

DEVMART API

Wilson Estrada, Wesley Vanegas y Juana Solano

DevMart API permite crear, actualizar, eliminar y obtener recursos utilizados en la plataforma de DevMart Web.

Base URL

<https://api.devmart.com/v1/>

Tipos de Solicitudes HTTP

Se accede a los recursos de datos mediante solicitudes HTTP estándar. La API web utiliza los siguientes verbos HTTP:

Método	Acción
GET	Devuelve recursos
POST	Crea recursos
PUT	Cambia o reemplaza información en los recursos
DELETE	Elimina recursos
PATCH	Actualización parcial de un recurso

Tipos de Respuestas

Status Code	Descripción
200	Ok - La solicitud ha sido exitosa
204	No content – La solicitud ha sido exitosa pero no devuelve un mensaje.
400	Bad Request – El servidor no pudo comprender la solicitud debido a una sintaxis incorrecta. El cuerpo del mensaje contendrá más información.
401	Unauthorized - La solicitud requiere autenticación del usuario o, si la solicitud incluía credenciales de autorización, se ha rechazado la autorización para esas credenciales.
403	Forbidden - El servidor entendió la solicitud, pero se niega a cumplirla.
404	Not Found - No se pudo encontrar el recurso solicitado.
422	Unprocessable Content - Indica que el servidor entendió el tipo de contenido de la solicitud y que la sintaxis del contenido de la solicitud era correcta, pero no pudo procesar las instrucciones contenidas.
409	Conflict – Conflicto con el estado actual del recurso objetivo
500	Internal Server Error – El servidor web encontró un problema inesperado y no puede completar la solicitud
550	Non-existent recipient - Suele ocurrir porque algo va mal con la dirección de correo electrónico o la configuración del destinatario.

Auth API

Get Session	
Tipo de Servicio	REST
Método HTTP	POST
URI	/auth

Descripción

Método que permite obtener token de autorización para otorgar el permiso de consumo de los otros métodos al usuario logueado.

Body

Nombre	Tipo	Requerido	Descripción
email	string	Sí	Correo electrónico del usuario
password	string	Sí	Contraseña

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "userId" : 54,
  "firstName" : "Juana",
  "lastName" : "Solano",
  "isAdmin" : true,
  "token":
  "7Oa5u0iD5ipMCK3oHAT1yhAg6bRaYIbxt4tAll0fzF7QlvTfxrkmxQe62y1DIIkS"
}
```

– Respuestas de error

400 - Bad Request

Si la contraseña es incorrecta

```
{
  "statusCode": "400",
  "message": "The password entered is incorrect."
}
```

404 – Not Found

Si el email no existe en la base de datos:

```
{  
  "statusCode": "404",  
  "message": "The email address entered is not connected to an account."  
}
```

Users API

Create User	
Tipo de Servicio	REST
Método HTTP	POST
URI	/users

Descripción

Método que permite registrar usuarios en la aplicación.

Body

Nombre	Tipo	Requerido	Descripción
email	string	Sí	Correo electrónico del usuario
password	string	Sí	Contraseña
firstName	string	Sí	Primer nombre del usuario
lastName	string	Sí	Primer apellido del usuario
city	string	No	Ciudad
mobilePhone	string	No	Número de celular
address	string	No	Dirección
zipCode	number	No	Código postal
isAdmin	boolean	Sí	Campo para identificar si el usuario va a ser un vendedor

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
```

```
"statusCode": "200",  
"message": "User created successfully."  
}
```

– Respuestas de error

400 - Bad Request

Si algunos de los parámetros requeridos son nulos o vacíos:

```
{  
  "statusCode": "400",  
  "message": "The following parameters are required: email, password, firstName  
and lastName."  
}
```

Si la contraseña no cumple con los parámetros requeridos:

```
{  
  "statusCode": "400",  
  "message": "The password must be at least 8 characters long, with uppercase,  
lowercase, numbers, and symbols."  
}
```

Si el teléfono no tiene formato colombiano válido:

```
{  
  "statusCode": "400",  
  "message": "The mobile phone must be in the format: +57XXXXXXXXXX"  
}
```

Si el código postal no cumple con una longitud de 6 o 9 dígitos:

```
{  
  "statusCode": "400",  
  "message": "The postal code must be 6 or 9 digits long."  
}
```

422 - Unprocessable Entity

Si el email proporcionado ya existe:

```
{  
  "statusCode": "422",
```

```
"message": "The email provided already exists."
}
```

550 - Non-existent recipient

Si el dominio del email no existe:

```
{
  "statusCode": "550",
  "message": "The email provided does not exist."
}
```

Get User	
Tipo de Servicio	REST
Método HTTP	GET
URI	/users/{id}

Descripción

Método que permite obtener información del usuario.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Bearer token	Sí	Es el token entregado por el servicio de Autenticación

Parameters

Nombre	Tipo	Requerido	Descripción
id	number	Sí	Identificación del usuario

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "id" : "1",
  "firstName" : "Juana",
  "lastName" : "Solano",
  "email" : "juana.solano@correo.com",
  "address" : "Calle 39C #XX - XX",
  "city" : "Medellín",
}
```

```
{
  "zipCode": 123455,
  "mobilePhone": "31231184XX",
  "isActive": true
}
```

– Respuestas de error

401 – Unauthorized

Si el token ha sido expirado.

