



Pratica S7/L5

CRUDAZON

Stai creando la parte front-end di uno shop online. In particolare sarai responsabile della creazione di un back-office, dove gli amministratori possono aggiungere e modificare i prodotti.

L'obiettivo di oggi è connettere un'interfaccia statica alle API per poter ricevere prodotti, crearne di nuovi, modificarli una volta creati e cancellarli all'occorrenza.

Questo è l'endpoint principale:

<https://striveschool-api.herokuapp.com/api/product/>

Questo è il modello di un prodotto:

```
{
  "name": "Nokia 3310", // REQUIRED
  "description": "Indestructible cellphone", // REQUIRED
  "brand": "Nokia", // REQUIRED
  "imageUrl": "https://example.com/3310.jpg", // REQUIRED
  "price": 99, // REQUIRED
  "_id": "5d318e1a8541744830bef139", // SERVER GENERATED
  "userId": "6385f782597b9d001545386b", // SERVER GENERATED
  "createdAt": "2019-07-19T09:32:10.535Z", // SERVER GENERATED
  "updatedAt": "2019-07-19T09:32:10.535Z", // SERVER GENERATED
  "__v": 0 // SERVER GENERATED
}
```

Per creare nuovi prodotti dovrai partire da questo modello come riferimento, e formarlo con alcune delle proprietà richieste per poi inviarlo come payload della chiamata POST.

OGNI CHIAMATA DOVRÀ ESSERE AUTENTICATA! L'autenticazione di queste API è una "Token Based Authentication" per rendere privato l'accesso ai suoi contenuti. Senza essere autenticato non potrai ottenere i dati di cui hai bisogno.

Per prima cosa dovrai recuperare un tuo token personale registrandoti su: <https://strive.school/studentlogin>

Il token dovrà venire utilizzato come "authorization" header.



Obiettivi generali:

- Avere una pagina back-office, in cui si potranno inserire i prodotti specificando i parametri obbligatori e facoltativi.
- Una homepage, dove l'utente possa vedere i prodotti disponibili
- Una pagina di dettaglio in cui visualizzare tutti i dettagli del prodotto.

Task:

- Nella pagina di back-office: usa POST su /product con un payload per creare una nuova risorsa.
- Aggiungi un bottone per la funzionalità di modifica di un prodotto già creato in precedenza (usa una PUT su /product/[PRODUCT_ID]).
- Aggiungi un bottone per la cancellazione di uno specifico prodotto già esistente (usa DELETE su /product/[PRODUCT_ID])
- Aggiungi una validazione di base per la creazione/modifica del prodotto nel form.
- Aggiungi un bottone "Reset" per resettare il form.
- Nella Homepage: premendo un bottone "modifica" su un prodotto si dovrà poterlo modificare.
- Nella pagina Dettaglio: A questa pagina ci si arriverà cliccando sulla card in homepage.

EXTRAS:

- In back-office: I bottoni "reset" e "delete" dovranno chiedere conferma prima di procedere con l'operazione.
- In homepage: aggiungi un indicatore di caricamento affianco al titolo principale della pagina durante il caricamento delle risorse.
- Crea un sistema di gestione degli errori. Mostra all'utente un messaggio di errore specifico per le varie tipologie di problema, quando qualcosa va storto, attraverso l'utilizzo di componenti di Bootstrap appropriati.

FAQ:



“Nelle mie API non c’è niente, come mai?”

All'inizio ognuno di voi partirà con array di prodotti vuoto, createne qualcuno prima! 😊

“Ricevo un errore con 500, c’è qualcosa che non va lato server?”

Molto probabilmente no, controlla sul tab Network che la tua richiesta contenga tutti i campi richiesti, con le chiavi corrette, e che sia inviata in formato JSON. La preview della risposta dal server spesso ti dice cosa c’è che non va. Di solito:

1. Manca un campo obbligatorio
2. Hai una “duplicate key”, che significa che qualcosa con lo stesso valore esiste già.
3. Stai inviando il tipo di dato sbagliato (es. una stringa invece di un numero sul campo “price”)

Imparare a leggere gli errori è molto importante, sono i tuoi migliori amici!

Dicci cosa ne pensate

Quanto è stato utile questo article?

Peggior articolo 😞

Non utile 😐

Utile 😊

Molto Utile 🥳