

Manual de Uso de la API de Gestión de Stock

Introducción

Esta API permite la gestión eficiente de productos, compras y ventas en la tienda. Entre sus funcionalidades más destacadas se encuentran:

- **Consulta del historial de movimientos de stock de un producto.**
- **Obtención del producto con mayor stock disponible.**
- **Gestión de listas de compras y ventas con actualización automática del stock.**

Inicialmente descargar el proyecto y abrirlo en Eclipse o Spring Tool Suite, esperar que el maven descargue las dependencias marcadas en pom.xml.

Correr la aplicacion y a continuacion Ubicar en el navegador el siguiente link:

<http://localhost:8080/swagger-ui/index.html>

SWAGGER

Este trabajo adopta como herramienta a Swagger para realizar las pruebas pertinentes de las funcionalidades.

Inventario API ^{1.0} QAS 3.0

/v3/api/docs

API para la gestión de inventario

Servers

producto-controller

PUT /api/productos/{id}/stock

GET /api/productos

POST /api/productos

GET /api/productos/mayorstock

DELETE /api/productos/{id}

Endpoints Destacados

1. Historial de Movimientos de Stock

Inicialmente como guía, realizo un GET de productos

GET /api/productos

busco y selecciono el código de barras de algún producto existente que tiene 2 movimientos (1 de compra y 1 de venta), ejemplo: 4634734734

```
{
  {
    "id": 4634734734,
    "descripcion": "Cuaderno A4",
    "categoria": {
      "id": 1,
      "descripcion": "Papeleria"
    },
    "costo": 100,
    "precio": 150,
    "stock": 30
  },
  {
    "id": 5485484585,
    "descripcion": "Juguete de construcción",
    "categoria": {
      "id": 3,
      "descripcion": "Jugueteria"
    }
  }
}
```

A continuacion voy al endpoint del Stock-Controller y pego el codigo de barras seleccionado

stock-controller

GET

/api/stock/historial/{codigoBarras}

Parameters

Name	Description
codigoBarras * required integer(\$int64) (path)	<input type="text" value="4634734734"/>

Endpoint: GET /api/stock/historial/{codigoBarras}

Descripción: Este endpoint permite consultar el historial completo de movimientos de stock (compras y ventas) de un producto específico.

Parámetros:

- **codigoBarras:** Código de barras único del producto.

Ejemplo de Uso:

GET /api/stock/historial/4634734734

Respuesta Esperada:

Response body

```
[
  {
    "fecha": "2025-01-10T00:00:00",
    "operacion": "COMPRA",
    "cantidad": 30,
    "stockActual": 30
  },
  {
    "fecha": "2025-02-01T00:00:00",
    "operacion": "VENTA",
    "cantidad": 5,
    "stockActual": 25
  }
]
```

Como se puede observar, el stockActual se actualiza a medida que se contabilizan los movimientos. Estos toman la fecha en el cual se hacen dichos movimientos.

2. Producto con Mayor Stock

Endpoint: GET /api/productos/mayorstock

Descripción: Devuelve el producto con la mayor cantidad de stock disponible en la tienda.

Ejemplo de Uso:

GET /api/productos/mayorstock

Respuesta Esperada:

Response body

```
{
  "codigo": 8685644584,
  "descripcion": "Lapiz HB",
  "stock": 100
}
```

Gestión de Compras y Ventas

1. Lista de Compras

Endpoint: GET /api/compras

Descripción: Devuelve la lista completa de compras realizadas, actualizando el stock de los productos comprados.

Ejemplo de Uso:

GET /api/compras

Respuesta Esperada:

```
{
  "id": 1,
  "producto": {
    "id": 4634734734,
    "descripcion": "Cuaderno A4",
    "categoria": {
      "id": 1,
      "descripcion": "Papeleria"
    },
    "costo": 100,
    "precio": 150,
    "stock": 30
  },
  "proveedor": {
    "id": 24634634436,
    "nombre": "Papeleria Central",
    "descripcion": "Proveedor especializado en papeleria y escritura"
  },
  "fecha": "2025-01-10",
  "cantidad": 30
},
{
  "id": 2,
  "producto": {
    "id": 8685644584,
    "descripcion": "Lapiz HB",
    "categoria": {
      "id": 2,
      "descripcion": "Escritura"
    }
  }
}
```

2. Lista de Ventas

Endpoint: GET /api/ventas

Descripción: Devuelve la lista completa de ventas realizadas, actualizando el stock de los productos vendidos.

