



Product Api

🕒 Created time	@July 19, 2023 12:16 AM
⋮ Tags	

API routes for Product :

Brand APIs

▼ Create Brand

POST /brands

Creates a new brand and links it to the vendor shop role of the authenticated user.

Request Body:

```
{
  "brand_name": "String"
}
```

Response (201):

```
{
  "message": "Brand created successfully",
  "brandId": "Number"
}
```

▼ Update Brand

Updates an existing brand associated with the vendor shop role of the authenticated user.

Request:

```
bashCopy code
PUT /brands
```

Request Body:

```
jsonCopy code
{
  "brand_id": "Number",
  "brand_name": "String"
}
```

Response (200):

```
jsonCopy code
{
  "message": "Brand updated successfully",
  "brandId": "Number"
}
```

Description:

- **brand_id**: The ID of the brand to be updated.
- **brand_name**: The new name for the brand.

Note:

- The **brand_id** must be associated with the vendor shop role of the authenticated user.
- If the update is successful, the response will contain a success message and the **brandId** of the updated brand.

▼ Update Brand

Updates an existing brand associated with the vendor shop role of the authenticated user.

Request:

```
bashCopy code
PUT /brands
```

Request Body:

```
jsonCopy code
{
  "brand_id": "Number",
  "brand_name": "String"
}
```

Response (200):

```
jsonCopy code
{
  "message": "Brand updated successfully",
  "brandId": "Number"
}
```

Description:

- **brand_id**: The ID of the brand to be updated.
- **brand_name**: The new name for the brand.

Note:

- The **brand_id** must be associated with the vendor shop role of the authenticated user.
- If the update is successful, the response will contain a success message and the **brandId** of the updated brand.

▼ Get Brand By ID

GET /brands/:id

Gets a brand by ID that is linked to the vendor shop role of the authenticated user.

Response (200):

```
{
  "id": "Number",
  "brand_name": "String",
  "added_by": "Number"
}
```

▼ Get All Brands

GET /brands

Gets all brands linked to the vendor shop role of the authenticated user.

Response (200):

```
[
  {
    "id": "Number",
    "brand_name": "String",
    "added_by": "Number"
  },
  ...
]
```

Unit APIs

▼ Create Unit

POST /units

Creates a new unit and links it to the vendor shop role of the authenticated user.

Request Body:

```
{
  "unit_name": "String"
}
```

Response (201):

```
{
  "message": "Unit created successfully",
  "unitId": "Number"
}
```

▼ Update Unit

PUT /units/:id

Updates a unit by ID that is linked to the vendor shop role of the authenticated user.

Request Body:

```
{
  "unit_name": "String"
}
```

```
}
```

Response (200):

```
{
  "message": "Unit updated successfully",
  "unitId": "Number"
}
```

▼ Delete Unit

Deletes an existing unit associated with the vendor shop role of the authenticated user.

Request:

```
bashCopy code
DELETE /units/:id
```

Response (200):

```
jsonCopy code
{
  "message": "Unit deleted successfully",
  "unitId": "Number"
}
```

Description:

- `id`: The ID of the unit to be deleted.

Note:

- The `id` must be associated with the vendor shop role of the authenticated user.
- If the delete is successful, the response will contain a success message and the `unitId` of the deleted unit.

▼ Get Units

GET /units

Gets all units linked to the vendor shop role of the authenticated user.

Response (200):

```
[
{
  "id": "Number",
  "unit_name": "String",
  "added_by": "Number"
},
...
]
```

▼ Get Unit By ID

GET /units/:id

Gets a unit by ID that is linked to the vendor shop role of the authenticated user.

Response (200):

```
{
  "id": "Number",
  "unit_name": "String",
  "added_by": "Number"
}
```

HSN APIs

▼ Create HSN

POST /hsn

Creates a new HSN code and links it to the vendor shop role of the authenticated user.

Request Body:

```
{
  "code": "String",
  "gst_rate": "Number",
  "cgst_rate": "Number",
  "sgst_rate": "Number",
  "igst_rate": "Number"
}
```

Response (201):

```
{
  "message": "HSN created successfully",
}
```

```
"hsnId": "Number"
}
```

▼ Update HSN

PUT /hsn/:id

Updates an HSN by ID that is linked to the vendor shop role of the authenticated user.

