

Introduzione

Questo è un libro pratico per gli sviluppatori web che vogliono affinare le loro competenze e costruire applicazioni full stack complete secondo gli standard del settore. Alla fine del libro sarete esperti nell'uso di GraphQL e React per le vostre esigenze di sviluppo full stack.

Questo libro vi aiuterà a implementare uno stack robusto con React, Apollo, Node.js e SQL seguendo buone pratiche. Ci concentreremo anche sulla soluzione di problemi complessi con GraphQL, come l'astrazione di un'architettura di database a più tabelle e la gestione del caricamento di immagini.

A chi è rivolto questo libro

Questo libro si rivolge a sviluppatori web che vogliono migliorare le loro competenze e costruire applicazioni full stack complete seguendo gli standard del settore. Il lettore ideale è qualcuno che vuole esplorare come usare GraphQL, React, Node.js e SQL per scrivere applicazioni complete con questo stack.

Struttura del libro

- Il Capitolo 1, *Preparazione dell'ambiente di sviluppo*, inizia con l'architettura dell'applicazione, esaminando i concetti fondamentali e preparando una configurazione funzionante di React. Vedremo come React e webpack si adattino l'uno all'altro ed esamineremo alcuni scenari di base per il lavoro con React. Vedremo anche come effettuare il debug del front end con React Dev Tools.
- Il Capitolo 2, *Configurazione di GraphQL con Express.js*, si concentra sull'impostazione di Express.js come sistema primario per servire il nostro back end. Vedremo come usare la funzionalità di routing di Express.js per implementare varie API. Inoltre, alla fine del capitolo, imposteremo un endpoint che accetti richieste GraphQL attraverso il package Apollo Server. Per essere sicuri che tutto funzioni a dovere, esamineremo rapidamente l'uso di Postman per testare e verificare la funzionalità del back end.
- Il Capitolo 3, *Connessione al database*, esamina come usare GraphQL per conservare e interrogare i dati. Come esempio, useremo il tradizionale SQL per costruire un'applicazione completa con MySQL. Per semplificare il codice del database, useremo

Sequelize, che consente di interrogare SQL Server con un normale oggetto JavaScript e lo tiene aperto se usiamo MySQL, MSSQL, PostgreSQL o qualsiasi altro file SQLite. Costruiremo modelli e schemi per utenti e post in Apollo e Sequelize.

- Il Capitolo 4, *Integrazione di React nel back end con Apollo*, spiega come agganciare Apollo a React e costruire componenti di front end. Questo capitolo non offre un'introduzione ai flussi di lavoro React di base ma spiega configurazioni specifiche per Apollo.
- Il Capitolo 5, *Componenti React riutilizzabili*, approfondisce la scrittura di componenti React più complessi e la condivisione di dati fra tali componenti.
- Il Capitolo 6, *Autenticazione con Apollo e React*, spiega i metodi comuni di autenticazione di un utente sul Web e in GraphQL e le relative differenze. Vi guiderà nella costruzione di un flusso di autenticazione completa secondo le pratiche migliori.
- Il Capitolo 7, *Gestione del caricamento di immagini*, tratta del caricamento di immagini via Apollo e del loro salvataggio in uno storage separato come AWS S3.
- Il Capitolo 8, *Routing in React*, spiega come implementare alcune altre funzionalità per l'utente finale, per esempio una pagina di profilo. A questo fine installeremo React Router.
- Il Capitolo 9, *Implementazione del rendering lato server*, spiega perché per molte applicazioni il rendering lato server sia una necessità. È importante per il SEO, ma può avere effetti positivi anche per gli utenti finali. Questo capitolo si concentrerà su come spostare l'applicazione a una configurazione che preveda il rendering lato server.
- Il Capitolo 10, *Subscriptions: abbonamenti in tempo reale*, si concentra sulla costruzione di una funzionalità di chat in tempo reale, con un sistema di notifiche. Ogni secondo è possibile che arrivi un nuovo messaggio e l'utente può esserne direttamente informato. Questa funzionalità verrà implementata attraverso una funzionalità più o meno sperimentale di GraphQL e Apollo chiamata abbonamento (*subscription*).
- Il Capitolo 11, *Scrittura di test*, usa il framework di unit testing Mocha e JavaScript. Questo capitolo si concentra principalmente sul testing del backend GraphQL e su quello delle applicazioni GraphQL.
- Il Capitolo 12, *Ottimizzazione di GraphQL con Apollo Engine*, risponde alle domande: “Quali sono le prestazioni della nostra API GraphQL?”, “Ci sono errori?” e “Come possiamo migliorare lo schema GraphQL?”. Risponderemo, in questo capitolo, utilizzando Apollo Engine.
- Il Capitolo 13, *Deployment continuo con CircleCI e Heroku*, è quello in cui vedremo come impostare la nostra app Heroku e come poter costruire e mettere in esercizio immagini Docker mediante un flusso di lavoro di deployment continuo.

