# Express.js Product API

A simple RESTful API built with Express.js for managing a collection of products. Features include:

* CRUD operations for products
* Middleware for logging, authentication, and validation
* Custom error handling with appropriate HTTP status codes
* Advanced features: filtering, pagination, search, and statistics

## 🛠️ Prerequisites

* Node.js v18 or higher
* npm (comes with Node.js)

## 🚀 Installation

1. **Clone the repository**

* git clone <your-repo-url>  
  cd express-api-project

1. **Install dependencies**

* npm install

1. **Configure environment variables**
   * Copy .env.example to .env
   * cp .env.example .env
   * Edit .env and set your API key and port:
   * PORT=3000  
     API\_KEY=12345
2. **Start the server**

* node server.js

The server will be running at http://localhost:3000.

## 📁 Project Structure

express-api-project/  
├── server.js # Main application file  
├── package.json # npm dependencies and scripts  
├── .env.example # Example environment variables  
├── README.md # Project documentation  
├── routes/  
│ └── products.js # Product route handlers  
├── utils/  
│ ├── auth.js # Authentication middleware  
│ ├── logger.js # Logging middleware  
│ ├── validation.js # Request validation middleware  
│ ├── NotFoundError.js # Custom 404 error class  
│ └── ValidationError.js # Custom 400 error class  
└── node\_modules/ # Installed packages

## 🔐 Environment Variables

| Variable | Description | Example |
| --- | --- | --- |
| PORT | Port number the server listens on | 3000 |
| API\_KEY | API key for protected endpoints | 12345 |

## 📘 API Endpoints

### Authentication

All protected routes require the header:

x-api-key: <your\_api\_key>

### Products

#### GET /api/products

Retrieve a list of products. Optional query parameters:

* category: filter by category
* search: search by name
* page: page number (default 1)
* limit: items per page (default 5)

**Example**:

GET http://localhost:3000/api/products?category=Electronics&search=phone&page=1&limit=10

#### GET /api/products/:id

Retrieve a single product by its ID.

**Example**:

GET http://localhost:3000/api/products/9b020971-2d48-4950-b13d-e3ef1eb44d31

#### POST /api/products

Create a new product.

**Headers**:

x-api-key: 12345  
Content-Type: application/json

**Body**:

{  
 "name": "Laptop",  
 "description": "Powerful machine",  
 "price": 1299,  
 "category": "Electronics",  
 "inStock": true  
}

#### PUT /api/products/:id

Update an existing product by ID.

**Headers** and **Body** same as POST.

#### DELETE /api/products/:id

Delete a product by ID.

**Headers**:

x-api-key: 12345

## 📊 Statistics Endpoint

#### GET /api/products/stats

Retrieve product statistics (total count and count by category).

**Headers**:

x-api-key: 12345

**Example Response**:

{  
 "totalProducts": 3,  
 "countByCategory": {  
 "Electronics": 2,  
 "Books": 1  
 }  
}

## 📝 License

This project is licensed under the MIT License.