Request Body:

```
{
  "code": "String",
  "gst_rate": "Number",
  "cgst_rate": "Number",
  "sgst_rate": "Number",
  "igst_rate": "Number"
}
```

Response (200):

```
{
  "message": "HSN updated successfully",
  "hsnId": "Number"
}
```

▼ Delete HSN

Deletes an existing HSN associated with the vendor shop role of the authenticated user.

Request:

```
bashCopy code
DELETE /hsn/:id
```

Parameters:

- **id** (URL parameter): The ID of the HSN to be deleted.

Response (200):

```
jsonCopy code
{
  "message": "HSN deleted successfully",
  "hsnId": "Number"
}
```

Description:

- **id**: The ID of the HSN to be deleted.

Note:

- The HSN must be associated with the vendor shop role of the authenticated user.
- If the deletion is successful, the response will contain a success message and the **hsnId** of the deleted HSN.

▼ Get HSN By ID

GET /hsn/:id

Gets an HSN by ID that is linked to the vendor shop role of the authenticated user.

Response (200):

```
{
  "id": "Number",
  "code": "String",
  "gst_rate": "Number",
  "cgst_rate": "Number",
  "sgst_rate": "Number",
  "igst_rate": "Number",
  "added_by": "Number"
}
```

▼ Get All HSNs

GET /hsn

Gets all HSN codes linked to the vendor shop role of the authenticated user.

Response (200):


```
[
{
  "id": "Number",
  "code": "String",
  "gst_rate": "Number",
  "cgst_rate": "Number",
  "sgst_rate": "Number",
  "igst_rate": "Number",
  "added_by": "Number"
},
...
]
```

Product APIs

▼ Create Product

POST /products

Creates a new product and links it to the vendor shop role of the authenticated user.

Request Body:

```
{
  "product_name": "String",
  "brand_id": "Number",
  "hsn_id": "Number"
}
```

Response (201):

```
{
  "message": "Product created successfully",
  "productId": "Number"
}
```

▼ Get Products

GET /products

Gets all products linked to the vendor shop role of the authenticated user.

Response (200):

```
[
{
  "id": "Number",
  "product_name": "String",
  "brand_id": "Number",
  "hsn_id": "Number",
  "added_by": "Number"
},
...
]
```

▼ Get Product By ID

GET /products/:id

Gets a product by ID that is linked to the vendor shop role of the authenticated user.

Response (200):

```
{
  "id": "Number",
  "product_name": "String",
  "brand_id": "Number",
  "hsn_id": "Number",
  "added_by": "Number"
}
```

▼ Update Product

PUT /products/:id

Updates a product by ID that is linked to the vendor shop role of the authenticated user.

Request Body:

```
{
  "product_name": "String",
  "brand_id": "Number",
  "hsn_id": "Number"
}
```

Response (200):

```
{  
  "message": "Product updated successfully",  
  "productId": "Number"  
}
```

▼ Delete Product

DELETE /products/:id

Deletes a product by ID that is linked to the vendor shop role of the authenticated user.

Response (200):

```
{  
  "message": "Product deleted successfully",  
  "productId": "Number"  
}
```

Variant APIs

▼ Create Variant

POST /variants

Creates a new product variant and links it to the vendor shop role of the authenticated user.

Request Body:

```
{  
  "product_id": "Number",  
  "unit_id": "Number",  
  "variant_name": "String",  
  "price": "Number",  
  "quantity": "Number",  
  "order_quantity": "Number",  
  "mrp": "Number"  
}
```

Response (201):

```
{  
  "message": "Variant created successfully",  
  "variantId": "Number"  
}
```

```
}
```

▼ Get Variants

GET /variants

Gets all variants linked to the vendor shop role of the authenticated user.

Response (200):

```
[  
  {  
    "id": "Number",  
    "product_id": "Number",  
    "unit_id": "Number",  
  
    "variant_name": "String",  
    "price": "Number",  
    "quantity": "Number",  
    "order_quantity": "Number",  
    "mrp": "Number",  
    "added_by": "Number"  
  },  
  ...  
]
```

▼ Get Variant By ID

GET /variants/:id

Gets a variant by ID that is linked to the vendor shop role of the authenticated user.

Response (200):

```
{  
  "id": "Number",  
  "product_id": "Number",  
  "unit_id": "Number",  
  "variant_name": "String",  
  "price": "Number",  
  "quantity": "Number",  
  "order_quantity": "Number",  
  "mrp": "Number",  
  "added_by": "Number"  
}
```

```
}
```

▼ Update Variant

PUT /variants/:id

Updates a variant by ID that is linked to the vendor shop role of the authenticated user.

