

# Category RESTful API Documentation

## Overview

Category RESTful API is designed to manage categories with functionalities to create, read, update, and delete category records. Authentication is required for all endpoints using an API Key.

---

## API Specification

- **Version:** 1.0
- **Base URL:** http://localhost:3020/apiandre
- **Authentication:** API Key in header (x-API-Key)

## Endpoints

### 1. List All Categories

**Endpoint:** GET /categories

- **Description:** Retrieves a list of all categories.
- **Authentication:** Required.
- **Response:**

```
{
  "code": 200,
  "status": "Success",
  "data": [
    {
      "id": "string",
      "name": "string"
    }
  ]
}
```

### 2. Create New Category

**Endpoint:** POST /categories

- **Description:** Creates a new category.
- **Authentication:** Required.
- **Request Body:**

```
{
  "name": "string"
}
```

**Response:**

```
{
  "code": 200,
  "status": "Success",
  "data": {
    "id": "string",
    "name": "string"
  }
}
```

### 3. Get Category by ID

**Endpoint:** GET /category/{categoryId}

- **Description:** Retrieves details of a category by its ID.
- **Authentication:** Required.
- **Path Parameter:**
  - categoryId: ID of the category (string).
- **Response:**

```
{
  "code": 200,
  "status": "Success",
  "data": {
    "id": "string",
    "name": "string"
  }
}
```

### 4. Update Category by ID

**Endpoint:** PUT /category/{categoryId}

- **Description:** Updates a category by its ID.
- **Authentication:** Required.
- **Path Parameter:**
  - categoryId: ID of the category (string).
- **Request Body:**

```
{
  "name": "string"
}
```

**Response:**

```
{
  "code": 200,
  "status": "Success",
  "data": {
    "id": "string",
    "name": "string"
  }
}
```

## 5. Delete Category by ID

**Endpoint:** DELETE /category/{categoryId}

- **Description:** Deletes a category by its ID.
- **Authentication:** Required.
- **Path Parameter:**
  - categoryId: ID of the category (string).
- **Response:**

```
{
  "code": 200,
  "status": "Category successfully deleted"
}
```

### Authentication

- **Type:** API Key
- **Header Name:** x-API-Key
- **Description:** Include the API Key in the header for all requests.

Example:

```
GET /categories HTTP/1.1
Host: localhost:3020
x-API-Key: your-api-key
```

### Data Models

## Category Object

```
{
  "id": "string",
  "name": "string"
}
```

## Create/Update Category

```
{
  "name": "string"
}
```

## Usage Example

### Retrieve All Categories

#### Request:

```
GET /categories HTTP/1.1
Host: localhost:3020
x-API-Key: your-api-key
```

#### Response:

```
{
  "code": 200,
  "status": "Success",
  "data": [
    {
      "id": "1",
      "name": "Electronics"
    },
    {
      "id": "2",
      "name": "Books"
    }
  ]
}
```

### Create a Category

#### Request:

```
POST /categories HTTP/1.1
Host: localhost:3020
Content-Type: application/json
x-API-Key: your-api-key
```

```
{
  "name": "Furniture"
}
```

### Response:

```
{
  "code": 200,
  "status": "Success",
  "data": {
    "id": "3",
    "name": "Furniture"
  }
}
```

### Notes

- Ensure the API Key is kept secure.
- Test the API endpoints using tools like Postman or cURL.
- Customize the x-API-Key value for your application as needed.