Ejemplo de Uso:

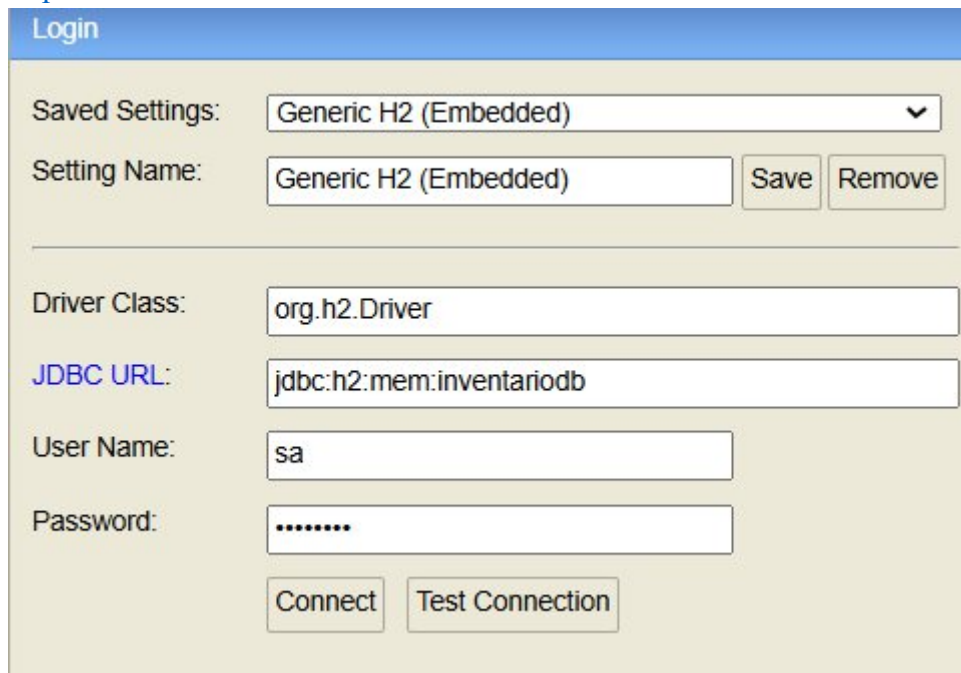
GET /api/ventas

Respuesta Esperada:

```
[
  {
    "id": 1,
    "producto": {
      "id": 4634734734,
      "descripcion": "Cuaderno A4",
      "categoria": {
        "id": 1,
        "descripcion": "Papeleria"
      },
      "costo": 100,
      "precio": 150,
      "stock": 30
    },
    "fecha": "2025-02-01",
    "cantidad": 5
  }
]
```

Para ver el gestor de base de datos h2, ubicar este link en el navegador:

<http://localhost:8080/h2-console/>

The image shows the H2 database console's login interface. At the top is a blue header with the word 'Login'. Below it, there's a section for 'Saved Settings' with a dropdown menu currently showing 'Generic H2 (Embedded)'. Underneath, a 'Setting Name' field also contains 'Generic H2 (Embedded)', with 'Save' and 'Remove' buttons to its right. A horizontal line separates this from the main configuration area. This area includes fields for 'Driver Class' (containing 'org.h2.Driver'), 'JDBC URL' (containing 'jdbc:h2:mem:inventariodb'), 'User Name' (containing 'sa'), and 'Password' (containing seven dots). At the bottom of this section are two buttons: 'Connect' and 'Test Connection'.

Asegurarse que la URL sea como esta indicada jdbc:h2:mem:inventariodb

Usuario: sa, Password: password

Conclusión

Este sistema proporciona una administración eficiente del inventario, facilitando el seguimiento del stock y las operaciones de compra y venta en tiempo real. Ante cualquier duda o mejora, no dudes en consultarnos.