

Documentación de la API - Carrito de Compras

Introducción

Esta documentación técnica describe el funcionamiento de la API RESTful del sistema de carrito de compras y permite a los usuarios:

- Listar productos disponibles.
- Agregar productos al carrito.
- Visualizar el contenido del carrito.
- Calcular totales de compra.
- Simular un proceso de checkout (confirmación y pago).

Carrito de compra.

1. Obtener Productos del Carrito

Método: GET

Ruta: /api/cart/:userId

Descripción

Este endpoint devuelve todos los productos del carrito de un usuario específico, incluyendo cantidades y el total a pagar.

```
{
  "message": "Cart retrieved successfully",
  "items": [
    {
      "id": 10,
      "quantity": 2,
      "product": {
        "id": 1,
        "name": "Camisa Azul",
        "price": 29.99,
        "imageUrl": "https://..."
      },
    },
  ],
  "totalItems": 2,
  "totalPrice": 59.98
}
```

2. Agregar Producto al Carrito

Método: POST

Ruta: /api/cart/add

Descripción

Este endpoint agrega un producto al carrito de un usuario.

Cuerpo de la solicitud:

```
{  
    "userId": "abc123",  
    "productId": 101,  
    "quantity": 1  
}
```

Respuesta:

```
{  
    "message": "Producto agregado al carrito",  
    "success": true  
}
```

3. Actualizar Cantidad en el Carrito

Método: PUT

Ruta: /api/Cart/quantity

Descripción

Este endpoint actualiza la cantidad de un producto específico en el carrito de un usuario.

El campo delta indica el cambio en la cantidad:

- $\text{delta} > 0$: Aumenta la cantidad
- $\text{delta} < 0$: Disminuye la cantidad
- Si al restar la cantidad queda 0 o menos, el producto puede eliminarse automáticamente.

Cuerpo de la solicitud:

```
{  
  "userId": 1,  
  "productId": 101,  
  "delta": 1  
}
```

Respuesta:

```
{  
  "message": "Cantidad actualizada correctamente",  
  "success": true,  
  "error": false  
}
```

4. Eliminar Producto del Carrito

Método: DELETE

Ruta: /api/cart/:userId/:productId

Descripción

Este endpoint elimina un producto específico del carrito de un usuario.

Cuerpo de la solicitud:

```
{  
  "userId": "1",  
  "productId": "1"  
}
```

Respuesta:

```
{  
  "message": " Producto eliminado del carrito",  
  "success": true  
}
```

5. Confirmar Orden (Checkout)

Método: POST

Ruta: /api/Cart/checkout?userId

Descripción

Este endpoint simula el pago exitoso. Se espera que el carrito del usuario ya esté cargado previamente.

Cuerpo de la solicitud:

```
{  
    "userId": "1",  
}
```

Respuesta:

```
{  
    "message": "Payment successful",  
    "success": true,  
    "error": false  
}
```

Productos

1. Obtener productos

Método: GET

Ruta: /api/Products

Descripción

Este endpoint devuelve una lista con todos los productos disponibles en el sistema.

Respuesta:

```
[
  {
    "id": 1,
    "name": "Camisa Azul",
    "description": "Camisa 100% algodón",
    "price": 29.99,
    "imageUrl": "https://..."
  },
  {
    "id": 1,
    "name": "Camisa Azul",
    "description": "Camisa 100% algodón",
    "price": 29.99,
    "imageUrl": "https://..."
  }
]
```

2. Obtener Producto por ID

Método: GET

Ruta: /api/Products/:productId

Descripción

Este endpoint permite obtener los detalles de un producto específico utilizando su ID.

Cuerpo de la solicitud:

```
{  
    productId ": "1",  
}
```

Respuesta:

```
{  
    "id": 1,  
    "name": "Laptop Dell XPS 15",  
    "description": "Laptop potente para trabajo y gaming.",  
    "price": 1500,  
    "imageUrl": "/images/laptop1.jpg"  
}
```


Usuarios

1. Registra un usuario

Método: POST

Ruta: /api/User/register

Descripción

Este endpoint registra un nuevo usuario en el sistema.

Cuerpo de la solicitud:

```
{  
    "name": "string",  
    "username": "string",  
    "password": "string"  
}
```

Respuesta:

```
{  
    "message": "user successfully registered ",  
    "data": {  
        "id": 2,  
        "userName": "string",  
        "name": "string"  
    },  
    "error": false,  
    "success": true  
}
```

2. Login de Usuario

Método: POST

Ruta: /api/User/login

Descripción

Este endpoint permite iniciar sesión a un usuario existente en el sistema.

Cuerpo de la solicitud:

```
{  
    "username": "string",  
    "password": "string"  
}
```

Respuesta:

```
{  
    "message": "Login successfully",  
    "data": {  
        "id": 2,  
        "userName": "string",  
        "name": "string"  
    },  
    "error": false,  
    "success": true  
}
```

3. Obtener un usuario específico.

Método: GET

Ruta: /api/User/:userId

Descripción

Este endpoint obtiene los datos de un solo usuario del sistema, a partir de su userId.

Cuerpo de la solicitud:

```
{  
    "userId": "2"  
}
```

Respuesta:

```
{  
    "message": "User found ",  
    "data": {  
        "id": 2,  
        "userName": "string",  
        "name": "string"  
    },  
    "error": false,  
    "success": true  
}
```

4. Obtener Usuarios

Método: GET

Ruta: /api/User

Descripción

Este endpoint devuelve la lista de todos los usuarios registrados en el sistema.

Respuesta:

```
{
  "message": "Users",
  "data": [
    {
      "id": 2,
      "userName": "string",
      "name": "string"
    },
    {
      "id": 3,
      "userName": "string1",
      "name": "string"
    }
  ],
  "error": false,
  "success": true
}
```