API DOCUMENTATION

This Documentation includes endpoints with filter, search and pagination, along with explanations of their parameters and response formats. It also retains the authentication section for JWT tokens and serves as a guide for developers to integrate with the API. Also enabled In Memory Caching to reduce the load.

Authentication

To access the API, you need to obtain a JSON Web Token (JWT). Include this token in the request headers as follows:

Authorization: Bearer < token>

Contents

1. Register

To Create a new user

Request

Method	URL
POST	/api/User/Register

Body

{
"email": "string",
"username": "string",
"password": "string",
"role": "string"
}

Response

Status	Response

200	{
	"statusCode": 200,
	"message": "Success",
	"data": null
	}

409	{
	"statusCode": 409,
	"message": "Conflict",
	"data": null
	}

400	{
	"statusCode": 400,
	"message": "Exception",
	"data": null
	}

2. Authenticate

Authenticate the user with the system and obtain the auth_token
Implemented authentication and authorization mechanism using JWT Token

Request

Method	URL
POST	api/User/Authenticate

Body

{
  "userName": "string",
  "password": "string"
}

Response

Status	Response

200	{
	  "statusCode": 200,
	  "message": "",
	  "data": {
	    "userName": "",
	    "token": ""
	  }
	}

404	{
	  "statusCode": 404,
	  "message": "UserName Not Found",
	  "data": null
	}

400	{
	  "statusCode": 400,
	  "message": "Exception",
	  "data": null
	}

3. GetAllProducts

Get all products.

Implemented caching for frequently accessed data to reduce the load.
Search history records will store in database including the search query, user ID, timestamp

Request

Method	URL
GET	/api/Product/GetAllProducts
Bearer	The token that was given in response to Authenticate

Parameters			
Parameter	**Type**	**Default Value**	**Description**
searchTerm	string		The search query
sortBy	String	Name Or Price	The parameter to sort by
page	Integer	1	The page number
pageSize	Integer	10	The number of results per page

Response

Status	Response

200	"statusCode": 200,
	"message": "",
	"data": {
	"totalCount": 2,
	"totalPages": 1,
	"currentPage": 1,
	"pageSize": 10,
	"data": []
	}

404	{ "statusCode": 404, "message": "Search Item Not Found", "data": null

	}

400	{
	"statusCode": 400,
	"message": "Exception",
	"data": null
	}

401	Unauthorized

4. Save Product

To add new product

Request

Method	URL
GET	/api/Product/SaveProduct
Bearer	The token that was given in response to Authenticate

Body

{
"name": "string",
"description": "string",
"price": 0,
"quantity": 0,
"brand": "string"
}

Response

Status	Response

200	{
	"statusCode": 200,
	"message": "Success",
	"data": null
	}

400	{
	"statusCode": 400,
	"message": "Exception",
	"data": null
	}

401	Unauthorized

5. DeleteProduct

To delete product (change isActive status to false)

Request

Method	URL
GET	api/Product/DeleteProduct
Bearer	The token that was given in response to Authenticate

Body

{
Id:integer
}

Response

Status	Response

200	{
	"statusCode": 200,
	"message": "Success",
	"data": null
	}

400	{
	"statusCode": 400,
	"message": "Exception",
	"data": null
	}

401	Unauthorized

6. Product Search History By User

Get all Product Search History By User

Request

Method	URL
GET	/api/Product/GetProductSearchHistoryByUser
Bearer	The token that was given in response to Authenticate

Parameters		
Parameter	Type	Default Value
userId	Integer	

Response

Status	Response
200	{ "statusCode": 200, "message": "", "data": [] }

404	{
	"statusCode": 404,
	"message": "Search Item Not Found",
	"data": null
	}

401	Unauthorized