



/v3/api-docs

Explore

# E-Commerce API

[/v3/api-docs](#)

API documentation for the E-Commerce application

## Servers

[http://localhost:7070 - Generated server url](http://localhost:7070)[Authorize](#)

## Authentication

User registration, login, email verification, and password management

**POST** /auth/reset-password Reset user password

Resets the user's password using a valid reset token

### Parameters

[Try it out](#)

No parameters

### Request body required

[application/json](#)

### Example Value Schema

```
{  
  "newPassword": "NewPassword123",  
  "token": "eyJhbGciOiJIUzI1NiJ9.reset.token"  
}
```

### Responses

Code	Description	Links
200	<p><b>Password reset successful</b></p> <p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">/*</div> <p>Controls Accept header.</p> <p><b>Example Value</b> Schema</p> <pre>{   "email": "john.doe@example.com",   "passwordChangedSuccess": true,   "statusMessage": "Password changed successfully",   "enabled": true,   "verified": true }</pre>	No links
401	<p><b>Token expired</b></p> <p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">/*</div> <p><b>Example Value</b> Schema</p> <pre>{   "errorStatus": "UNAUTHORIZED",   "errorMessage": "Invalid username or password",   "errorDescription": "uri=/auth/login",   "errorTimestamp": "2025-01-01T12:00:00" }</pre>	No links
404	<p><b>Token or user not found</b></p> <p>Media type</p> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;">/*</div> <p><b>Example Value</b> Schema</p> <pre>{   "errorStatus": "UNAUTHORIZED",   "errorMessage": "Invalid username or password",   "errorDescription": "uri=/auth/login",   "errorTimestamp": "2025-01-01T12:00:00" }</pre>	No links

POST /auth/register Register a new user ^

Creates a new user account and sends a verification email

Parameters

Try it out

No parameters

**Request body** required**application/json****Example Value** Schema

```
{
  "username": "john_doe",
  "password": "Password123",
  "firstName": "John",
  "lastName": "Doe",
  "email": "john.doe@example.com"
}
```

**Responses**

Code	Description	Links
------	-------------	-------

201	User registered successfully	<i>No links</i>
-----	------------------------------	-----------------

Media type

**\*/\***

Controls Accept header.

**Example Value** Schema

```
{
  "id": 1,
  "addresses": [
    {
      "id": 0,
      "addressLine1": "string",
      "addressLine2": "string",
      "city": "string",
      "country": "string"
    }
  ],
  "user_name": "john_doe",
  "email": "john.doe@example.com",
  "first_name": "John",
  "last_name": "Doe",
  "is_enabled": true,
  "is_verified": true
}
```

409	User already exists	<i>No links</i>
-----	---------------------	-----------------

Media type

**\*/\*****Example Value** Schema

Code	Description	Links
	<pre>{     "errorStatus": "UNAUTHORIZED",     "errorMessage": "Invalid username or password",     "errorDescription": "uri=/auth/login",     "errorTimestamp": "2025-01-01T12:00:00" }</pre>	
500	Email service failure	No links
	<p>Media type</p> <div style="border: 1px solid black; padding: 5px; width: fit-content;">/*</div>	
	<p>Example Value Schema</p> <pre>{     "errorStatus": "UNAUTHORIZED",     "errorMessage": "Invalid username or password",     "errorDescription": "uri=/auth/login",     "errorTimestamp": "2025-01-01T12:00:00" }</pre>	
POST	/auth/login User login	^
	Authenticates user credentials and returns a JWT access token	
	<p>Parameters</p>	<div style="border: 1px solid black; padding: 5px; width: fit-content;">Try it out</div>
	No parameters	
	<p>Request body <span style="color: red;">required</span></p>	<div style="border: 1px solid black; padding: 5px; width: fit-content;">application/json</div>
	<p>Example Value Schema</p> <pre>{     "username": "john_doe",     "password": "Password123" }</pre>	
	<p>Responses</p>	

