

Product Order API Documentation

Table of Contents

1. Overview	2
2. Product API	2
2.1. Create Product	2
2.2. CURL Request	2
2.3. HTTP Request	3
2.4. Request Body	3
2.5. HTTP Response	3
2.6. Response Body	3
2.7. Create Products	3
2.8. CURL Request	3
2.9. HTTP Request	3
2.10. Request Body	4
2.11. HTTP Response	4
2.12. Response Body	4
2.13. Delete Product	4
2.14. CURL Request	4
2.15. HTTP Request	4
2.16. Request Body	5
2.17. HTTP Response	5
2.18. Response Body	5
2.19. Delete Products	5
2.20. CURL Request	5
2.21. HTTP Request	5
2.22. Request Body	5
2.23. HTTP Response	5
2.24. Response Body	6
2.25. Get Product	6
2.26. CURL Request	6
2.27. HTTP Request	6
2.28. Request Body	6
2.29. HTTP Response	6
2.30. Response Body	6
2.31. Get Products	6
2.32. CURL Request	7
2.33. HTTP Request	7
2.34. Request Body	7

2.35. HTTP Response	7
2.36. Response Body	7
2.37. Update Product	7
2.38. CURL Request	8
2.39. HTTP Request	8
2.40. Request Body	8
2.41. HTTP Response	8
2.42. Response Body	8
2.43. Update Products	8
2.44. CURL Request	8
2.45. HTTP Request	9
2.46. Request Body	9
2.47. HTTP Response	9
2.48. Response Body	9
3. Order API	9
3.1. Create Order	9
3.2. Pay Order	10
3.3. Cancel Order	10

1. Overview

A scalable Spring Boot REST API for managing Products and Orders with real-time stock reservation, expiration handling (30 minutes), and Kafka-based messaging.

Features:

- Create, update, delete, and list Products
- Create, cancel, and pay for Orders

2. Product API

2.1. Create Product

REST API endpoint for creating a Product

2.2. CURL Request

```
$ curl 'http://localhost:8080/api/products' -i -X POST \
  -H 'Content-Type: application/json' \
  -d '{"name":"Test Product","price":10.0,"stockQuantity":100}'
```

2.3. HTTP Request

```
POST /api/products HTTP/1.1
Content-Type: application/json
Content-Length: 56
Host: localhost:8080
```

```
{"name":"Test Product","price":10.0,"stockQuantity":100}
```

2.4. Request Body

```
{"name":"Test Product","price":10.0,"stockQuantity":100}
```

2.5. HTTP Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 63
```

```
{"id":1,"name":"Test Product","price":10.0,"stockQuantity":100}
```

2.6. Response Body

```
{"id":1,"name":"Test Product","price":10.0,"stockQuantity":100}
```

2.7. Create Products

REST API endpoint for creating a list of Products

2.8. CURL Request

```
$ curl 'http://localhost:8080/api/products/batch' -i -X POST \
  -H 'Content-Type: application/json' \
  -d '[{"name":"Product 1","price":10.0,"stockQuantity":100},{ "name":"Product 2","price":20.0,"stockQuantity":200}]'
```

2.9. HTTP Request

```
POST /api/products/batch HTTP/1.1
```

```
Content-Type: application/json
Content-Length: 109
Host: localhost:8080
```

```
[{"name":"Product 1","price":10.0,"stockQuantity":100}, {"name":"Product 2","price":20.0,"stockQuantity":200}]
```

2.10. Request Body

```
[{"name":"Product 1","price":10.0,"stockQuantity":100}, {"name":"Product 2","price":20.0,"stockQuantity":200}]
```

2.11. HTTP Response

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 123
```

```
[{"id":1,"name":"Product 1","price":10.0,"stockQuantity":100}, {"id":2,"name":"Product 2","price":20.0,"stockQuantity":200}]
```

2.12. Response Body

```
[{"id":1,"name":"Product 1","price":10.0,"stockQuantity":100}, {"id":2,"name":"Product 2","price":20.0,"stockQuantity":200}]
```

2.13. Delete Product

REST API endpoint for deleting a Product

2.14. CURL Request

```
$ curl 'http://localhost:8080/api/products/1' -i -X DELETE
```

2.15. HTTP Request

```
DELETE /api/products/1 HTTP/1.1
Host: localhost:8080
```

2.16. Request Body

2.17. HTTP Response

```
HTTP/1.1 204 No Content
```

2.18. Response Body

2.19. Delete Products

REST API endpoint for deleting a list of Products

2.20. CURL Request

```
$ curl 'http://localhost:8080/api/products' -i -X DELETE \  
-H 'Content-Type: application/json' \  
-d '[1,2]'
```

2.21. HTTP Request

```
DELETE /api/products HTTP/1.1  
Content-Type: application/json  
Content-Length: 5  
Host: localhost:8080
```

```
[1,2]
```

2.22. Request Body

```
[1,2]
```

2.23. HTTP Response

```
HTTP/1.1 204 No Content
```

2.24. Response Body

2.25. Get Product

REST API endpoint for getting a Product

2.26. CURL Request

```
$ curl 'http://localhost:8080/api/products/1' -i -X GET \  
-H 'Content-Type: application/json'
```

