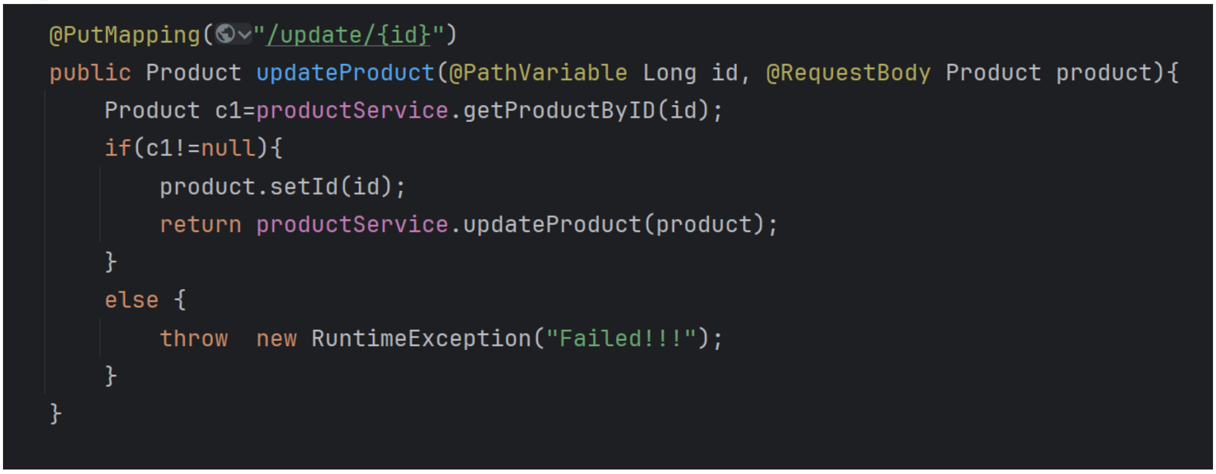
```java
@GetMapping("/getone/{id}")
public Product getoneProduct(@PathVariable Long id) {

    return  productService.getProductByID(id);
}
```

Ou bien



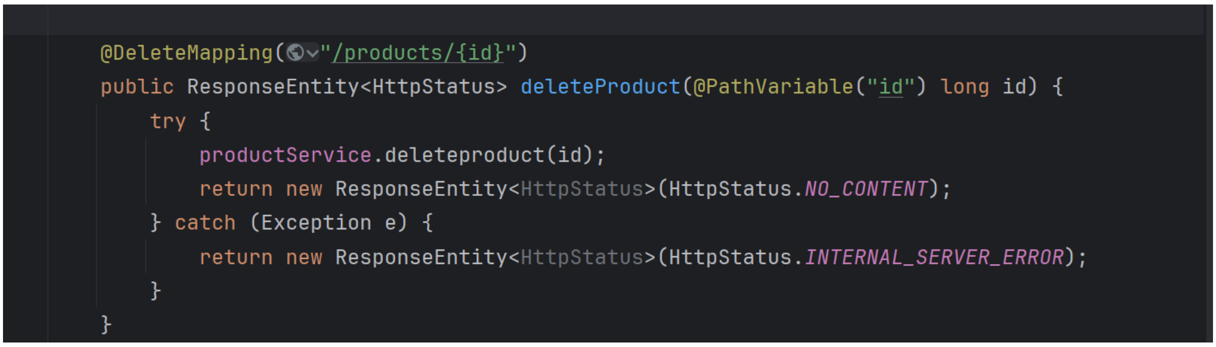
```java
@GetMapping("/products/{id}")
public ResponseEntity<Product> getProductById(@PathVariable("id") long id) {
    Product product = productService.getProductByID(id);

    if (product != null) {
        return new ResponseEntity<Product>(product, HttpStatus.OK);
    } else {
        return new ResponseEntity<Product>(HttpStatus.NOT_FOUND);
    }
}
```