```
{
  "statusCode": "401",
  "message": "Bad or expired token."
}
```

403 – Forbidden

Si el usuario logueado no debe tener acceso a la información del usuario que busca.

```
{
  "statusCode": "403",
  "message": "Do not have permission to this action."
}
```

404 - Not Found

Si el usuario no existe.

```
{
  "statusCode": "404",
  "message": "User not found."
}
```

Update User	
Tipo de Servicio	REST
Método HTTP	PUT
URI	/users/{id}

Descripción

Método que permite actualizar información del usuario.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Bearer token	Sí	Es el token entregado por el

			servicio de Autenticación
--	--	--	---------------------------

Parameters

Nombre	Tipo	Requerido	Descripción
id	number	Sí	Identificación del usuario

Body

Nombre	Tipo	Requerido	Descripción
email	string	Sí	Correo electrónico del usuario
password	string	No	Contraseña
firstName	string	No	Primer nombre del usuario
lastName	string	No	Primer apellido del usuario
city	string	No	Ciudad
mobilePhone	string	No	Número de celular
address	string	No	Dirección
zipCode	number	No	Código postal
isActive	boolean	Sí	Estado del usuario

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "id" : "1",
  "firstName" : "Juana",
  "lastName" : "Solano",
  "email" : "juana.solano@correo.com",
  "address" : "Calle 39C #XX - XX",
  "city" : "Medellín",
  "zipCode": 123455,
  "mobilePhone" : "31231184XX",
  "password": "*****",
  "isActive": true
}
```

– Respuestas de error

400 - Bad Request

Si absolutamente todos los parámetros del cuerpo o alguno de los parámetros requeridos se envían nulo o vacío:

```
{  
  "statusCode": "400",  
  "message": "Nothing has been updated because no data has been provided."  
}
```

Si la contraseña no cumple con los parámetros requeridos:

```
{  
  "statusCode": "400",  
  "message": "The password must be at least 8 characters long, with uppercase,  
  lowercase, numbers, and symbols."  
}
```

Si el teléfono no tiene formato colombiano válido:

```
{  
  "statusCode": "400",  
  "message": "The mobile phone must be in the format: +57XXXXXXXXXX"  
}
```

Si el código postal no cumple con una longitud de 6 o 9 dígitos:

```
{  
  "statusCode": "400",  
  "message": "The postal code must be 6 or 9 digits long."  
}
```

401 – Unauthorized

Si el token ha sido expirado.

```
{  
  "statusCode": "401",  
  "message": "Bad or expired token."  
}
```

403 – Forbidden

Si el usuario logueado no debe tener permisos para actualizar la información del usuario.

```
{  
  "statusCode": "403",
```



```
"message": "Do not have permission to this action."
}
```

404 - Not Found

Si no se encuentra el usuario:

```
{
  "statusCode": "404",
  "message": "User not found."
}
```

422 - Unprocessable Entity

Si el email proporcionado ya existe:

```
{
  "statusCode": "422",
  "message": "The email provided already exists."
}
```

550 - Non-existent recipient

Si el dominio del email no existe:

```
{
  "statusCode": "550",
  "message": "The email provided does not exist."
}
```

Orders API

Create Order	
Tipo de Servicio	REST
Método HTTP	POST
URI	/orders

Descripción

Método que permite crear órdenes.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Bearer token	Sí	Es el token entregado por el servicio de Autenticación

Body

Nombre	Tipo	Requerido	Descripción
userId	number	Sí	ID del usuario que creó la orden
products	Object { "Id" : number, "count": number, "supplierId": number }	Sí	Productos escogidos por el usuario. Se guarda el Id del producto, la cantidad (debe ser mayor a 0) y el Id del usuario que lo vende.
paymentMethod	number	Sí	Método de pago escogido por el usuario (Contado(1), tarjeta(2), transferencia(3))
address	string	Sí	Dirección a la que se va a hacer el envío.

