

Verbale interno del 2024-03-04

2024-03-04 – v1.0



overture.unipd@gmail.com

Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin Gruppo <i>Overture</i>
Responsabile	Francesco Furno
Redattori	Alex Vedovato
Verificatori	Eleonora Amadori

Registro delle modifiche

Versione	Data	Autori	Verificatori	Dettaglio
1.0	2024-03-04	Alex Vedovato	Eleonora Amadori	Stesura del verbale

Indice

1) Contenuti del verbale	4
1.1) Informazioni sulla riunione	4
1.2) Ordine del giorno	4
2) Sintesi dell'incontro	4
3) Decisioni prese	4
4) Attività individuate	5

1) Contenuti del verbale

1.1) Informazioni sulla riunione

- **Luogo:** Chiamata Discord;
- **Ora di inizio:** 9:20;
- **Ora di fine:** 9:50;
- **Partecipanti:** Eleonora Amadori, Michele Bettin, Riccardo Bonavigo, Francesco Costantino Bulychov, Riccardo Fabbian, Francesco Furno, Alex Vedovato.

1.2) Ordine del giorno

- Discussione sull'incontro avvenuto con il Prof. Riccardo Cardin.

2) Sintesi dell'incontro

Subito dopo l'incontro con il Prof. Riccardo Cardin il gruppo si è riunito per discutere dei temi trattati. Il nostro prodotto adotta un'architettura esagonale, accessibile tramite richieste HTTP dall'unica porta di ingresso chiamata RequestPort; dal colloquio è emerso che questa interfaccia attualmente ha troppi compiti, infatti presenta 5 metodi che non hanno nulla a che fare l'uno con l'altro. Ora RequestPort centralizza quindi le dipendenze e non soddisfa il Single Responsibility Principle. Provvederemo a dividere i metodi, creando altre interfacce che fungeranno da porte d'ingresso all'interno dell'architettura. Inoltre, attualmente l'application logic del nostro prodotto non esegue nessun tipo di controllo sui dati prima di passarli alla business logic. Per rimediare a questo problema aggiungeremo una verifica tale che, se la libreria fallisce la conversione da JSON a oggetto, una determinata richiesta non verrà portata a termine prima ancora di raggiungere la business logic del prodotto.

Dopo ancora, visto che la libreria da noi scelta per l'implementazione del protocollo JMAP utilizza dei builder per la costruzione degli oggetti, abbiamo deciso, in accordo col professore, che riserveremo una piccola sezione per la descrizione di questo design pattern e del suo utilizzo da parte nostra all'interno della Specifica Tecnica.

Provvederemo poi a cambiare i nomi delle classi chiamate erroneamente "Impl" perchè implementano interfacce; verranno denominate "Repository", dato che il loro compito è quello di interfacciarsi con il database.

Successivamente, è emersa la necessità di rivalutare lo scopo della classe RethinkDBConnection, che ora come ora costituisce una configurazione per costruire una connessione tramite Guice, il che è preferibilmente evitabile.

Infine la classe methodHandler attualmente chiama il metodo corretto di un tipo tramite uno switch, rendendo il codice poco manutenibile. Il professore ci ha consigliato di approfondire il pattern filter in java e valutare se ha senso cambiare questa parte di codice, oppure di applicare il pattern chain-of-responsability già valutato. Entrambe le soluzioni non ci convincono troppo ma i Progettisti decideranno nei prossimi giorni qual è la cosa migliore da fare.

3) Decisioni prese

- Descrivere nella Specifica Tecnica il pattern builder utilizzato attraverso la libreria da noi scelta per l'implementazione del protocollo JMAP;
- Aggiungere altre interfacce come porte d'ingresso dell'esagono;
- Aggiungere la validazione iniziale dei dati, prima di passarli alla business logic;
- Refactoring dei nomi delle classi "Impl" in "Repository";
- Studiare il pattern filter per valutare un cambiamento nella classe methodHandler.

4) Attività individuate

ID	Dettaglio	Assegnatari
docs #323	Stesura di <i>questo</i> verbale	Alex Vedovato
docs #324	Specifica Tecnica: risoluzione dei problemi rilevati	Eleonora Amadori, Michele Bettin, Riccardo Fabbian, Francesco Furno