

Riunione con IMOLA INFORMATICA

RAMtastic6

21 marzo 2024



email: ramtastic6@gmail.com

Informazioni sul documento

Redattori: Filippo T.

Verificatori: Davide B.

Verificatori esterni:



Indice

1 Partecipanti	3
2 Riassunto dell'incontro	3
2.0.1 Dubbi su collegamento tra <i>frontend</i> e <i>backend</i>	3
2.0.2 Realizzazione ordinazione collaborativa	3
2.0.3 Ruoli di progetto nella fase di sviluppo	4

1 Partecipanti

NOME	DURATA
Alessandro (IMOLA INFORMATICA)	30m
Filippo T.	30m
Leonardo B.	30m
Michele Z.	30m
Riccardo Z.	30m
Davide B.	30m
Samuele V.	30m

2 Riassunto dell'incontro

2.0.1 Dubbi su collegamento tra *frontend* e *backend*

JSON come mezzo di comunicazione Il primo dubbio esposto dal gruppo ha riguardato l'attuale gestione della comunicazione tra il *backend* e il *frontend*. L'attuale approccio implica l'utilizzo di file JSON, con le API REST del *frontend* che interrogano il database del *frontend* tramite questo tipo di file.

Alessandro ha comunicato al gruppo che è l'approccio corretto: i file JSON, in quanto file di testo, sono un buono strumento per mettere in comunicazione linguaggi di programmazione differenti.

Chiamate asincrone - 1 A seguito di un'ulteriore domanda riguardo il collegamento tra *frontend* e *backend*, in cui veniva chiesto se la funzione, messa a disposizione da *NextJs*, per fare chiamate asincrone direttamente lato server fosse un'altra possibile via con cui implementare questa comunicazione.

Il proponente ha risposto alla domanda chiarificando il fatto che la funzione in esame è più per servire dati statici, come documenti. Per dati applicativi, la scelta giusta è la comunicazione via API REST e JSON.

Chiamate asincrone - 2 E' stato in ultimo chiesto se gestire le *promises* tramite chiamate asincrone sia l'approccio corretto.

Il proponente ha risposto affermativamente, ed ha anche aggiunto di andare ad utilizzare dei *custom hooks* per evitare di dover riscrivere più volte il codice.

2.0.2 Realizzazione ordinazione collaborativa

Il gruppo ha richiesto consigli su quali siano le tecnologie in circolazione che aiutino nell'implementazione di pagine editabili da più di un utente alla volta, feature che il progetto software dovrà includere al suo interno.

Alessandro di IMOLA INFORMATICA ha introdotto brevemente, a tal proposito, il concetto di *socket*: in ambito web, permette di creare un canale di comunicazione bidirezionale che può essere continuo; viene usato tipicamente per chat o meccanismi di comunicazione collaborativa. Entrando nel merito dell'esempio della chat, il proponente ha spiegato che: essa ha un canale in background, sempre aperto verso il *backend*, che permette di comunicare verso due endpoint differenti; con la *socket*, è possibile creare un canale in cui ciò che modifica un "utente 1" viene inviato al backend, che si occupa di lanciare un evento alla pagina *frontend* di un "utente 2", la quale si aggiorna

correttamente secondo quanto modificato da "utente 1", e viceversa.

Il consiglio dunque è stato quello di andare a dare un occhio alla libreria di riferimento *socket.io*, e alla documentazione di *NESTJS* relativo.

2.0.3 Ruoli di progetto nella fase di sviluppo

L'incontro si è concluso con alcuni dubbi riguardo i ruoli presenti in un gruppo durante la fase di sviluppo del progetto software vero e proprio. Nello specifico, il gruppo ha chiesto se è prevista, in aziende come IMOLA INFORMATICA, la presenza di una figura professionale volta a: assicurare la correttezza di quanto sta venendo svolto dai programmatori; aiutare i programmatori nell'implementazione delle feature richieste dai documenti prodotti in precedenza; etc.

Anche in questo caso dal proponente è giunta risposta affermativa: Alessandro stesso si occupa dell'*architettura* del progetto (questo è il nome di questa parte dello sviluppo), sottolineando come anche in piccoli team, questo tipo di mansioni trovano comunque un loro attuatore nella figura del *project manager*.