

ecommerce

POST add categories

http://127.0.0.1:8000/api/categories

POST /api/categories

This endpoint allows the client to create a new category.

Request Body

- raw (application/json)
 - name: (string) The name of the category to be created.

Response

The response is a JSON object with the following schema:

json

```
{
  "name": "string",
  "updated_at": "string",
  "created_at": "string",
  "id": "integer"
}
```

- Status: 201
- Content-Type: application/json

Body raw (json)

json

```
{
  "name": "Electronics"
}
```

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this page.

http://127.0.0.1:8000/api/categories/

GET /api/categories/

This endpoint retrieves a list of categories.

Request

No request body is required for this request.

Response

The response will be a JSON array containing objects with the following properties:

- `id` (number): The unique identifier for the category.
- `name` (string): The name of the category.
- `created_at` (string): The timestamp when the category was created.
- `updated_at` (string): The timestamp when the category was last updated.

Example response:

json

```
[
  {
    "id": 0,
    "name": "",
    "created_at": "",
    "updated_at": ""
  }
]
```

GET get categories by id

http://127.0.0.1:8000/api/categories/2

Retrieve Category by ID

This endpoint retrieves a specific category by its ID.

Request

- Method: GET
- URL: `http://127.0.0.1:8000/api/categories/2`

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this page.

The response will be a JSON object with the following schema:

```
json

{
  "type": "object",
  "properties": {
    "id": {
      "type": "integer"
    },
    "name": {
      "type": "string"
    },
    "created_at": {
      "type": "string"
    }
  }
}
```

PUT update categories

`http://127.0.0.1:8000/api/categories/1`

Update Category

This endpoint is used to update a specific category.

Request

- Method: PUT
- URL: `http://127.0.0.1:8000/api/categories/1`
- Body:

```
json

{
  "name": "decorations"
}
```

Response

- Status: 204
- Content-Type: text/xml
- Body: null

JSON Schema

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this page.

```
{
  "type": "object",
  "properties": {
    "status": {
      "type": "number"
    },
    "content-type": {
      "type": "string"
    },
    "body": {
      "type": "null"
    }
  }
}
```

Body raw (json)

```
json

{
  "name": "decorations"
}
```

DELETE delete categories

http://127.0.0.1:8000/api/categories/1

Delete Category

This endpoint is used to delete a specific category.

Request

- Method: DELETE
- URL: `http://127.0.0.1:8000/api/categories/1`

Response

The response for this request is a 204 status code with no content.

JSON Schema

```
json
```

```
{  
  "type": "object",
```

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this page.

```
  "additionalProperties": false  
}
```

POST add sub categories

http://127.0.0.1:8000/api/subcategories

The endpoint `POST /api/subcategories` is used to create a new subcategory.

Request Body

- The request body should be in JSON format and include the following parameters:
 - `name` : (string) The name of the subcategory.
 - `category_id` : (integer) The ID of the category to which the subcategory belongs.

Example:

json

```
{  
  "name": "Mobile Phones",  
  "category_id": 3  
}
```

Response (JSON Schema)

json

```
{  
  "type": "object",  
  "properties": {  
    "name": {"type": "string"},  
    "category_id": {"type": "integer"},  
    "updated_at": {"type": "string"},  
    "created_at": {"type": "string"},  
    "id": {"type": "integer"}  
  }  
}
```

Body raw (json)

```
{  
  "name": "Mobile Phones",  
  "category_id": 3  
}
```

GET get all sub categories

Get Subcategories

This endpoint makes an HTTP GET request to retrieve a list of subcategories.

Request Body

This request does not require a request body.

Response

- Status: 200
- Content-Type: application/json

The response will contain an array of subcategories, where each subcategory object includes the following fields:

- `id` (number): The unique identifier for the subcategory.
- `name` (string): The name of the subcategory.
- `category_id` (number): The identifier of the category to which the subcategory belongs.
- `created_at` (string): The timestamp indicating the creation date of the subcategory.
- `updated_at` (string): The timestamp indicating the last update date of the subcategory.
- `category` (object): An embedded category object containing the following fields:
 - `id` (number): The unique identifier for the category.
 - `name` (string): The name of the category.
 - `created_at` (string): The timestamp indicating the creation date of the category.
 - `updated_at` (string): The timestamp indicating the last update date of the category.

