**SAMPLE DATABASE API**

This API allows you to add, modify and delete products, categories and brands.

The API is available at <http://127.0.0.1:5000> locally.

**ENDPOINTS:**

# **FETCHING DATA**

# STATUS

GET/Status

Returns the status of the API.

# LIST OF PRODUCTS DETAIL

GET/products

Returns a list of products.

# GET SINGLE PRODUCT DETAIL

GET /product/:productid

The product ID should be an integer.

Retrieve detailed information about a product.

# LIST OF CATEGORIES

GET/categories

Returns a list of categories.

# GET A SINGLE CATEGORY

GET /category/:categoryid

The category ID should be an integer.

Retrieve detailed information about a category.

# LIST OF BRANDS

GET/brands

Returns a list of brands.

# GET A SINGLE BRAND

GET /brand/:brandid

The brand ID should be an integer.

Retrieve detailed information about a brand.

# Products General Information

GET/

# Single Product General Information

GET/:productid

# **ADDING DATA**

# ADD PRODUCTS

Allows you to insert a new product.

The request body needs to be in JSON format and include the following properties:

* "product\_name": string
* "brand\_id": integer
* "category\_id": integer
* "model\_year": integer
* "list\_price": decimal

**Example**

POST /products

{

    "product\_name" : "Yamaha",

    "brand\_id":8,

    "category\_id":6,

    "model\_year":2018,

    "list\_price": 200.99

}

The response body will contain the product Id.

# ADD CATEGORIES

Allows you to insert a new category.

The request body needs to be in JSON format and include the following properties:

* "category\_name": string

**Example**

POST /categories

{

    "category\_name" : "Yamaha",

}

The response body will contain the category Id.

# ADD BRANDS

Allows you to insert a new brand.

The request body needs to be in JSON format and include the following properties:

* "brand\_name": string

**Example**

POST /brands

{

    "brand\_name" : "Toyota",

}

The response body will contain the brand Id.

# **MODIFYING DATA**

# MODIFY PRODUCT

PUT/products/:productid

Allows you to modify a product based on its ID. Product ID should be an integer.

The request body needs to be in JSON format and include the following properties:

* "product\_name": string
* "brand\_id": integer
* "category\_id": integer
* "model\_year": integer
* "list\_price": decimal

**Example**

PUT /products/12

{

    "product\_name" : "Suzuki",

    "brand\_id":7,

    "category\_id":4,

    "model\_year":2019,

    "list\_price": 201.99

}

The response body will contain the updated product.

# MODIFY CATEGORY

PUT/categories/:categoryid

Allows you to modify a category based on its ID. Category ID should be an integer.

The request body needs to be in JSON format and include the following properties:

* "category\_name": string

**Example**

PUT /category/4

{

    "category\_name" : "Suzuki",

}

The response body will contain the updated category.

# MODIFY BRAND

PUT/brands/:brandid

Allows you to modify a brand based on its ID. Brand ID should be an integer.

The request body needs to be in JSON format and include the following properties:

* "brand\_name": string

**Example**

PUT /brand/4

{

    "brand\_name" : "Suzuki",

}

The response body will contain the updated brand.

# **DELETING DATA**

# DELETING PRODUCT

DELETE/products/:productid

Delete an existing product.

The request body needs to be empty.

**Example**

DELETE/products/1

# DELETING CATEGORY

DELETE/ categories/:categoryid

Delete an existing category.

The request body needs to be empty.

**Example**

DELETE/categories/1

# DELETING BRAND

DELETE/ brands/:brandid

Delete an existing brand.

The request body needs to be empty.

**Example**

DELETE/brands/1

**POSSIBLE ERRORS**

|  |  |  |  |
| --- | --- | --- | --- |
| **STATUS CODE** | **TYPE** | **DETAILS** | **RESPONSE** |
| 400 | Bad Request | Improper Formatting (missing brackets) | This request was not properly formatted. Please send again. |
| 404 | Not Found | The requested resource could not be found | The requested URL was not found on the server. If you entered the URL manually, please check your spelling and try again. |
| 415 | Unsupported Media Type | When API supports only JSON, and client sends xml etc. | The application/xml MediaType is not supported. |
| 422 | Un-processable Entity. | When unexpected data is sent. | Request had invalid or missing data. |