Code	Description	Links
200	Login successful	No links
	<p>Media type</p> <div style="border: 2px solid #ccc; padding: 2px; display: inline-block;">/*</div> <p>Controls Accept header.</p> <p><b>Example Value</b> Schema</p> <pre>{   "jwt": "eyJhbGciOiJIUzI1NiJ9eyJzdWIiOiJ1c2VyIn0.signature",   "success": true,   "failureMessage": "Invalid username or password" }</pre>	
401	Invalid credentials	No links
	<p>Media type</p> <div style="border: 2px solid #ccc; padding: 2px; display: inline-block;">/*</div> <p><b>Example Value</b> Schema</p> <pre>{   "errorStatus": "UNAUTHORIZED",   "errorMessage": "Invalid username or password",   "errorDescription": "uri=/auth/login",   "errorTimestamp": "2025-01-01T12:00:00" }</pre>	
403	User email not verified	No links
	<p>Media type</p> <div style="border: 2px solid #ccc; padding: 2px; display: inline-block;">/*</div> <p><b>Example Value</b> Schema</p> <pre>{   "errorStatus": "UNAUTHORIZED",   "errorMessage": "Invalid username or password",   "errorDescription": "uri=/auth/login",   "errorTimestamp": "2025-01-01T12:00:00" }</pre>	
404	User not found	No links
	<p>Media type</p> <div style="border: 2px solid #ccc; padding: 2px; display: inline-block;">/*</div> <p><b>Example Value</b> Schema</p> <pre>{   "errorStatus": "UNAUTHORIZED",   "errorMessage": "Invalid username or password",   "errorDescription": "uri=/auth/login",   "errorTimestamp": "2025-01-01T12:00:00" }</pre>	

Code	Description	Links
	<pre>        "errorTimestamp": "2025-01-01T12:00:00"     }  POST /auth/forgot-password Initiate password reset</pre>	<a href="#">^</a>
	Sends password reset instructions to the user's email	
<b>Parameters</b>		<a href="#">Try it out</a>
No parameters		
<b>Request body</b> <small>required</small>		<a href="#">application/json</a>
<b>Example Value</b> Schema		
	<pre>{     "email": "john.doe@example.com",     "username": "john_doe" }</pre>	
<b>Responses</b>		
Code	Description	Links
200	Reset email sent	<a href="#">No links</a>
	Media type	
	<a href="#">/*</a>	
	Controls Accept header.	
	<b>Example Value</b> Schema	
	<code>string</code>	
403	User email not verified	<a href="#">No links</a>
	Media type	
	<a href="#">/*</a>	

Code	Description	Links
	<p><b>Example Value</b> Schema</p> <pre>{   "errorStatus": "UNAUTHORIZED",   "errorMessage": "Invalid username or password",   "errorDescription": "uri=/auth/login",   "errorTimestamp": "2025-01-01T12:00:00" }</pre>	
404	User not found	No links

GET	/auth/verify Verify user email	^
Verifies user email using a verification token		
<b>Parameters</b>		<b>Try it out</b>
Name	Description	
<b>token</b> * required string (query)	token	
<b>Responses</b>		
Code	Description	Links
200	Email verified successfully	No links
	Media type	

Code	Description	Links
	<p>*/*</p> <p>Controls Accept header.</p> <p><b>Example Value</b> Schema</p> <pre>{   "email": "john.doe@example.com",   "passwordChangedSuccess": true,   "statusMessage": "Password changed successfully",   "enabled": true,   "verified": true }</pre>	
401	Token expired	No links
	<p>Media type</p> <p>*/*</p> <p><b>Example Value</b> Schema</p> <pre>{   "errorStatus": "UNAUTHORIZED",   "errorMessage": "Invalid username or password",   "errorDescription": "uri=/auth/login",   "errorTimestamp": "2025-01-01T12:00:00" }</pre>	
404	Token or user not found	No links
	<p>Media type</p> <p>*/*</p> <p><b>Example Value</b> Schema</p> <pre>{   "errorStatus": "UNAUTHORIZED",   "errorMessage": "Invalid username or password",   "errorDescription": "uri=/auth/login",   "errorTimestamp": "2025-01-01T12:00:00" }</pre>	

GET /auth/me Get authenticated user profile



Returns profile information of the currently logged-in user

**Parameters****Try it out**

No parameters

## Responses

Code	Description	Links
200	<p>Profile retrieved successfully</p> <p>Media type</p> <div style="border: 2px solid #ccc; padding: 5px; width: fit-content;">/*</div> <p>Controls Accept header.</p> <p><b>Example Value</b> Schema</p> <pre>{   "id": 1,   "addresses": [     {       "id": 0,       "addressLine1": "string",       "addressLine2": "string",       "city": "string",       "country": "string"     }   ],   "userName": "john_doe",   "email": "john.doe@example.com",   "first_name": "John",   "last_name": "Doe",   "is_enabled": true,   "is_verified": true }</pre>	No links
401	<p>Unauthorized</p> <p>Media type</p> <div style="border: 2px solid #ccc; padding: 5px; width: fit-content;">/*</div> <p><b>Example Value</b> Schema</p> <pre>{   "errorStatus": "UNAUTHORIZED",   "errorMessage": "Invalid username or password",   "errorDescription": "uri=/auth/login",   "errorTimestamp": "2025-01-01T12:00:00" }</pre>	No links
404	<p>User not found</p> <p>Media type</p> <div style="border: 2px solid #ccc; padding: 5px; width: fit-content;">/*</div> <p><b>Example Value</b> Schema</p> <pre>{   "errorStatus": "UNAUTHORIZED",</pre>	No links

Code	Description	Links
	<pre>"errorMessage": "Invalid username or password", "errorDescription": "uri=/auth/login", "errorTimestamp": "2025-01-01T12:00:00" }</pre>	

## test-controller



GET /test



### Parameters

[Try it out](#)

No parameters

### Responses

Code	Description	Links
200	OK	<i>No links</i>

Media type

/\*

Controls Accept header.

