
METODOLOGIE DI PROGRAMMAZIONE PER IL WEB

REST CON EXPRESS

In questo quarto laboratorio, inizierai a prendere familiarità con il framework Express e costruire una serie di API REST per supportare un task manager come quello che hai realizzato nei laboratori precedenti.

ESERCIZIO 1 – PROGETTARE LE API REST

Progetta un insieme di API REST per supportare le funzionalità principali di un task manager, che funzioni come quello sviluppato nei laboratori precedenti.

In particolare, per riassumere, un **task** è composto dalle seguenti proprietà:

- *id*, un identificatore univoco per il task (obbligatorio);
- *descrizione*, la descrizione testuale del task (obbligatoria);
- *importante*, se il task è importante o meno (default: “non importante”, modellato come un booleano);
- *privato* or *condiviso*, se il task è privato o condiviso con altri utenti (default: “private”); consideralo un valore booleano, come nei laboratori precedenti;
- *progetto*, il progetto in cui il task è inserito (opzionale, stringa);
- *scadenza*, la data (con eventuale ora) entro cui il task dovrà essere completato (opzionale);
- *completato*, se il task è stato completato o meno (booleano, default a “falso”).

L’API dovrà permettere a un’applicazione client di:

- Recuperare la lista di tutti i task disponibili.
- Creare un nuovo task, fornendo tutte le informazioni rilevanti (eccetto l’id).
- Recuperare un singolo task, dato il suo id.
- Aggiornare un task esistente, fornendo tutte le informazioni rilevanti (tutte le proprietà tranne l’id sovrascriveranno le proprietà correnti del task esistente con lo stesso ‘id’).
- Cancellare un task, dato il suo id.

In aggiunta, *potresti* volere un’API per:

- Segnare un task come “completato”.
- Recuperare una lista di tutti i task che rispettino una certa proprietà, per esempio tutti i task “importanti”, tutti quelli con una certa scadenza, ecc. Pensa alle funzionalità dei “filtri” e dei “progetti” dei laboratori precedenti per una lista completa di opzioni.

I dati passati o ricevuti dall’API devono essere in formato JSON.

Apri un editor di testo (VS Code va bene) e progetta lì la tua API. Assicurati di identificare quali sono le risorse che vuoi rappresentare. Potresti seguire questa struttura per la progettazione:

[Metodo HTTP] [URL, con eventuali parametri]
[Descrizione (una riga) su cosa fa questa API]
[Esempio di richiesta, con il suo corpo (se disponibile)]
[Esempio di risposta, con il suo corpo (se disponibile)]

Suggerimento: puoi seguire le informazioni fornite nelle slide e a lezione, quanto mostrato nell'esercizio settimanale e/o aiutarti con la Google API Design Guide (<https://cloud.google.com/apis/design/>).

ESERCIZIO 2 – IMPLEMENTA (E TESTA) L'API

Implementa l'API REST progettata in Node.js con Express. I task sono memorizzati in un database SQLite, come mostrato nell'esempio sviluppato durante le lezioni (<https://github.com/luigidr/2020-metweb-es-express>).

Testa le API realizzate con un tool come Postman (<https://www.postman.com/downloads/>) o RestClient per Firefox (<https://addons.mozilla.org/en-US/firefox/addon/restclient/>).

Suggerimenti:

1. Per iniziare rapidamente questo esercizio, puoi usare il database SQLite precompilato disponibile su GitHub: <https://github.com/luigidr/2020-metweb-lab4-express>
2. Per visualizzare e modificare il database, puoi utilizzare l'applicazione "DB Browser for SQLite", scaricabile da <https://sqlitebrowser.org/>