Scuola universitaria professionale della Svizzera italiana
Dipartimento tecnologie innovative

SUPSI

Progetto pratico: Sviluppo client http con framework Svelte

Students
Alessandro Cantoni
Mattia Verdolin

Course
Applicazioni web 2

Date
08.04.2025

STUDENTSUPSI

Indice

1 Introduction

Context and motivation – Project Overview

L'obiettivo principale è realizzare un componente configurabile in grado di:

- **Eseguire richieste HTTP** (GET, POST, PUT, DELETE)
- Gestire **headers e body personalizzati**
- **Visualizzare le risposte** (HTML, immagini, PDF)
- **Organizzare richieste in collections** gestibili via sidebar
- Offrire funzionalità di **ricerca, salvataggio, import/export**
- Il componente dev'essere **configurabile tramite parametri**

Viene utilizzato il framework Svelte con Vite.js per lo sviluppo rapido e modulare.

Context and motivation – Svelte Overview

Svelte è un moderno framework JavaScript per lo sviluppo di interfacce utente reattive, noto per la sua semplicità e prestazioni elevate.

A differenza di altri framework come React o Vue, Svelte non utilizza un virtual DOM, ma compila i componenti in JavaScript ottimizzato al momento della build.

Context and motivation – Svelte Overview

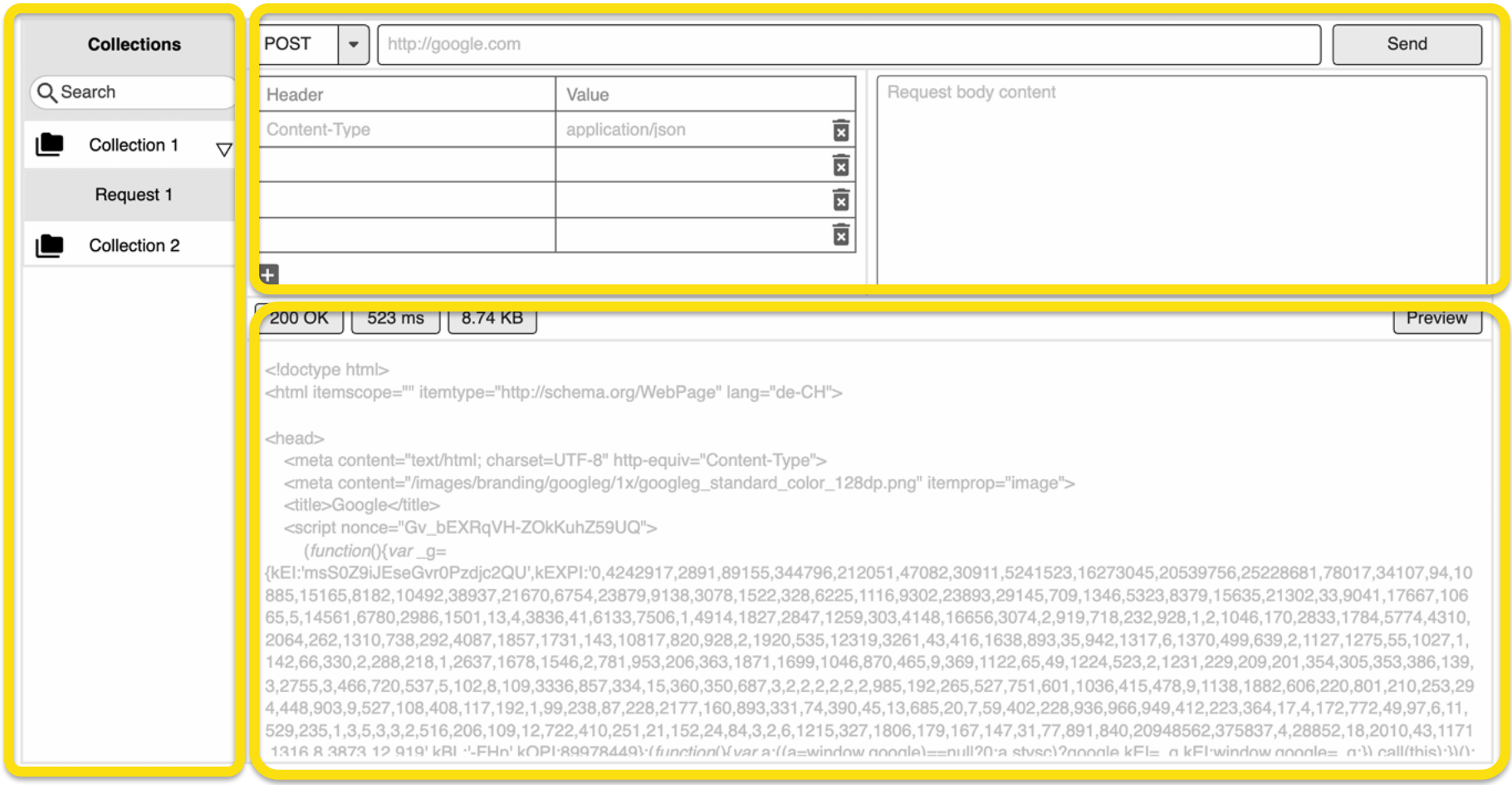
Questo si traduce in:

- Applicazioni più veloci e leggere
- Meno codice boilerplate
- Maggiore controllo sul comportamento dei componenti

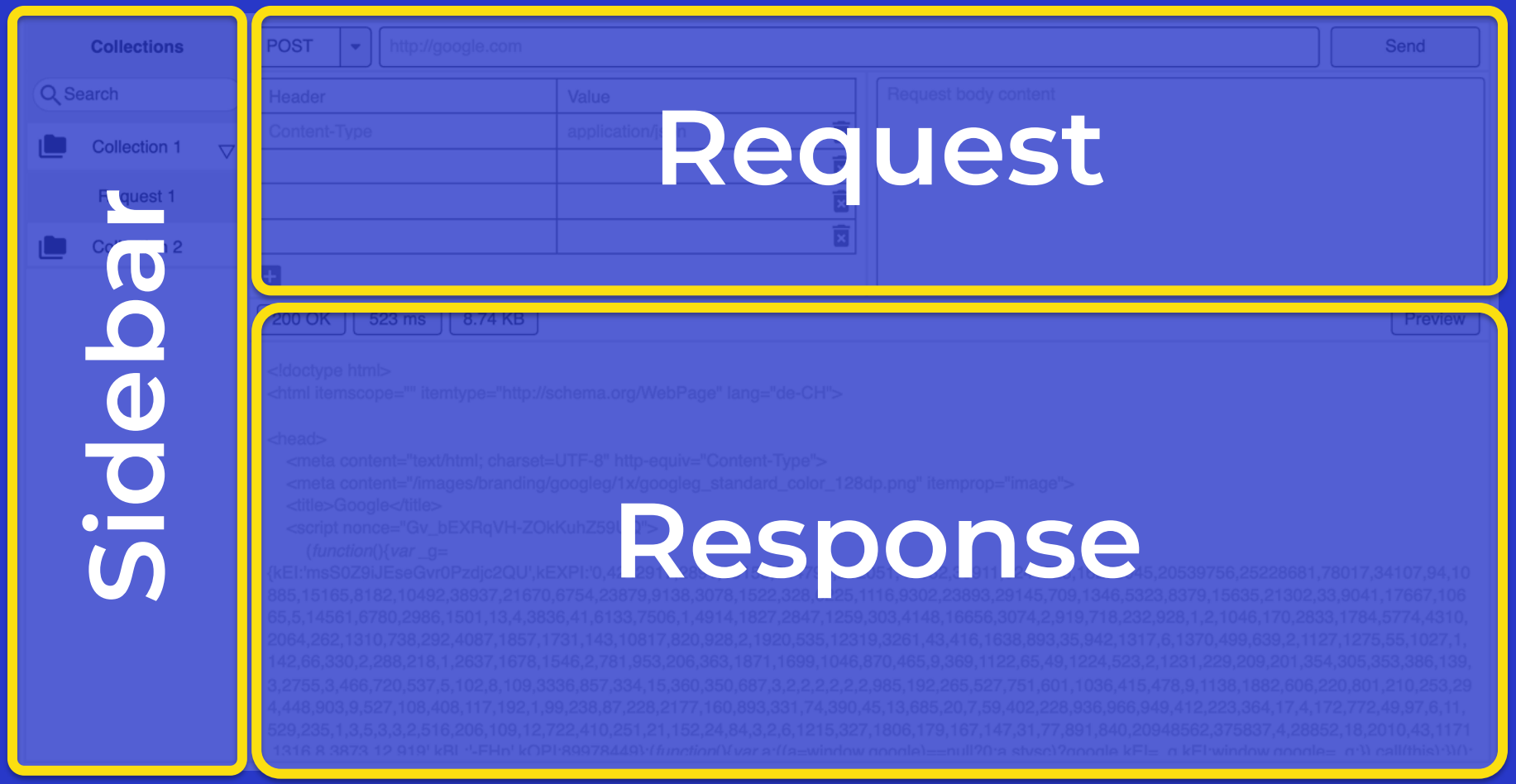
Nel nostro progetto, Svelte ha facilitato la comunicazione tra componenti, la gestione dello stato e la creazione di un'interfaccia dinamica e modulare.