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "id" : 111,
  "userId" : 1,
  "products" : [
    {
      "id" : 23,
      "count" : 2,
      "supplierId" : 32
    }
  ]
}
```

```

    },
    {
        "id" : 23,
        "count" : 2,
        "supplierId" : 32
    }
],
"paymentMethod" : 2,
"total" : $400.000,
"address" : "Calle 39C #XX - XX",
"created" : "2013-07-02T21:36:25.344Z",
"lastUpdated": "2013-07-02T21:36:25.344Z"
}

```

– Respuestas de error

400 - Bad Request

Si algunos de los parámetros se envían nulos o vacíos:

```

{
    "statusCode": "400",
    "message": "Missing one of the following parameters: User ID, Products, Payment Method or Address."
}

```

Si la cantidad de un producto en específico es 0:

```

{
    "statusCode": "400",
    "productId": 23,
    "message": "The count for Protector Solar Oil Control SPF 50 x 130ml is zero."
}

```

Si no hay stock para proveer el producto tomado:

```

{

```

```
"statusCode": "400",
"productId": 23,
"message": "The Protector Solar Oil Control SPF 50 x 130ml product is out of
stock."
}
```

401 – Unauthorized

Si el token ha sido expirado o el usuario no está logueado.

```
{
  "statusCode": "401",
  "message": "Bad or expired token."
}
```

403 – Forbidden

Si el usuario logueado no debe tener acceso a crear la orden.

```
{
  "statusCode": "403",
  "message": "Do not have permission to this action."
}
```

Get Order	
Tipo de Servicio	REST
Método HTTP	GET
URI	/orders/{id}

Descripción

Método que permite obtener orden por ID.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Bearer token	Sí	Es el token entregado por el servicio de Autenticación

Parameters

Nombre	Tipo	Requerido	Descripción
--------	------	-----------	-------------

id	number	Sí	Identificación de la orden
----	--------	----	----------------------------

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "id" : 111,
  "userId" : 1,
  "products" : [
    {
      "id" : 23,
      "count" : 2,
      "supplierId" : 32
    },
    {
      "id" : 23,
      "count" : 2,
      "supplierId" : 32
    }
  ],
  "paymentMethod" : 2,
  "total" : $400.000,
  "address" : "Calle 39C #XX - XX",
  "created" : "2013-07-02T21:36:25.344Z",
  "lastUpdated": "2013-07-02T21:36:25.344Z"
}
```

– Respuestas de error

404 - Not Found

Si no se encuentra la orden:

```
{
  "statusCode": "404",
  "message": "Order not found."
}
```

Get Orders By User	
Tipo de Servicio	REST
Método HTTP	GET
URI	/users/{userId}/orders

Descripción

Método que permite obtener lista de ordenes por usuario.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Bearer token	Sí	Es el token entregado por el servicio de Autenticación

Parameters

Nombre	Tipo	Requerido	Descripción
userId	number	Sí	Identificación del usuario
page	number	Sí	Número de pagina que se debe mostrar
limit	number	Sí	Límite de ordenes que se deben mostrar

Respuestas del Servicio

- Respuestas exitosas

200 - OK

```
{
```

```
[
  {
    "id" : 111,
    "userId" : 1,
    "productsCount" : 4,
    "total" : "$400.000"
  },
  {
    "id" : 111,
    "userId" : 1,
    "productsCount" : 4,
    "total" : "$400.000"
  }
]
```

204 – No content

Si no existen ordenes asociadas al usuario aún.

```
[]
```

– Respuestas de error

401 – Unauthorized

Si el token ha sido expirado o el usuario no está logueado.

```
{
  "statusCode": "401",
  "message": "Bad or expired token."
}
```

403 – Forbidden

Si el usuario logueado no debe tener acceso a crear la orden.

```
{
  "statusCode": "403",
  "message": "Do not have permission to this action."
}
```

404 - Not Found

Si el usuario no existe:

```
{  
  
    "statusCode": "404",  
  
    "message": "User not found."  
  
}
```

Update Order	
Tipo de Servicio	REST
Método HTTP	PUT
URI	/orders/{id}

Descripción

Método que permite actualizar una orden.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Bearer token	Sí	Es el token entregado por el servicio de Autenticación

Parameters

Nombre	Tipo	Requerido	Descripción
id	number	Sí	Identificación de la orden

Body

Nombre	Tipo	Requerido	Descripción
products	Object { "Id" : number, "count": number, "supplierId": number	No	Productos escogidos por el usuario. Se guarda el Id del producto, la cantidad (debe ser mayor a 0) y el

	}		Id del usuario que lo vende.
paymentMethod	number	No	Método de pago escogido por el usuario (Contado(1), tarjeta(2), transferencia(3))
address	string	No	Dirección a la que se va a hacer el envío.

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  [
    {
      "id" : 111,
      "userId" : 1,
      "productsCount" : 4,
      "total" : "$400.000"
    },
    {
      "id" : 111,
      "userId" : 1,
      "productsCount" : 4,
      "total" : "$400.000"
    }
  ]
}
```

– Respuestas de error

400 - Bad Request

Si todos los parámetros del cuerpo se envían nulos o vacíos:

```
{
  "statusCode": "400",
  "message": "Nothing has been updated because no data has been provided."
}
```

401 – Unauthorized

Si el token ha sido expirado o el usuario no está logueado.

