Planty of Food

Realizzato da Samuele Furnari

Project

Planty of Food

01

Planty of Food mira a rendere il cibo plant-based più accessibile attraverso una nuova iniziativa focalizzata sui gruppi di acquisto. Questo progetto prevede lo sviluppo di una piattaforma separata dal sito e-commerce esistente per facilitare questi gruppi. Il mio ruolo è quello di progettare e implementare API RESTful JSON per gestire le funzionalità dei gruppi.

Requirements

REST Structure

02

Le API dovranno rispettare l'architettura REST, con particolare attenzione al naming, ai metodi HTTP e agli status code di risposta. Inoltre dovranno consentire operazioni di inserimento (POST), modifica (PUT) e cancellazzione (DELETE) sui prodotti, sugli utenti e sugli ordini. Oltre queste operazioni le API consentiranno di visuallizare (GET) tutti gli oridini, potendoli anche filtrare per "id" o in preferenza per "data di inserimento"

Tools

Framework & Library

03

Gli strumenti utilizzati per la realizzazione di questo progetto sono diversi:
ENV: libreria per la gestione delle variabili d'ambiente
Node.js: ambiente che permette l'esecuzione di JavaScript lato server
Express: framework progettato per costruire applicazioni web e API
Nodemon: framework che facilita lo sviluppo, monitorando cambiamenti
Mongoose: libreria di modellazione oggetti (ODM) per MongoDB e Node

Endpoint

/api/product

04

Questo endpoint gestisce tutte le operazioni relative ai prodotti, i metodi:
GET: /api/product/:id? mostra tutti i prodotti o soltanto l'id equivalente
POST: /api/product/ crea un nuovo prodotto con propietà "name"
PUT: /api/product/:id modifica un prodotto collegato all'id esistente
DELETE: /api/product/:id elimina un prodotto collegato all'id esistente

Endpoint

/api/user

05

Questo endpoint gestisce tutte le operazioni relative agli user, i metodi:
GET: /api/user/:id? mostra tutti gli utenti o soltanto l'id equivalente
POST: /api/user/ crea un utente con propietà "name", "surname", "email"
PUT: /api/user/:id modifica un utente collegato all'id esistente
DELETE: /api/user/:id elimina un utente collegato all'id esistente

Endpoint

/api/order

06

Questo endpoint gestisce tutte le operazioni relative agli order, i metodi:
GET: /api/order/:id? mostra tutti gli ordini o soltanto l'id equivalente
GET: /api/order?date=2002-11-28 mostra tutti gli ordini filtrati per date
POST: /api/order/ crea un ordine con propietà "products", "users"
PUT: /api/order/:id modifica un ordine collegato all'id esistente
DELETE: /api/order/:id elimina un ordine collegato all'id esistente

Database

MongoDB

07

La scelta del database è ricaduta su MongoDB, un database NoSQL orientato ai documenti, altamente scalabile e flessibile. L'utilizzo di MongoDB insieme a Mongoose ha permesso di gestire in modo efficiente e flessibile i dati del progetto, consentendo operazioni CRUD rapide e sicure. La struttura dei dati è stata ben definita e l'uso delle variabili di ambiente ha garantito la sicurezza delle credenziali di accesso.

Useful Links

GitHub

08

https://github.com/samuelefrni/RESTful-API

La realizzazione di questo progetto ha rappresentato un'opportunità significativa per approfondire la mia comprensione e padronanza di Node.js. Ogni fase del progetto, dalla configurazione iniziale, alla gestione dei dati e alla creazione delle API, ha contribuito ad approfondire la mia comprensione e a consolidare le mie competenze in questo campo. Per ulteriori informazioni ti consiglio di consultare il file README su GitHub.