

Documentación API RESTful - Sistema de Gestión de Productos

Proyecto: Lab09 - API REST con Spring Security

Fecha: 10 de noviembre de 2025

Base URL: <http://localhost:8080/api/product>

Autenticación: HTTP Basic (Usuario: admin, Contraseña: pass123)

Información General

API RESTful para gestión de productos con operaciones CRUD completas. Todos los endpoints requieren autenticación HTTP Basic.

Tecnologías:

- Spring Boot 3.5.7
- Spring Security (HTTP Basic)
- MySQL (Base de datos Northwind)
- Formato de datos: JSON

Endpoints Disponibles

1. GET /api/product - Listar Todos los Productos

URL Completa: <http://localhost:8080/api/product>

Argumentos de Entrada:

- Ninguno

Respuesta Exitosa (200 OK):

```
[  
  {  
    "productId": 1,  
    "productName": "Chais",  
    "supplierId": 1,  
    "categoryId": 1,  
    "unit": "10 boxes x 20 bags",  
    "price": 18.0,  
    "category": {  
      "categoryId": 1,  
      "categoryName": "Beverages",  
      "description": "Soft drinks, coffees, teas, beers, and ales"  
    },  
    "supplier": {  
      "supplierId": 1,  
      "supplierName": "Exotic Liquid",  
      "contactName": "Charlotte Cooper",  
      "city": "Londona",  
      "country": "UK"  
    }  
  }  
]
```

Códigos de Estado HTTP:

- 200 OK: Lista de productos devuelta exitosamente
- 401 Unauthorized: Credenciales faltantes o inválidas
- 500 Internal Server Error: Error en el servidor

2. GET /api/product/{id} - Obtener Producto por ID

URL Completa: <http://localhost:8080/api/product/{id}>

Argumentos de Entrada:

- Parámetro de ruta: id (Integer) - ID del producto

Ejemplo: /api/product/1

Respuesta Exitosa (200 OK):

```
{  
    "productId": 1,  
    "productName": "Chais",  
    "supplierId": 1,  
    "categoryId": 1,  
    "unit": "10 boxes x 20 bags",  
    "price": 18.0,  
    "category": { ... },  
    "supplier": { ... }  
}
```

Códigos de Estado HTTP:

- 200 OK: Producto encontrado
- 404 Not Found: Producto no existe
- 401 Unauthorized: Credenciales inválidas
- 500 Internal Server Error: Error en el servidor

3. POST /api/product - Crear Nuevo Producto

URL Completa: <http://localhost:8080/api/product>

Argumentos de Entrada:

- Content-Type: application/json
- Cuerpo (JSON):

```
{  
    "productName": "Nuevo Producto",  
    "supplierId": 1,  
    "categoryId": 1,  
    "unit": "10 units",  
    "price": 25.50  
}
```

Campos Requeridos:

- productName (String)
- supplierId (Integer)
- categoryId (Integer)
- unit (String)
- price (Double)

Respuesta Exitosa (201 Created):

```
{  
    "productId": 89,  
    "productName": "Nuevo Producto",  
    "supplierId": 1,  
    "categoryId": 1,  
    "unit": "10 units",  
    "price": 25.50,  
    "category": { ... },  
    "supplier": { ... }  
}
```

Códigos de Estado HTTP:

- 201 Created: Producto creado exitosamente
- 400 Bad Request: Datos inválidos
- 401 Unauthorized: Credenciales inválidas
- 500 Internal Server Error: Error en el servidor

4. PUT /api/product/{id} - Actualizar Producto

URL Completa: <http://localhost:8080/api/product/{id}>

Argumentos de Entrada:

- **Parámetro de ruta:** id (Integer) - ID del producto a actualizar
- **Content-Type:** application/json
- **Cuerpo (JSON):**

```
{  
    "productName": "Producto Actualizado",  
    "supplierId": 2,  
    "categoryId": 1,  
    "unit": "20 units",  
    "price": 30.00  
}
```

Respuesta Exitosa (200 OK):

```
{  
    "productId": 1,  
    "productName": "Producto Actualizado",  
    "supplierId": 2,  
    "categoryId": 1,  
    "unit": "20 units",  
    "price": 30.00,  
    "category": { ... },  
    "supplier": { ... }  
}
```

Códigos de Estado HTTP:

- 200 OK: Producto actualizado
- 404 Not Found: Producto no existe
- 400 Bad Request: Datos inválidos
- 401 Unauthorized: Credenciales inválidas

5. DELETE /api/product/{id} - Eliminar Producto

URL Completa: <http://localhost:8080/api/product/{id}>

Argumentos de Entrada:

- **Parámetro de ruta:** id (Integer) - ID del producto a eliminar

Ejemplo: /api/product/88

Respuesta Exitosa (204 No Content):

- Sin contenido (vacío)

Códigos de Estado HTTP:

- 204 No Content: Producto eliminado exitosamente
- 404 Not Found: Producto no existe
- 401 Unauthorized: Credenciales inválidas
- 500 Internal Server Error: Error en el servidor

Modelos de Datos

Product

```
{  
    "productId": Integer,  
    "productName": String,  
    "supplierId": Integer,  
    "categoryId": Integer,  
    "unit": String,  
    "price": Double,  
    "category": { ... },  
    "supplier": { ... }  
}
```

Category

```
{  
    "categoryId": Integer,  
    "categoryName": String,  
    "description": String  
}
```

Supplier

```
{  
    "supplierId": Integer,  
    "supplierName": String,  
    "contactName": String,  
    "address": String,  
    "city": String,  
    "country": String,  
    "phone": String  
}
```

Resumen de Códigos HTTP

Código	Descripción
200	OK - Solicitud exitosa
201	Created - Recurso creado
204	No Content - Eliminación exitosa
400	Bad Request - Datos inválidos
401	Unauthorized - Sin autenticación
404	Not Found - Recurso no encontrado
500	Internal Server Error - Error del servidor

Ejemplos de Uso

Listar Productos

```
curl -u admin:pass123 http://localhost:8080/api/product
```

Obtener Producto por ID

```
curl -u admin:pass123 http://localhost:8080/api/product/1
```

Crear Producto

```
curl -X POST http://localhost:8080/api/product
-u admin:pass123
-H "Content-Type: application/json"
-d '{ "productName": "Nuevo Producto", "supplierId": 1, "categoryId": 1, "unit": "10 units", "price": 25.50 }'
```

Actualizar Producto

```
curl -X PUT http://localhost:8080/api/product/1
-u admin:pass123
-H "Content-Type: application/json"
-d '{ "productName": "Producto Actualizado", "supplierId": 2, "categoryId": 1, "unit": "20 units", "price": 30.00 }'
```

Eliminar Producto

```
curl -X DELETE http://localhost:8080/api/product/88
-u admin:pass123
```