```
{
  "statusCode": "401",
  "message": "Bad or expired token."
}
```

403 – Forbidden

Si el usuario logueado no debe tener acceso a crear la orden.

```
{
  "statusCode": "403",
  "message": "Do not have permission to this action."
}
```

404 - Not Found

Si no se encuentra la orden:

```
{
  "statusCode": "404",
  "message": "Order not found."
}
```

Delete Order	
Tipo de Servicio	REST
Método HTTP	DELETE
URI	/orders/{id}

Descripción

Método que permite eliminar una orden.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Bearer token	Sí	Es el token entregado por el servicio de Autenticación

Parameters

Nombre	Tipo	Requerido	Descripción
id	number	Sí	Identificación de la orden

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "statusCode": "200",
  "message": "Order deleted successfully."
}
```

– Respuestas de error

400 - Bad Request

Si el parametro del ID se envía nulo o vacío:

```
{
  "statusCode": "400",
  "message": "The order ID must be provided."
}
```

401 – Unauthorized

Si el token ha sido expirado o el usuario no está logueado.

```
{
  "statusCode": "401",
  "message": "Bad or expired token."
}
```

403 – Forbidden

Si el usuario logueado no debe tener permiso a eliminar la orden.

```
{
```

```
"statusCode": "403",
"message": "Do not have permission to this action."
}
```

404 - Not Found

Si no se encuentra la orden:

```
{
  "statusCode": "404",
  "message": "Order not found."
}
```

Products API

Get Products	
Tipo de Servicio	REST
Método HTTP	GET
URI	/products

Descripción

Método para obtener todos los productos.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Beare token	Si	Es el token entregado por el servicio de Autenticacion

Parameters

Nombre	Tipo	Requerido	Descripción
name	string	No	Nombre del producto
page	number	Sí	Numero de página que debe mostrarse

limit	number	Sí	Límite de productos a mostrar
-------	--------	----	-------------------------------

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "data": [
    {
      "id": 1,
      "name": "Zapatos Deportivos",
      "description": "Cómodos y resistentes",
      "price": 150000.00,
      "stock": 50,
      "images": ["https://example.com/img1.jpg"],
    },
    // More Products...
  ],
}
```

– Respuestas de error

400 - Bad Request

Si la url tiene parámetros erróneos:

```
{
  "statusCode" : "400",
  "message" : "Invalid Parameters"
}
```

401 - Unauthorized

Si el token expira o falta:

```
{
  "statusCode" : "401",
  "message" : "Bad or expired token."
}
```

403 - Forbidden

Si el usuario no tiene permisos suficientes:

```
{
  "statusCode": "403",
  "message": "Do not have permission to this action."
}
```

404 - Not Found

Si en un filtro o búsqueda específica no encuentra nada

```
{
  "statusCode": "404",
  "message": "Product not found."
}
```

Get Product	
Tipo de Servicio	REST
Método HTTP	GET
URI	/products/{id}

Descripción

Método que permite obtener información del producto asociado al ID.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Beare token	Si	Es el token entregado por el servicio de Autenticación

Parameters

Nombre	Tipo	Requerido	Descripción
id	number	Sí	Identificación del producto

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "data": [
    {
      "id": 1,
      "name": "Zapatos Deportivos",
      "description": "Cómodos y resistentes",
      "price": 150000.00,
      "stock": 50,
      "images": ["https://example.com/img1.jpg"],
    },
    // More products...
  ],
}
```

– Respuestas de error

400 - Bad Request

Si la url tiene parámetros erróneos:

```
{
```

```
"statusCode" : "400",  
"message" : "Invalid Parameters"  
}
```

401 - Unauthorized

Si el token expira o falta:

```
{  
  "statusCode" : "401",  
  "message" : "Bad or expired token."  
}
```

403 - Forbidden

Si el usuario no tiene permisos suficientes:

```
{  
  "statusCode": "403",  
  "message": "Do not have permission to this action."  
}
```

404 - Not Found

Si el producto no existe:

```
{  
  "statusCode": "404",  
  "message": "Product not found."  
}
```

Get Products By User	
Tipo de Servicio	REST
Método HTTP	GET

URI	/users/{userId}/products
-----	--------------------------

Descripción

Método que permite obtener productos asociados a un usuario.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Beare token	Sí	Es el token entregado por el servicio de Autenticación

Parameters

Nombre	Tipo	Requerido	Descripción
userId	number	Sí	Identificación del usuario
page	number	Sí	Pagina de los productos a mostrarse
limit	number	sí	Limite de productos a mostrar

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "data": [
    {
      "id": 1,
      "name": "Zapatos Deportivos",
      "description": "Cómodos y resistentes",
      "price": 150000.00,
      "stock": 50,
      "images": ["https://example.com/img1.jpg"],
    },
    // More products...
```

```
],  
}
```

204 - No Content

Si no existen productos asociados al usuario aún.