2.27. HTTP Request

```
GET /api/products/1 HTTP/1.1  
Content-Type: application/json  
Host: localhost:8080
```

2.28. Request Body

2.29. HTTP Response

```
HTTP/1.1 200 OK  
Content-Type: application/json  
Content-Length: 63  
  
{ "id": 1, "name": "Test Product", "price": 10.0, "stockQuantity": 100 }
```

2.30. Response Body

```
{ "id": 1, "name": "Test Product", "price": 10.0, "stockQuantity": 100 }
```

2.31. Get Products

REST API endpoint for getting a list of Products

2.32. CURL Request

```
$ curl 'http://localhost:8080/api/products?page=0&size=10' -i -X GET \
-H 'Content-Type: application/json'
```

2.33. HTTP Request

```
GET /api/products?page=0&size=10 HTTP/1.1
Content-Type: application/json
Host: localhost:8080
```

2.34. Request Body

2.35. HTTP Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 437

{"content":[{"id":1,"name":"Product
1","price":10.0,"stockQuantity":100},{id":2,"name":"Product
2","price":20.0,"stockQuantity":200}],pageable":{"pageNumber":0,"pageSize":10,"sort":
{"empty":true,"unsorted":true,"sorted":false},"offset":0,"paged":true,"unpaged":false}
,"totalPages":1,"totalElements":2,"last":true,"size":10,"number":0,"sort":{"empty":tru
e,"unsorted":true,"sorted":false},"numberOfElements":2,"first":true,"empty":false}
```

2.36. Response Body

```
{"content":[{"id":1,"name":"Product 1","price":10.0,"stockQuantity":100},{id":2,"
name":"Product 2","price":20.0,"stockQuantity":200}],pageable":{"pageNumber":0,
"pageSize":10,"sort":{"empty":true,"unsorted":true,"sorted":false},"offset":0,"paged":
true,"unpaged":false},"totalPages":1,"totalElements":2,"last":true,"size":10,"number":
0,"sort":{"empty":true,"unsorted":true,"sorted":false},"numberOfElements":2,"first":tr
ue,"empty":false}
```

2.37. Update Product

REST API endpoint for updating a Product

2.38. CURL Request

```
$ curl 'http://localhost:8080/api/products' -i -X PUT \  
-H 'Content-Type: application/json' \  
-d '{"id":1,"name":"Updated Product","price":20.0,"stockQuantity":50}'
```

2.39. HTTP Request

```
PUT /api/products HTTP/1.1  
Content-Type: application/json  
Content-Length: 65  
Host: localhost:8080
```

```
{"id":1,"name":"Updated Product","price":20.0,"stockQuantity":50}
```

2.40. Request Body

```
{"id":1,"name":"Updated Product","price":20.0,"stockQuantity":50}
```

2.41. HTTP Response

```
HTTP/1.1 200 OK  
Content-Type: application/json  
Content-Length: 65
```

```
{"id":1,"name":"Updated Product","price":20.0,"stockQuantity":50}
```

2.42. Response Body

```
{"id":1,"name":"Updated Product","price":20.0,"stockQuantity":50}
```

2.43. Update Products

REST API endpoint for updating a list of Products

2.44. CURL Request

```
$ curl 'http://localhost:8080/api/products/batch' -i -X PUT \  
-H 'Content-Type: application/json' \  
-
```



```
-d '[{"id":1,"name":"Updated Product
1","price":15.0,"stockQuantity":150},{ "id":2,"name":"Updated Product
2","price":25.0,"stockQuantity":250}]'
```

2.45. HTTP Request

```
PUT /api/products/batch HTTP/1.1
Content-Type: application/json
Content-Length: 139
Host: localhost:8080
```

```
[{"id":1,"name":"Updated Product
1","price":15.0,"stockQuantity":150},{ "id":2,"name":"Updated Product
2","price":25.0,"stockQuantity":250}]
```

2.46. Request Body

```
[{"id":1,"name":"Updated Product 1","price":15.0,"stockQuantity":150},{ "id":2,"name":
"Updated Product 2","price":25.0,"stockQuantity":250}]
```

2.47. HTTP Response

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 139
```

```
[{"id":1,"name":"Updated Product
1","price":15.0,"stockQuantity":150},{ "id":2,"name":"Updated Product
2","price":25.0,"stockQuantity":250}]
```

2.48. Response Body

```
[{"id":1,"name":"Updated Product 1","price":15.0,"stockQuantity":150},{ "id":2,"name":
"Updated Product 2","price":25.0,"stockQuantity":250}]
```

3. Order API

3.1. Create Order

REST API endpoint for creating an Order

```
POST /api/orders HTTP/1.1
Content-Type: application/json;charset=UTF-8
Content-Length: 40
Host: localhost:8080
```

```
{"items":[{"productId":1,"quantity":2}]}
```

```
HTTP/1.1 201 Created
Content-Type: application/json
Content-Length: 96
```

```
{"id":1,"status":"CREATED","createdAt":"2025-06-16T22:24:01.380282","paidAt":null,"products":[]}
```

3.2. Pay Order

REST API endpoint for paying for an Order

```
POST /api/orders/1/pay HTTP/1.1
Host: localhost:8080
Content-Type: application/x-www-form-urlencoded
```

```
HTTP/1.1 200 OK
```

3.3. Cancel Order

REST API endpoint for canceling an Order

```
POST /api/orders/1/cancel HTTP/1.1
Host: localhost:8080
Content-Type: application/x-www-form-urlencoded
```

```
HTTP/1.1 200 OK
```