2 Components

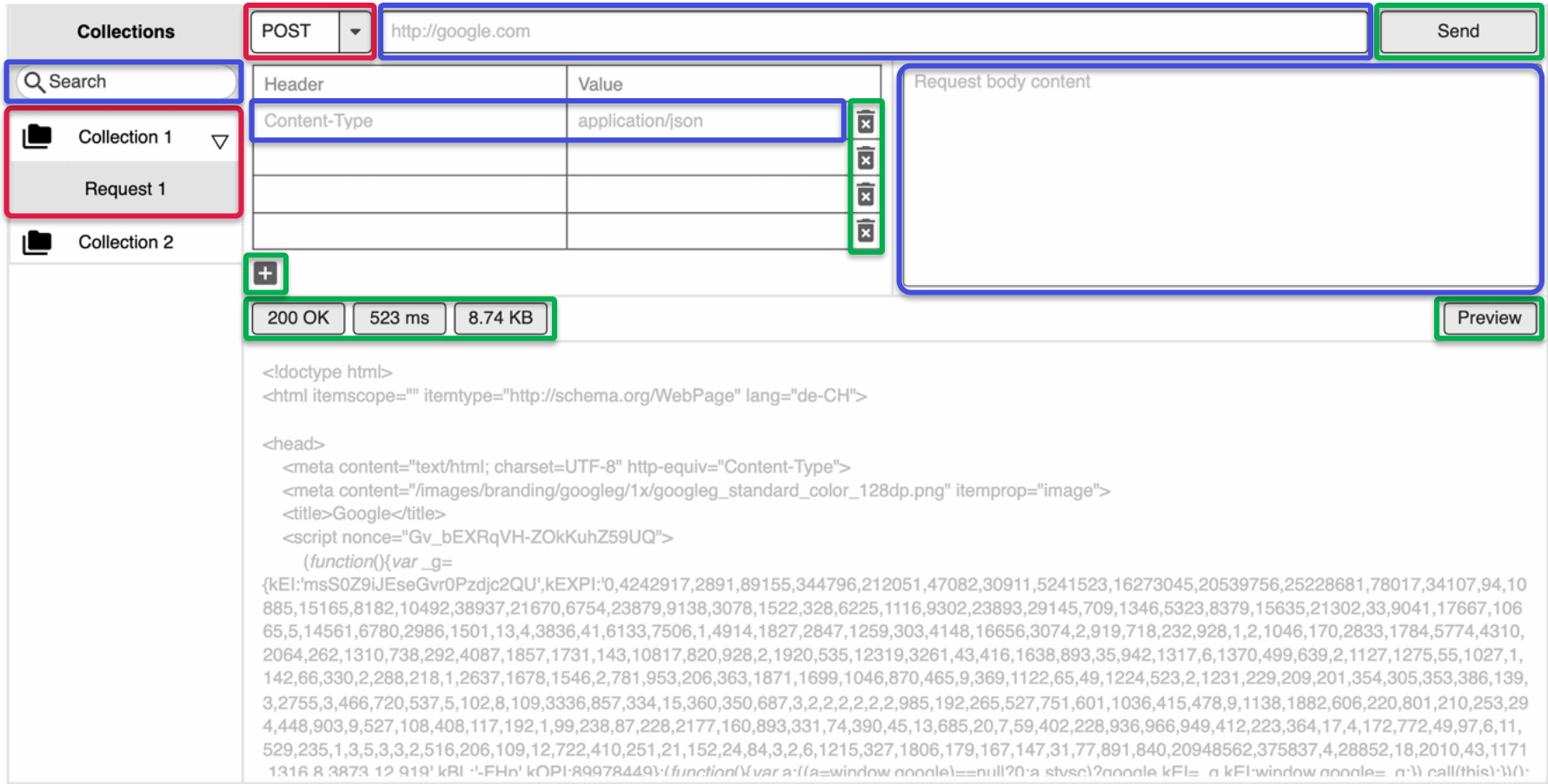
Divisione in componenti – big division



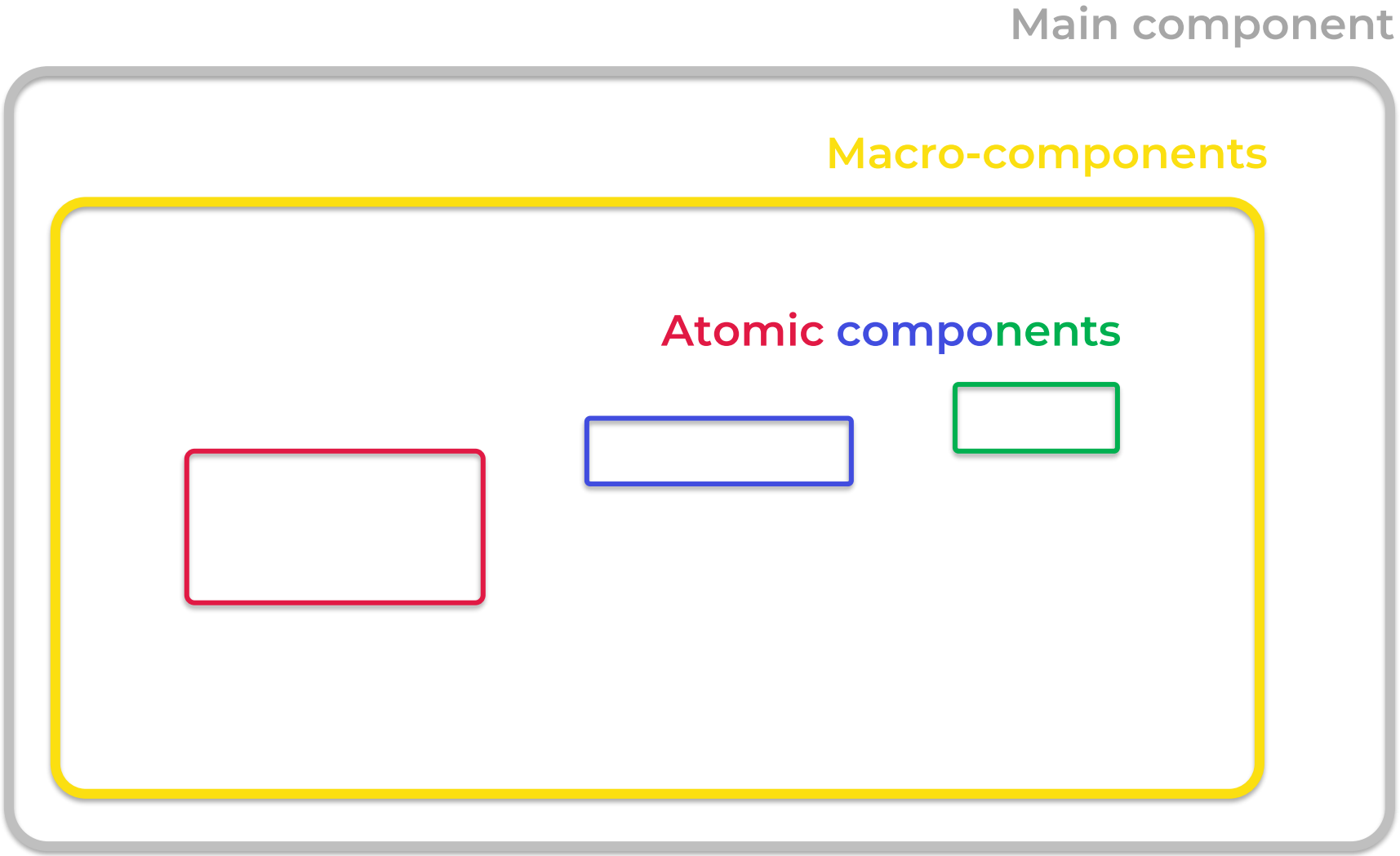
Divisione in componenti – big division



Divisione in componenti – small division



Divisione in componenti - composizione



3 Context

Svelte’s Context