```
[]
```

– Respuestas de error

400 - Bad Request

Si la url tiene parámetros erróneos:

```
{  
  
  "statusCode" : "400",  
  
  "message" : "Invalid Parameters"  
  
}
```

401 - Unauthorized

Si el token expira o falta:

```
{  
  
  "statusCode" : "401",  
  
  "message" : "Bad or expired token."  
  
}
```

403 - Forbidden

Si el usuario no tiene permisos suficientes:

```
{  
  
  "statusCode": "403",  
  
  "message": "Do not have permission to this action."  
  
}
```

404 - Not Found

Si el usuario no existe:

```
{  
  
  "statusCode": "404",  
  
  "message": "User not found."  
  
}
```

Create Product	
Tipo de Servicio	REST
Método HTTP	POST
URI	/products

Descripción

Método que permite crear un producto.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Bearer token	Si	Es el token entregado por el servicio de Autenticación

Body

Nombre	Tipo	Requerido	Descripción
name	string	Sí	Nombre del producto
description	string	No	Descripción del producto
price	float	Si	Precio del producto

stock	number	No	Cantidad existente del producto
images	string	No	Imagen del producto
sellerId	number	Sí	ID del usuario/admin que crea el producto

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "statusCode": "200",
  "message": "Product created successfully."
}
```

– Respuestas de error

400 - Bad Request

Si alguno de los parametros requeridos son nulos o vacíos:

```
{
  "statusCode" : "400",
  "message" : "The following parameters are required: seller ID, name and price"
}
```

401 - Unauthorized

Si el token expira o falta:

```
{
  "statusCode" : "401",
  "message" : "Bad or expired token."
}
```

403 - Forbidden

Si el usuario no tiene permisos suficientes:

```
{  
  
  "statusCode": "403",  
  
  "message": "Do not have permission to this action."  
  
}
```

Update Product	
Tipo de Servicio	REST
Método HTTP	PUT
URI	/products/ {id}

Descripción

Método que permite actualizar productos

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Bearer token	Si	Es el token entregado por el servicio de Autenticación

Parameters

Nombre	Tipo	Requerido	Descripción
id	number	Sí	Identificación del Producto

Body

Nombre	Tipo	Requerido	Descripción
name	string	Sí	Nombre del producto
description	string	No	Descripción del producto

price	float	Si	Precio del producto
stock	number	No	Cantidad existente del producto
images	string	No	Imagen del producto

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "name" : "Zapatos puma",
  "description" : "Mejores",
  "price" : 1500000.00,
  "stock" : 3,
  "images" : "imagen"
}
```

– Respuestas de error

400 - Bad Request

Si alguno de los parámetros requeridos son nulos o vacíos:

```
{
  "statusCode" : "400",
  "message" : "The following parameters are required: name and price"
}
```

401 - Unauthorized

Si el token expira o falta:

```
{
  "statusCode" : "401",
  "message" : "Bad or expired token."
}
```

403 - Forbidden

Si el usuario no tiene permisos suficientes:

```
{  
  "statusCode": "403",  
  "message": "Do not have permission to this action."  
}
```

404 - Not Found

Si el producto no existe:

```
{  
  "statusCode": "404",  
  "message": "Product not found."  
}
```

Delete Product	
Tipo de Servicio	REST
Método HTTP	DELETE
URI	/products/ {id}

Descripción

Método que permite eliminar productos.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Beare token	Si	Es el token entregado por el servicio de Autenticación

Parameters

Nombre	Tipo	Requerido	Descripción
id	number	Sí	Identificación del Producto

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{  
  "statusCode": "200",  
  "message": "Product deleted successfully."  
}
```

– Respuestas de error

400 - Bad Request

Si alguno de los parámetros requeridos es nulo o vacío:

```
{  
  "statusCode" : "400",  
  "message" : "The Product Id parameter is missing"  
}
```

401 - Unauthorized

Si el token expira o falta:

```
{  
  "statusCode" : "401",  
  "message" : "Bad or expired token."  
}
```


403 - Forbidden

Si el usuario no tiene permisos suficientes:

```
{  
  
  "statusCode": "403",  
  
  "message": "Do not have permission to this action."  
  
}
```

404 - Not Found

Si el producto no existe:

```
{  
  
  "statusCode": "404",  
  
  "message": "Product not found."  
  
}
```

Categories API

Categories	
Tipo de Servicio	REST
Método HTTP	GET
URI	/categories

Descripción

Método para obtener todas las categorías de los productos.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Bearer token	Si	Es el token entregado por el servicio de Autenticación

Respuestas del Servicio

– **Respuestas exitosas**

200 - OK

```
{
  "data": [
    {
      "id": 1,
      "name": "Calzado"
    },
    // More categories...
  ],
}
```

– **Respuestas de error**

401 - Unauthorized

Si el token expira o falta:

```
{
  "statusCode" : "401",
  "message" : "Bad or expired token."
}
```