**Example Value** Schema

string

## product-controller



GET /products



### Parameters

[Try it out](#)

No parameters

## Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

Controls Accept header.

Example Value Schema

```
[  
  {  
    "id": 0,  
    "name": "string",  
    "price": 0.1,  
    "shortDescription": "string",  
    "longDescription": "string"  
  }  
]
```

## order-controller



GET /orders



### Parameters

[Try it out](#)

No parameters

### Responses

Code	Description	Links
200	OK	No links

Media type

\*/\*

Controls Accept header.

Example Value Schema

```
[  
  {  
    "id": 0,  
    "name": "string",  
    "price": 0.1,  
    "shortDescription": "string",  
    "longDescription": "string"  
  }  
]
```

Code	Description	Links
<pre> "user": {     "id": 0,     "userN<sup>m</sup>ame": "string",     "email": "string",     "firstName": "string",     "lastName": "string",     "isEnabled": true,     "isVerified": true }, "address": {     "id": 0,     "addressLine1": "string",     "addressLine2": "string",     "city": "string",     "country": "string" }, "orderQuantities": [     {         "id": 0,         "product": {             "id": 0,             "name": "string",             "price": 0.1,             "shortDescription": "string",             ...         }     } ] </pre>		

## Schemas



### ResetPasswordBodyDTO ^ Collapse all object

Reset password request payload

```

newPassword* > Expand all string [8, 2147483647] characters matches ^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{8,}$
token* > Expand all string ≥ 1 characters

```

### ErrorResponseDTO ^ Collapse all object

Standard error response

```

errorStatus > Expand all string
errorMessage > Expand all string
errorDescription > Expand all string
errorTimestamp > Expand all string date-time

```

### UserStatusDTO ^ Collapse all object

User status response

```

email > Expand all string
passwordChangedSuccess > Expand all boolean

```

```
statusMessage > Expand all string  
enabled boolean  
verified boolean
```

## RegistrationBodyDTO ^ Collapse all object

User registration request payload

```
username* > Expand all string [3, 2147483647] characters  
password* > Expand all string [8, 2147483647] characters matches ^(?=.*[A-Za-z])(?=.*\d)[A-Za-z\d]{8,}$_  
firstName* > Expand all string ≥ 1 characters  
lastName* > Expand all string ≥ 1 characters  
email* > Expand all string email ≥ 1 characters
```

## Address ^ Collapse all object

```
id integer int64  
addressLine1 string  
addressLine2 string  
city string  
country string
```

## UserDTO ^ Collapse all object

User profile information

```
id > Expand all integer int64  
addresses > Expand all array<object>  
userName > Expand all string  
email > Expand all string  
first_name > Expand all string  
last_name > Expand all string  
is_enabled > Expand all boolean  
is_verified > Expand all boolean
```

## LoginBodyDTO ^ Collapse all object

Login request payload

```
username* > Expand all string ≥ 1 characters  
password* > Expand all string ≥ 1 characters
```

**LoginResponseDTO** ^ Collapse all **object**

Login response containing JWT token and login status

**jwt** > Expand all **string**

**success** > Expand all **boolean**

**failureMessage** > Expand all **string**

**ForgetPasswordBodyDTO** ^ Collapse all **object**

Forgot password request payload

**email\*** > Expand all **string** ≥ 1 characters

**username\*** > Expand all **string** ≥ 1 characters

**Product** ^ Collapse all **object**

**id** **integer** **int64**

**name** **string**

**price** **number** **double**

**shortDescription** **string**

**longDescription** **string**

**LocalUser** ^ Collapse all **object**

**id** **integer** **int64**

**userName** **string**

**email** **string**

**firstName** **string**

**lastName** **string**

**isEnabled** **boolean**

**isVerified** **boolean**

**OrderQuantities** ^ Collapse all **object**

**id** **integer** **int64**

**product** > Expand all **object**

**quantity** **integer** **int32**

**WebOrder** ^ Collapse all **object**

**id** **integer** **int64**

**user** > Expand all **object**

address > Expand all **object**  
orderQuantities > Expand all **array<object>**  
localUser > Expand all **write-only object**