ProductsApi Documentation

The API manages products so that the department can create, update and delete products.

Methods

- GetProducts
- AddProduct
- UpdateProduct
- DeleteProduct

Status codes

This API uses HTTP status codes to communicate with the API consumer.

- 200 OK - Response to a successful GET, PUT or DELETE.
- 201 Created - Response to a POST that results in a creation.
- 400 Bad Request - Malformed request; request body validation errors.
- 404 Not Found - When a non-existent resource(ProductId) is requested.
- 500 Server Error - Something went wrong on the API server.

Methods:

GetProducts

This method will return all the products.

GET	/api/Products

<u>Parameters:</u> No Parameters

<u>Response Example:</u>

<u>Code</u>	<u>Response Body</u>
| 200 | ```json
[
  {
    "id": "0c3762f3-7053-4876-a312-4ab8302df811",
    "name": "Razer",
    "description": "Gaming Headsets"
  },
  {
    "id": "d0a34125-afb7-4313-ab15-8f3e244957d3",
    "name": "Apple",
    "description": "Apple Iphone"
  },
  {
    "id": "cb13339d-b3cc-4c49-88c0-e2cfe112a2e6",
    "name": "Dell",
    "description": "Dell Laptop"
  }
]
``` |

AddProduct

This method will add a product.

POST	/api/Products

Parameters: No Parameters

Request Body Example (application/json):

```
{
  "name": "Nike",
  "description": "Tshirt"
}
```

Validations:

Error Message	Description
Product name cannot be empty.	Validate if the product name is empty or not.
Product already exists.	Validate if the product name already exists or not.

Response Example:

Code	Response Body
201	```{ "id": "da2c9024-a8ab-4203-b4f3-2db95b7eea5f", "name": "Nike", "description": "Tshirt" }```

UpdateProduct

This method will update a product.

PUT	/api/Products

Parameters: No Parameters

Request Body Example (application/json):

```json
{
  "id": "3fa85f64-5717-4562-b3fc-2c963f66afa6",
  "name": "Nike",
  "description": "Shoes"
}
```

Validations:

Error Message	Description
Product name cannot be empty.	Validates if the product name is empty or not.
Product already exists.	Validates if the product name already exists or not.
Product not found.	Validates if the product Id is valid or not.

Response Example:

Code	Response Body
200	```json\n{\n "id": "da2c9024-a8ab-4203-b4f3-2db95b7eea5f",\n "name": "Nike",\n "description": "Shoes"\n}\n```

DeleteProduct

This method will delete a product.

DELETE `/api/Products`

Parameters: No Parameters

Request Body Example (application/json):

```
{
  "id": "fec2ba5f-4220-46e6-b6cb-393d33cc1618"
}
```

Validation:

Error Message	Description
Product not found.	Validates if the product Id is valid or not.

Response Example:

Code	Response Body
200	{}