403 - Forbidden

Si el usuario no tiene permisos suficientes:

```
{
  "statusCode": "403",
  "message": "Do not have permission to this action."
}
```

Get Category

Tipo de Servicio	REST
Método HTTP	GET
URI	/categories/ {id}

Descripción

Método que permite obtener información de la categoría asociada al ID.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Bearer token	Si	Es el token entregado por el servicio de Autenticación

Parameters

Nombre	Tipo	Requerido	Descripción
id	number	Sí	Identificación de la categoría

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "data": {
    "id": 1,
    "name": "Calzado"
  }
}
```

– Respuestas de error

400 - Bad Request

Si alguno de los parámetros requeridos es nulo o vacío:

```
{
```

```
"statusCode" : "400",  
"message" : "The Category Id parameter is missing"  
}
```

401 - Unauthorized

Si el token expira o falta:

```
{  
  "statusCode" : "401",  
  "message" : "Bad or expired token."  
}
```

404 - Not Found

Si la categoría no existe:

```
{  
  "statusCode": "404",  
  "message": "Category not found."  
}
```

Create Category	
Tipo de Servicio	REST
Método HTTP	POST
URI	/categories

Descripción

Método que permite crear categorías.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Beare token	Si	Es el token entregado por el servicio de Autenticación

Body

Nombre	Tipo	Requerido	Descripción
name	string	Sí	Nombre de la categoría
description	string	No	Descripción de la categoría

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "statusCode": "200",
  "message": "Category created successfully."
}
```

– Respuestas de error

400 - Bad Request

Si el nombre se envía nulo o vacío:

```
{
  "statusCode": "400",
  "message": "The name must be provided."
}
```

401 – Unauthorized

Si el token ha sido expirado o el usuario no está logueado.

```
{
    "statusCode": "401",
    "message": "Bad or expired token."
}
```

403 – Forbidden

Si el usuario logueado no tiene permisos.

```
{
    "statusCode": "403",
    "message": "Do not have permission to this action."
}
```

Update Caregory	
Tipo de Servicio	REST
Método HTTP	PUT
URI	/categories/ {id}

Descripción

Método que permite actualizar categoría.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Bearer token	Si	Es el token entregado por el servicio de Autenticación

Parameters

Nombre	Tipo	Requerido	Descripción
id	number	Sí	Identificación de la categoría

Body

Nombre	Tipo	Requerido	Descripción
name	string	Sí	Nombre de la categoría
description	string	No	Descripción de la categoría

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{  
  "name" : "Zapatos",  
  "description" : "Zapatos diferentes"  
}
```

– Respuestas de error

400 - Bad Request

Si alguno de los parámetros requeridos es nulo o vacío:

```
{  
  "statusCode" : "400",  
  "message" : "The following parameters are required: category Id and name"  
}
```

401 - Unauthorized

Si el token expira o falta:

```
{  
  "statusCode" : "401",  
  "message" : "Bad or expired token."  
}
```

403 – Forbidden

Si el usuario logueado no tiene permisos.

```
{
  "statusCode": "403",
  "message": "Do not have permission to this action."
}
```

404 - Not Found

Si la categoría no existe:

```
{
  "statusCode": "404",
  "message": "Category not found."
}
```

Categories	
Tipo de Servicio	REST
Método HTTP	DELETE
URI	/categories/ {id}

Descripción

Método que permite eliminar productos

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Beare token	Sí	Es el token entregado por el servicio de Autenticación

Parameters

Nombre	Tipo	Requerido	Descripción
id	number	Sí	Identificación de la categoría

Respuestas del Servicio

– **Respuestas exitosas**

200 - OK

```
{  
    "statusCode": "200",  
    "message": "Category deleted successfully."  
}
```

– **Respuestas de error**

400 - Bad Request

Si alguno de los parámetros requeridos es nulo o vacío:

```
{  
    "statusCode" : "400",  
    "message" : "The following parameters are required: category Id and name"  
}
```

401 - Unauthorized

Si el token expira o falta:

```
{  
    "statusCode" : "401",  
    "message" : "Bad or expired token."  
}
```

403 – Forbidden

Si el usuario logueado no tiene permisos.

```
{  
    "statusCode": "403",  
    "message": "Do not have permission to this action."  
}
```

404 - Not Found

Si la categoría no existe:

```
{  
    "statusCode": "404",  
    "message": "Category not found."  
}
```

Shipments API

Get Shipments	
Tipo de Servicio	REST
Método HTTP	GET
URI	/shipments

Descripción

Método para obtener todos los envíos.

Headers

Nombre	Valor	Requerido	Descripcion
Authorization	Bearer token	Si	Es el token entregado por el servicio de Autenticación

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "data": [
    {
      "id": 1,
      "orderId": 101,
      "trackingNumber": "TRK-20250824-00001",
      "status": "EN_TRANSITO",
      "carrier": "Transportadora XYZ",
      "createdAt": "2025-08-24T12:00:00Z",
      "updatedAt": "2025-08-24T12:30:00Z"
    },
    // More shipments...
  ],
}
```