Come ottenere il massimo da questo libro

Vi consigliamo di leggere il primo capitolo per essere sicuri di avere presenti i concetti fondamentali di React e webpack in generale. Dopo, potete leggere i capitoli che desiderate. I vari capitoli sono autonomi, ma sono ordinati per complessità crescente; ciascuno può richiedere tecniche spiegate nei capitoli precedenti e, più si va avanti, più complessa diventa l'applicazione.

L'applicazione è pensata per un uso reale, ma sono state escluse alcune parti, come una gestione accurata degli errori e altre funzionalità che un'applicazione per il mondo reale

deve avere, come gli analytics, che sono al di fuori del raggio d'azione di questo libro. Il libro si propone di insegnarvi le tecniche alla base di tutto ciò di cui parleremo; presuppone però che abbiate una buona conoscenza dei blocchi da costruzione e di come creare un'applicazione web con React e GraphQL. Sarà utile qualche esperienza precedente con lo sviluppo in JavaScript e magari in React, o almeno con qualche altro framework JavaScript moderno, perché molti concetti non sono specifici dell'applicazione ma sono buone pratiche in generale (per esempio, il rendering reattivo). Soprattutto, questo è un libro che potete usare per accelerare la vostra curva di apprendimento dello sviluppo React e GraphQL, concentrandovi sui capitoli che vi interessano maggiormente.

Scaricare i file degli esempi

Potete scaricare i file di codice degli esempi dal sito dell'editore originale, Packt Publishing, seguendo questa procedura

1. Effettuate il login o registratevi all'indirizzo www.packt.com.
2. Selezionate la scheda *SUPPORT*.
3. Fate clic su *Code Downloads & Errata*.
4. Inserite il titolo originale del libro, *Full-Stack Web Development with GraphQL and React*, nella casella *Search* e seguite le istruzioni che compaiono.

Il bundle del codice per il libro si trova anche su GitHub all'indirizzo <https://github.com/PacktPublishing/Hands-on-Full-stack-Web-Development-with-GraphQL-and-React>. Nel caso vi sia un aggiornamento del codice, verrete aggiornati sul repository GitHub esistente.

Convenzioni

In questo libro sono utilizzate alcune convenzioni per il testo.

CodiceNelTesto: è utilizzato per espressioni del codice che compaiono nel testo, nomi delle tabelle di database, nomi di cartelle, nomi ed estensioni di file, nomi di percorsi, URL fittizi, input dell'utente e handle di Twitter. Un esempio: “Passiamo come modello il nostro file `index.html` creato in precedenza”.

Un blocco di codice è reso tipograficamente in questo modo:

```
state = { posts: posts
}
```

Gli input da riga di comando sono scritti in questo modo:

```
mkdir ~/graphbook cd ~/graphbook
```

Corsivo: indica un termine che viene introdotto per la prima volta, una parola importante, gli elementi dell'interfaccia utente (quelli che compaiono nei menu o nelle finestre di dialogo). Un esempio: “Fatto questo, fate clic sul pulsante *Create*”.

NOTA

Avvertimenti, osservazioni importanti, trucchi e suggerimenti compaiono in questo formato.