Request Body:

```
{
  "product_id": "Number",
  "unit_id": "Number",
  "variant_name": "String",
  "price": "Number",
  "quantity": "Number",
  "order_quantity": "Number",
  "mrp": "Number"
}
```

Response (200):

```
{
  "message": "Variant updated successfully",
  "variantId": "Number"
}
```

▼ Delete Variant

DELETE /variants/:id

Deletes a variant by ID that is linked to the vendor shop role of the authenticated user.

Response (200):

```
{
  "message": "Variant deleted successfully",
  "variantId": "Number"
}
```

Order APIs

▼ Create Order

POST /orders

Creates a new order for the given customer and links it to the vendor shop of the authenticated user.

Request Body:

```
{
  "customer_id": "Number",
  "order_items": [
    {
      "product_name": "String",
      "variant_price": "Number",
      "gst_price": "Number",
      "selling_price": "Number",
      "max_retail_price": "Number",
      "quantity": "Number"
    },
    ...
  ]
}
```

Response (201):

```
{
  "message": "Order created successfully",
  "orderId": "Number",
  "transactionId": "Number"
}
```

▼ Get Order By ID

GET /orders/:id

Gets an order by ID, along with order items and customer name. Accessible only if the order belongs to the authenticated user's vendor shop or customer.

Response (200):

```
{
  "order": {
    "id": "Number",
    "order_status": "String",
    "added_by": "Number",
    "customer_shops": "Number"
  },
}
```

```

"customer_name": "String",
"items": [
{
  "id": "Number",
  "order_id": "Number",
  "product_name": "String",
  ...
},
...
]
}

```

▼ Delete Order

DELETE /orders/:id

Deletes an order by ID. Only possible if the order belongs to the authenticated user's vendor shop.

Response (200):

```

{
  "message": "Order deleted successfully"
}

```

Price Allowance APIs

▼ Create Allowance

POST /allowance

Creates a new price allowance percentage for the vendor shop of the authenticated user.

Request Body:

```

{
  "allowance": "Number"
}

```

Response (201):

```

{
  "message": "Price allowance created successfully",
  "allowanceId": "Number"
}

```

```
}
```

▼ Update Allowance

PUT /allowance/:id

Updates the price allowance percentage for the vendor shop of the authenticated user.

Request Body:

```
{  
  "allowance": "Number"  
}
```

Response (200):

```
{  
  "message": "Price allowance updated successfully"  
}
```

▼ Get Allowance

GET /allowance/:id

Gets the price allowance for the vendor shop of the authenticated user.

Response (200):

```
{  
  "id": "Number",  
  "allowance": "Number",  
  "added_by": "Number"  
}
```

▼ Delete Allowance

DELETE /allowance/:id

Deletes the price allowance for the vendor shop of the authenticated user.

Response (200):

```
{  
  "message": "Price allowance deleted successfully"  
}
```

Route: /registerAdminUser

▼ Get All Transactions by User

Fetches all transactions associated with the authenticated user's vendor shop role.

Request:

```
bashCopy code
GET /transactions
```

Response (200):

```
jsonCopy code
{
  "message": "Transactions fetched successfully",
  "transactions": [
    {
      "id": 1,
      "order_id": 123,
      "amount": 250.00,
      "created_at": "2023-07-19T12:34:56Z"
    },
    {
      "id": 2,
      "order_id": 124,
      "amount": 150.00,
      "created_at": "2023-07-19T13:45:22Z"
    },
    // More transaction objects...
  ]
}
```

Description:

- The `GET /transactions` endpoint is used to fetch all transactions associated with the authenticated user's vendor shop role.
- The `transactions` array in the response contains objects representing individual transactions, with each transaction having the following properties:
 - `id`: The unique identifier of the transaction.
 - `order_id`: The ID of the order to which the transaction is linked.
 - `amount`: The total amount of the transaction.
 - `created_at`: The timestamp indicating when the transaction was created.

- The authenticated user's vendor shop role is determined by their user ID.
- The `vendor_shop_role_id` is used to fetch all transactions linked to orders created by the user.
- If no transactions are found for the user, the response will contain a 404 status code with the message "No Transactions Found."

Note:

- The user must be authenticated to access this endpoint.
- The `vendor_shop_role_id` is used to filter the transactions to only those associated with the user's vendor shop role.