– Respuestas de error

401 - Unauthorized

Si el token expira o falta:

```
{
  "statusCode" : "401",
```

```
    "message" : "Bad or expired token."
}
```

403 – Forbidden

Si el usuario logueado no tiene permisos.

```
{
    "statusCode": "403",
    "message": "Do not have permission to this action."
}
```

Get Shipment	
Tipo de Servicio	REST
Método HTTP	GET
URI	/Shipments/{trackingNumber}

Descripción

Método que permite obtener información de los envíos asociados al Numero de guía.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Beare token	Si	Es el token entregado por el servicio de Autenticacion

Parameters

Nombre	Tipo	Requerido	Descripción
--------	------	-----------	-------------

trackingNumber	number	Sí	Número de seguimiento (formato TRK-YYYYMMDD-XXXXX).
----------------	--------	----	---

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "data": {
    "id": 1,
    "orderId": 101,
    "trackingNumber": "TRK-20250824-00001",
    "status": "EN_TRANSITO",
    "carrier": "Transportadora XYZ",
    "createdAt": "2025-08-24T12:00:00Z",
    "updatedAt": "2025-08-24T12:30:00Z",
    "history": [
      {
        "status": "PENDIENTE",
        "timestamp": "2025-08-24T12:00:00Z"
      },
      {
        "status": "EN_TRANSITO",
        "timestamp": "2025-08-24T12:30:00Z"
      }
    ]
  }
}
```

– Respuestas de error

400 - Bad Request

Si alguno de los parámetros requeridos es nulo o vacío:

```
{  
    "statusCode" : "400",  
    "message" : "The trackingNumber parameter is required."  
}
```

401 - Unauthorized

Si el token expira o falta:

```
{  
    "statusCode" : "401",  
    "message" : "Bad or expired token."  
}
```

403 – Forbidden

Si el usuario logueado no tiene permisos.

```
{  
    "statusCode": "403",  
    "message": "Do not have permission to this action."  
}
```

404 - Not Found

Si el envío no existe:

```
{  
    "statusCode": "404",  
    "message": "Shipment not found."  
}
```

Create Shipment	
Tipo de Servicio	REST
Método HTTP	POST
URI	/shipments

Descripción

Método que permite crear Envíos.

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Beare token	Si	Es el token entregado por el servicio de Autenticacion

Body

Nombre	Tipo	Requerido	Descripción
orderId	number	Si	ID de la orden asociada
trackingNumber	string	Sí	Número de seguimiento (formato TRK-YYYYMMDD-XXXXX)
carrier	string	No	Nombre de la transportadora
status	string	No	Estado inicial (default: PENDIENTE).

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "id": "1",
```

```
    "orderId": "101",
    "trackingNumber": "TRK-20250824-00001",
    "carrier": "Transportadora XYZ",
    "status": "PENDIENTE",
    "createdAt": "2025-08-24T12:00:00Z"
}
```

– Respuestas de error

400 - Bad Request

Si alguno de los parámetros requeridos es nulo o vacío:

```
{
    "statusCode" : "400",
    "message" : "The following parameters are required: trackingNumber and
order ID."
}
```

401 - Unauthorized

Si el token expira o falta:

```
{
    "statusCode" : "401",
    "message" : "Bad or expired token."
}
```

403 – Forbidden

Si el usuario logueado no tiene permisos.

```
{
    "statusCode": "403",
    "message": "Do not have permission to this action."
}
```


}

Update Shipment	
Tipo de Servicio	REST
Método HTTP	PUT
URI	/shipments/{id}

Descripción

Método que permite actualizar Envíos

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Beare token	Si	Es el token entregado por el servicio de Autenticación

Parameters

Nombre	Tipo	Requerido	Descripción
id	number	Sí	Identificación del Envío

Body

Nombre	Tipo	Requerido	Descripción
orderId	number	Si	ID de la orden asociada
trackingNumber	string	Sí	Número de seguimiento (formato TRK-YYYYMMDD-XXXXX)
carrier	string	No	Nombre de la transportadora
status	string	No	Estado inicial (default: PENDIENTE).

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{
  "id": "1",
  "orderId": "101",
  "trackingNumber": "TRK-20250824-00001",
  "status": "EN_ENTREGA",
  "carrier": "Transportadora XYZ",
  "notes": "En vehículo de reparto",
  "updatedAt": "2025-08-24T12:45:00Z"
}
```

– Respuestas de error

400 - Bad Request

Si alguno de los parámetros requeridos es nulo o vacío:

```
{
  "statusCode" : "400",
  "message" : "The following parameters are required: trackingNumber, order ID
and shipment ID."
}
```

401 - Unauthorized

Si el token expira o falta:

```
{
  "statusCode" : "401",
  "message" : "Bad or expired token."
}
```

403 – Forbidden

Si el usuario logueado no tiene permisos.