Il **Context** di Svelte permette di condividere dati tra componenti senza doverli passare manualmente come props.

Utilizzato nel progetto per gestire la comunicazione tra:

- **Sidebar** (gestione delle richieste salvate)
- **RequestHandler** (modifica ed invio della richiesta)
- **ResponseHandler** (visualizzazione della risposta)

Context – how it works

Sidebar

Request handler

Response handler

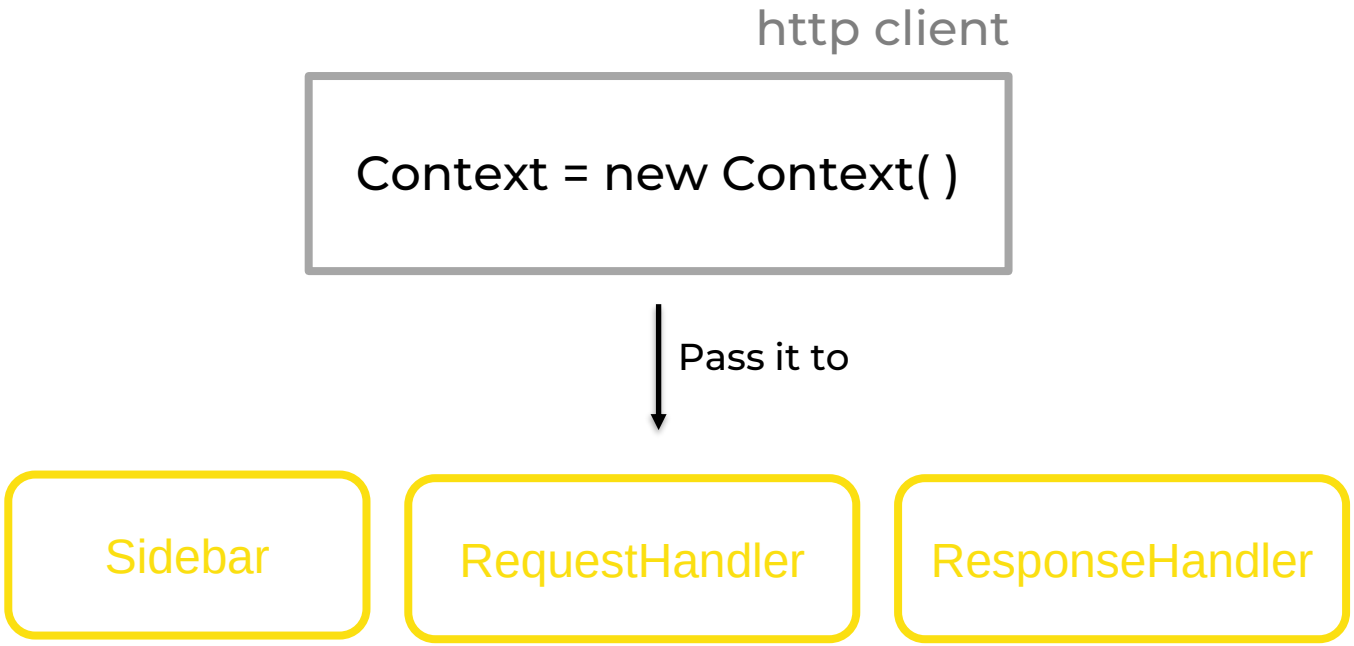
Context – how it works

In generale, utilizzando un context reattivo, ogni modifica allo stato globale viene rilevata automaticamente dai componenti che lo utilizzano, come se tutti guardassero nella stessa scatola condivisa

