

UNIVERSIDAD NACIONAL AUTONOMA DE
NICARAGUA
UNAN-LEÓN



Componente: Software como un servicio

Tema: Api de Cosmetico

Grupo:1

Integrante: Gema Mercedes Bravo Lanzas

Documentación de la API de Cosméticos

Introducción

Esta API permite gestionar un catálogo de productos cosméticos. Los desarrolladores pueden realizar operaciones CRUD (Crear, Leer, Actualizar, Eliminar) sobre los cosméticos, así como filtrar por categoría y buscar por nombre o descripción.

```
class CosmeticsController < ApplicationController
```

```
  before_action :set_cosmetico, only: [:show, :update, :destroy]
```

```
  # GET /cosmetics
```

```
  def index
```

```
    @cosmetics = Cosmetico.all
```

```
    render json: @cosmetics
```

```
  end
```

```
  # búsqueda por Categoría
```

```
  def index
```

```
    if params[:category]
```

```
      @cosmetics = Cosmetico.where(category: params[:category])
```

```
    else
```

```
      @cosmetics = Cosmetico.all
```

```
    end
```

```
    render json: @cosmetics
```

```
  end
```

```
  #busqueda especifica
```

```
  def search
```

```
    @cosmetics = Cosmetico.where("nombre ILIKE :query OR descripcion ILIKE :query", query: "%#{params[:q]}%")
```

```
    render json: @cosmetics
```

end

GET /cosmeticos/:id

def show

render json: @cosmetico

end

POST /cosmeticos

def create

@cosmetico = Cosmetico.new(cosmetico_params)

if @cosmetico.save

render json: @cosmetico, status: :created, location: @cosmetico

else

render json: @cosmetico.errors, status: :unprocessable_entity

end

end

PATCH/PUT /cosmeticos/:id

def update

if @cosmetico.update(cosmetico_params)

render json: @cosmetico

else

render json: @cosmetico.errors, status: :unprocessable_entity

end

end

DELETE /cosmeticos/:id

```
def destroy
```

```
  @cosmetico.destroy
```

```
end
```

```
private
```

Encontrar un cosmético por su ID

```
def set_cosmetico
```

```
  @cosmetico = Cosmetico.find(params[:id])
```

```
  rescue ActiveRecord::RecordNotFound
```

```
    render json: { error: 'Cosmético no encontrado' }, status: :not_found
```

```
end
```

Permitir solo los parámetros permitidos

```
def cosmetico_params
```

```
  params.require(:cosmetico).permit(:nombre, :categoria, :precio, :descripcion)
```

```
end
```

```
end
```

Endpoints

1. Obtener todos los cosméticos

- **URL:** /cosmeticos
- **Método:** GET
- **Descripción:** Devuelve una lista de todos los cosméticos. También se puede filtrar por categoría.
- **Parámetros:** (esto es opcional)
 - category: Filtra los cosméticos por categoría.

2. Buscar cosméticos

URL: /cosmeticos/search

Método: GET

Descripción: Busca cosméticos que contengan el término de búsqueda en su nombre o descripción.

Parámetros

q: Término de búsqueda.

. Obtener un cosmético por ID

- **URL:** /cosmeticos/:id
- **Método:** GET
- **Descripción:** Devuelve los detalles de un cosmético específico.
- **Parámetros:**
 - id (requerido): ID del cosmético.

Crear un nuevo cosmético

- **URL:** /cosmeticos
- **Método:** POST
- **Descripción:** Crea un nuevo cosmético.

Actualizar un cosmético

- **URL:** /cosmeticos/:id
- **Método:** PATCH o PUT
- **Descripción:** Actualiza los detalles de un cosmético específico.
- **Parámetros:**
 - id (requerido): ID del cosmético.

Eliminar un cosmético

- **URL:** /cosmeticos/:id
- **Método:** DELETE
- **Descripción:** Elimina un cosmético específico.
- **Parámetros:**
 - id (requerido): ID del cosmético.
- **Respuesta:**
 - **Código 204:** Sin contenido (indica que la eliminación fue exitosa).
 -

Conclusión

Esta API proporciona un acceso sencillo para gestionar cosméticos, permitiendo a los desarrolladores integrar funcionalidades relacionadas con productos cosméticos en sus aplicaciones. Tiene la opción de crear inventarios actualizados, búsqueda avanzada ya sea un producto específico por medio de su nombre, categoría o su información.