```
{
```

```
"statusCode": "403",  
"message": "Do not have permission to this action."  
}
```

404 - Not Found

Si el envío no existe:

```
{  
  
  "statusCode": "404",  
  "message": "Shipment not found."  
}
```

Delete Shipment	
Tipo de Servicio	REST
Método HTTP	DELETE
URI	/shipments/{id}

Descripción

Método que permite eliminar productos

Headers

Nombre	Valor	Requerido	Descripción
Authorization	Beare token	Si	Es el token entregado por el servicio de Autenticación

Parameters

Nombre	Tipo	Requerido	Descripción
id	number	Sí	Identificación del Envío

Respuestas del Servicio

– Respuestas exitosas

200 - OK

```
{  
    "statusCode": "200",  
    "message": "Shipment deleted successfully."  
}
```

– Respuestas de error

400 - Bad Request

Si alguno de los parámetros requeridos es nulo o vacío:

```
{  
    "statusCode" : "400",  
    "message" : "The shipment ID parameter is required."  
}
```

401 - Unauthorized

Si el token expira o falta:

```
{  
    "statusCode" : "401",  
    "message" : "Bad or expired token."  
}
```

403 – Forbidden

Si el usuario logueado no tiene permisos.

```
{  
    "statusCode": "403",  
    "message": "Do not have permission to this action."  
}
```

404 - Not Found

Si el envío no existe:

```
{  
  
  "statusCode": "404",  
  
  "message": "Shipment not found."  
  
}
```

Notifications API

Create Notification	
Tipo de Servicio	REST
Método HTTP	POST
URI	/notifications

Descripción

Crea nueva notificación cuando se actualice el tracking o estado

Body

Nombre	Tipo	Requerido	Descripción
userId	number	Sí	ID del usuario
orderId	number	Si	ID de la orden
orderStatus	string	Si	Estado de la orden
createdAt	Date	Si	Marca temporal YYYY-MM-DDTHH:mm:ss.sssZ
message	String	Si	Contenido de la notificación

Respuesta

200 - Ok

```
{  
  
  "notificationId": "123"  
  
  "message" : "Notification sent successfully"  
  
}
```

Error

400 – Bad Request

```
{  
  "error": "400",  
  "message": "You've sent a bad request"  
}
```

401 – Unauthorized

```
{  
  "error": "401",  
  "message": " requires authentication"  
}
```

404 – Not Found

```
{  
  "error": "404",  
  "message": "User does not have order assigned"  
}
```

409 – Conflict

```
{  
  "error": "409",  
  "message": "Its not permitted to roll back states"  
}
```

500 – Internal Server Error

```
{  
  "error": "500",  
  "message": "The server encountered an error and could not complete your  
request"  
}
```

Get Notification	
Tipo de Servicio	REST
Método HTTP	GET
URI	/notifications/:id

Descripción

Muestra los detalles de una notificación en específico

Body

Nombre	Tipo	Requerido	Descripción
notificationId	number	Sí	ID de la notificación

Respuesta

200 - Ok

```
{
  "notificationId": 123,
  "userId": 001,
  "orderId": 001
  "orderStatus": "PENDIENTE",
  "CreatedAt": "YYYY-MM-DDTHH:mm:ss.sssZ"
  "message" : "Your order is pending"
}
```

Error

404 – Not Found

```
{
  "error": "404",
  "message": "The notification does not exist"
}
```

401 – Unauthorized

```
{
```

```
"error": "401",  
  "message": " requires authentication"  
}
```

500 – Internal Server Error

```
{  
  "error": "500",  
  "message": "The server encountered an error and could not complete your  
request"  
}
```

Get Notifications	
Tipo de Servicio	REST
Método HTTP	GET
URI	/notifications

Descripción

Muestra el historial de notificaciones de un usuario.

Body

Nombre	Tipo	Requerido	Descripción
userId	number	Sí	ID del usuario

Respuesta

200 – Ok

```
[  
  {  
    "notificationId": 123,  
    "userId": 001,  
    "orderId": 001,  
    "orderStatus": "PENDIENTE",  
    "CreatedAt": "YYYY-MM-DDTHH:mm:ss.sssZ"  
    "message": "Your order is pending"  
  }  
]
```



```
    },  
    {  
      "notificationId": 123,  
      "userId":001,  
      "orderId":001  
      "orderStatus": "PREPARANDO",  
      "CreatedAt":" YYYY-MM-DDTHH:mm:ss.sssZ"  
      "message" : "Your order is being prepared"  
    }  
  ]
```

Error

404 – Not Found

```
{  
  "error": "404",  
  "message": "No notifications found"  
}
```

401 – Unauthorized

```
{  
  "error": "401",  
  "message": " requires authentication"  
}
```

500 – Internal Server Error

```
{  
  "error": "500",  
  "message": "The server encountered an error and could not complete your request"  
}
```