Ma...

Come abbiamo gestito l’indipendenza dello stato di ogni client?

Architecture logic



Ognuno dei componenti a cui passiamo il context come prop, accede agli store **indipendenti** attraverso getters e setters.

SUPSI

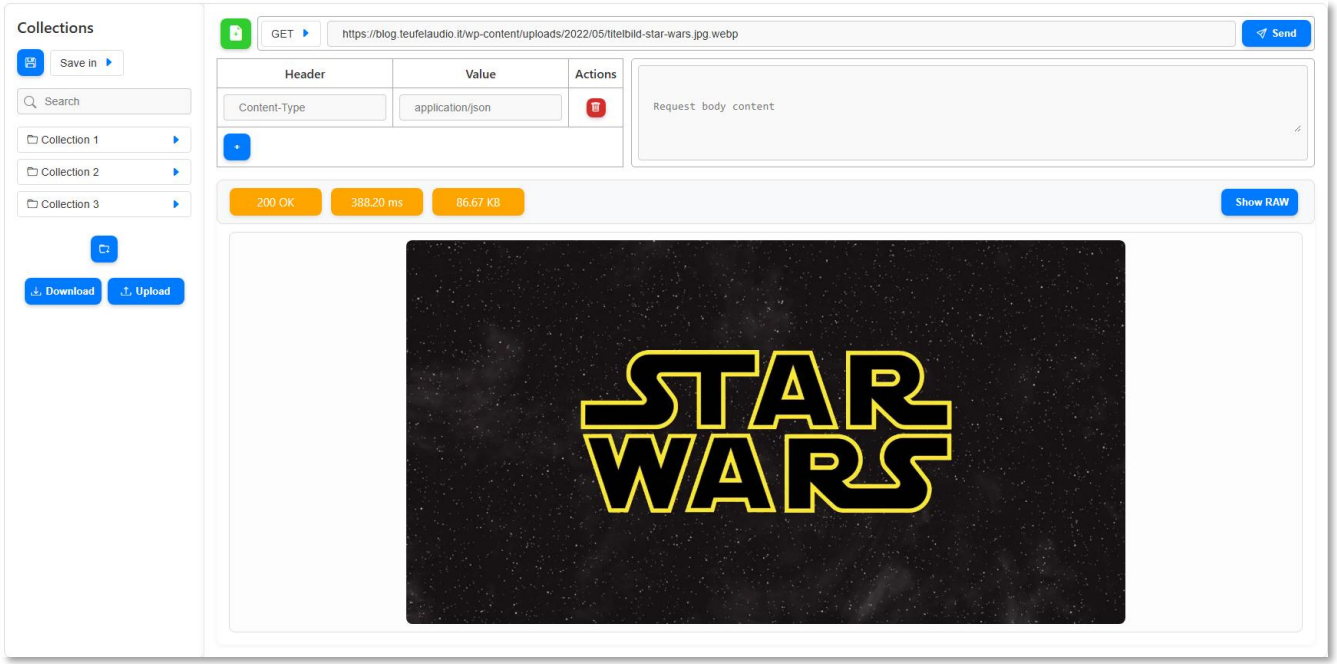
Demo Presentation

Clicca qui

5 Considerations and conclusions

Considerations and conclusions

Il progetto ha dimostrato come, grazie a Svelte e a un'architettura ben strutturata, sia possibile costruire un'interfaccia moderna, reattiva e modulare, mantenendo il codice pulito, scalabile e facile da mantenere.



Grazie!