- UpdateProduct



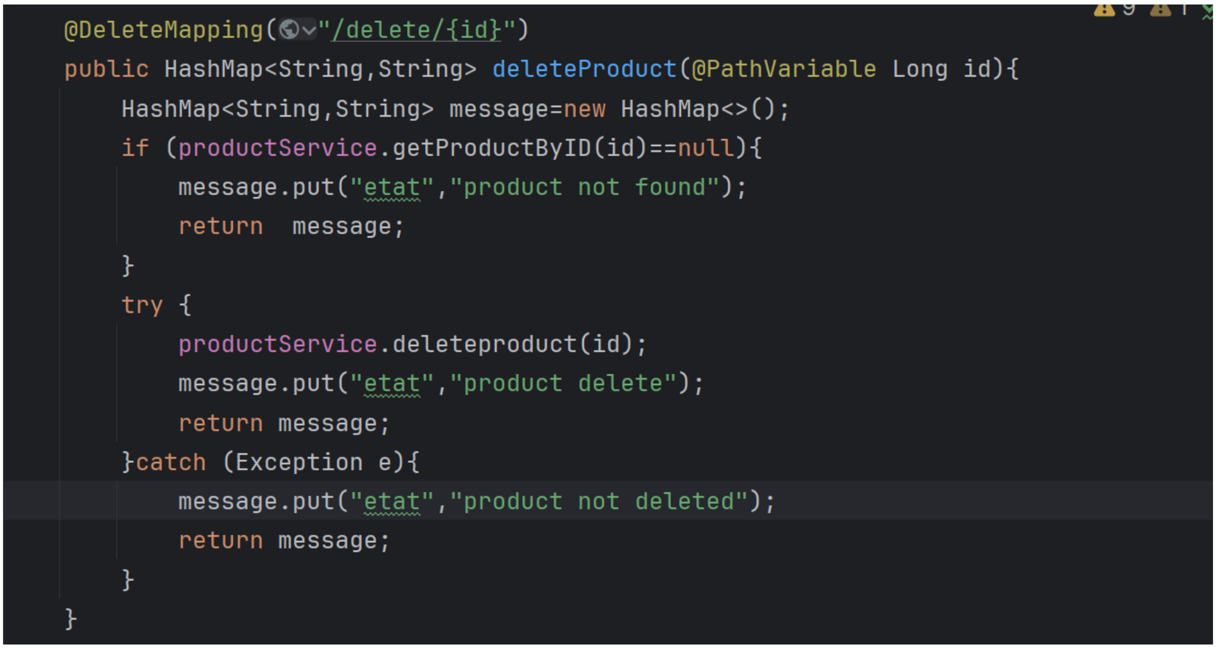
```java
@PutMapping("/update/{id}")
public Product updateProduct(@PathVariable Long id, @RequestBody Product product){
    Product c1=productService.getProductByID(id);
    if(c1!=null){
        product.setId(id);
        return productService.updateProduct(product);
    }
    else {
        throw  new RuntimeException("Failed!!!");
    }
}
```

- DeleteProduct



```java
@DeleteMapping("/products/{id}")
public ResponseEntity<HttpStatus> deleteProduct(@PathVariable("id") long id) {
    try {
        productService.deleteproduct(id);
        return new ResponseEntity<HttpStatus>(HttpStatus.NO_CONTENT);
    } catch (Exception e) {
        return new ResponseEntity<HttpStatus>(HttpStatus.INTERNAL_SERVER_ERROR);
    }
}
```

Ou bien



```java
@DeleteMapping("/delete/{id}")
public HashMap<String,String> deleteProduct(@PathVariable Long id){
    HashMap<String,String> message=new HashMap<>();
    if (productService.getProductByID(id)==null){
        message.put("etat","product not found");
        return  message;
    }
    try {
        productService.deleteproduct(id);
        message.put("etat","product delete");
        return message;
    }catch (Exception e){
        message.put("etat","product not deleted");
        return message;
    }
}
```