GET get sub categories by id

<http://127.0.0.1:8000/api/subcategories/1>

The endpoint makes an HTTP PUT request to update a subcategory with the specified ID. The response returned is a JSON object with the following schema:

JSON object with the following schema.

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this page.

```
{
  "type": "object",
  "properties": {
    "id": {
      "type": "integer"
    },
    "name": {
      "type": "string"
    },
    "category_id": {
      "type": "integer"
    }
  }
}
```

PUT update sub categories

http://127.0.0.1:8000/api/subcategories/1

This endpoint makes an HTTP PUT request to update a subcategory with the specified ID. The request body should contain the new name of the subcategory and the ID of the category it belongs to.

Request Body

- name (string, required): The new name of the subcategory.
- category_id (integer, required): The ID of the category to which the subcategory belongs.

Response

The response is in JSON format and has the following schema:

json

```
{
  "type": "object",
  "properties": {
    "id": {
      "type": "integer"
    },
    "name": {
      "type": "string"
    },
    "category_id": {
      "type": "integer"
    }
  }
}
```

Body raw (json)

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this page.

```
{  
  "name": "Smartphones",  
  "category_id": 1  
}
```

DELETE delete sub categories

http://127.0.0.1:8000/api/subcategories/2

Delete Subcategory

This endpoint is used to delete a specific subcategory.

Request

- Method: DELETE
- URL: `http://127.0.0.1:8000/api/subcategories/2`

Response

The response for this request is a 204 status code with no content.

JSON Schema

json

```
{  
  "type": "object",  
  "properties": {},  
  "required": [],  
  "additionalProperties": false  
}
```

POST add product

http://127.0.0.1:8000/api/products

Create a Product

This endpoint allows the creation of a new product.

- Method: POST

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this page.

Request Body

- Type: JSON
 - `name` : (string) The name of the product.
 - `description` : (string) The description of the product.
 - `price` : (number) The price of the product.
 - `subcategory_id` : (number) The ID of the subcategory to which the product belongs.

Response (JSON Schema)

json

```
{
  "type": "object",
  "properties": {
    "name": { "type": "string" },
    "description": { "type": "string" },
    "price": { "type": "number" },
    "subcategory_id": { "type": "number" },
    "updated_at": { "type": "string" },
    "created_at": { "type": "string" },
    "id": { "type": "number" }
  }
}
```

Body raw (json)

json

```
{
  "name": "iPhone 14",
  "description": "Latest Apple iPhone",
  "price": 999.99,
  "subcategory_id": 1
}
```

GET get all products

http://127.0.0.1:8000/api/products

The endpoint retrieves a list of products from the server.

The response of this request can be documented as a JSON schema:

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this page.

```
{
  "type": "array",
  "items": {
    "type": "object",
    "properties": {
      "id": {
        "type": "integer"
      },
      "name": {
        "type": "string"
      }
    }
  }
}
```

GET get product by id

Retrieve Product Details

This endpoint retrieves details of a specific product by its ID.

Request

- Method: GET
- URL: `http://127.0.0.1:8000/api/products/2`

Response

The response will be in JSON format and will adhere to the following schema:

json

```
{
  "type": "object",
  "properties": {
    "id": { "type": "number" },
    "name": { "type": "string" },
    "description": { "type": "string" },
    "price": { "type": "string" },
    "subcategory_id": { "type": "number" },
    "created_at": { "type": "string" },
    "updated_at": { "type": "string" },
    "subcategory": {

```

PUT update product

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this page.

This endpoint allows the client to update a specific product by sending an HTTP PUT request to the specified URL.

Request Body

- The request body should be in raw format with a JSON payload containing the following parameters:
 - `name` (string): The updated name of the product.
 - `description` (string): The updated description of the product.
 - `price` (number): The updated price of the product.
 - `subcategory_id` (number): The ID of the updated subcategory to which the product belongs.

Response (200 - OK)

The response to this request will be in JSON format with the following schema:

json

```
{
  "type": "object",
  "properties": {
    "id": {
      "type": "number"
    },
    "name": {
      "type": "string"
    },
    "description": {
      "type": "string"
    }
  }
}
```

Body raw (json)

json

```
{
  "name": "iPhone 14 Pro",
  "description": "Updated model with better features",
  "price": 1099.99,
  "subcategory_id": 1
}
```

DELETE delete product

 This documentation is not published yet, this is only a preview. Changes to the publishing settings will reload this page.

documented as a JSON schema.