**API Doc (PRODUCT)**

To access the APIs below, you must first generate an OAuth token. Once generated, copy the token and include it in the request header as a bearer token. This process follows a client-based architecture, where authorization is handled using a client ID and secret. Only after completing this step will you be able to consume the APIs.

**Token API:**

Endpoint: <http://localhost:8181/oauth/token>

Curl :

curl --location --request POST 'http://localhost:8181/oauth/token' \

--header 'Content-Type: application/x-www-form-urlencoded' \

--data-urlencode 'grant\_type=client\_credentials' \

--data-urlencode 'client\_id=edstruments' \

--data-urlencode 'client\_secret=258888d4-9fd5-4000-b26a-f0148ac3bf06'

**Project Additional Features -**

1. OAuth Authentication
2. Request Body Interceptors
3. Response Body Interceptors
4. Auth Interceptors
5. Api Req Resp table to track all generated request and response
6. Encryption on Api Req Resp table
7. Log4j file integration
8. Ip Address track in logs
9. Global Exception Handler (Managing Custom Exception)

We are using UUID as our primary source of identification, since disclosing the primary key is not a good practice.

1. Get product by ID **:**

Endpoint: <http://localhost:8181/api/products/6514bf1e-9228-49aa-b04e-18819c0e55bf>

Curl :

curl --location --request GET 'http://localhost:8181/api/products/6514bf1e-9228-49aa-b04e-18819c0e55bf' \

--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzY29wZSI6WyJyZWFkIiwid3JpdGUiXSwiZXhwIjoxNzI5NDM2MDk3LCJqdGkiOiI5UDRyWDVkNUc3Rm1hLVVUVHhyT2pEN2FOYm8iLCJjbGllbnRfaWQiOiJlZHN0cnVtZW50cyJ9.EvOZENcvHxm2pc\_SJw9JGusE7I0pucRwyG-cfzWcw58' \

--header 'Content-Type: application/json' \

--data-raw '{

"name":"test2",

"description":"test",

"price":"1.02090"

}'

1. Get list of product

Endpoint: <http://localhost:8181/api/products>

Curl :

curl --location --request GET 'http://localhost:8181/api/products' \

--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzY29wZSI6WyJyZWFkIiwid3JpdGUiXSwiZXhwIjoxNzI5NDM2MDk3LCJqdGkiOiI5UDRyWDVkNUc3Rm1hLVVUVHhyT2pEN2FOYm8iLCJjbGllbnRfaWQiOiJlZHN0cnVtZW50cyJ9.EvOZENcvHxm2pc\_SJw9JGusE7I0pucRwyG-cfzWcw58' \

--data-raw ''

1. Create new product

Endpoint:<http://localhost:8181/api/products>

Curl :

curl --location --request POST 'http://localhost:8181/api/products' \

--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzY29wZSI6WyJyZWFkIiwid3JpdGUiXSwiZXhwIjoxNzI5NDM5MTM2LCJqdGkiOiJPU3phZG9hM2U3NWpxTXlfMTVsczFEWkhTZ0kiLCJjbGllbnRfaWQiOiJlZHN0cnVtZW50cyJ9.sYIxpXCtHWXuy-SPvX2wJX-lJgeBX37qPdAKcS65sAM' \

--header 'Content-Type: application/json' \

--data-raw '{

"name":"test5",

"description":"test",

"price":"1.02090"

}'

1. Update existing product

Endpoint: <http://localhost:8181/api/products/57c1db55-1f4b-4568-9f25-0284f95c5ffe>

Curl :

curl --location --request PUT 'http://localhost:8181/api/products/57c1db55-1f4b-4568-9f25-0284f95c5ffe' \

--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzY29wZSI6WyJyZWFkIiwid3JpdGUiXSwiZXhwIjoxNzI5NDQwMjY3LCJqdGkiOiJSX0JDMlV6SGFvVzcxbEwweElOektpVDV2aW8iLCJjbGllbnRfaWQiOiJlZHN0cnVtZW50cyJ9.P1g4nmzzfLmv2m\_5yGwS3dqtBASCnh7IdGv5sBEIlCw' \

--header 'Content-Type: application/json' \

--data-raw '{

"name":"test4",

"description":"test",

"price":"1.020"

}'

1. Delete a product

Endpoint:

<http://localhost:8181/api/products/6514bf1e-9228-49aa-b04e-18819c0e55bf>

Curl :

curl --location --request DELETE 'http://localhost:8181/api/products/6514bf1e-9228-49aa-b04e-18819c0e55bf' \

--header 'Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzY29wZSI6WyJyZWFkIiwid3JpdGUiXSwiZXhwIjoxNzI5NDM2MDk3LCJqdGkiOiI5UDRyWDVkNUc3Rm1hLVVUVHhyT2pEN2FOYm8iLCJjbGllbnRfaWQiOiJlZHN0cnVtZW50cyJ9.EvOZENcvHxm2pc\_SJw9JGusE7I0pucRwyG-cfzWcw58' \

--data-raw ''

Validation :

1. Name cannot be null or empty - length limit - 100 character
2. Description cannot be null or empty, length limit - 300 character
3. Price cannot be null, has to be greater than Zero

The validation mentioned above works by consuming the existing annotation like @Size, etc on the field level.