Verbale esterno del 2023-12-18

2023-12-27 — v1.0

overture.unipd@gmail.com

Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin *Zextras* Gruppo *Overture*
Responsabile	Michele Bettin
Redattori	Riccardo Bonavigo
Verificatori	Francesco Costantino Bulychov Riccardo Fabbian

Registro delle modifiche

Versione	Data	Autori	Verificatori	Dettaglio
1.0	2023-12-27	Riccardo Bonavigo	Francesco Costantino Bulychov	Finalizzazione del verbale
0.1	2023-12-18	Riccardo Bonavigo	Riccardo Fabbian	Prima bozza: stuttura e appunti dalla riunione

Indice

1) Contenuti del verbale

1.1) Informazioni sulla riunione

- **Luogo**: Chiamata sulla piattaforma dell'azienda *Zextras*;
- **Ora di inizio**: 17:00;
- **Ora di fine**: 18:00;
- **Partecipanti**: Eleonora Amadori, Michele Bettin, Riccardo Bonavigo, Francesco Costantino Bulychov, Francesco Furno
- **Partecipanti esterni**: Federico Rispo, Carlo Facci (COO dell'azienda).

1.2) Ordine del giorno

- Mostrare all'azienda il lavoro svolto finora sul PoC.

2) Sintesi dell'incontro

Nella fase iniziale dell'incontro abbiamo illustrato all'azienda le scelte prese sulle tecnologie del PoC. In particolare, useremo:
- **iNPUTmice/jmap**: la libreria consigliata;
- **Spark**: web server;
- **RethinkDB**: database NOSQL, utilizzando l'ORM integrato per le query (agevola mock);
- **Gradle** come sistema di build;
- **Docker** (e **Docker Compose**) per la gestione dei container;
- **Just** come esecutore dei comandi comuni.

Successivamente sono state mostrate le funzionalità attualmente implementate:
- comunicazione con il server;
- autenticazione degli utenti;
- ricezione dell'oggetto `session` definito dallo standard JMAP;
- upload e download di file generici (necessario per la gestione degli allegati);
- inserimento e recupero di oggetti generici nel database. Inserimento e recupero di email *utilizzando la libreria* non è ancora completo e verrà concluso nei prossimi giorni.

Sono stati discussi alcuni dubbi sui test di unità e sui test di integrazione.

Zextras ha confermato che stiamo andando nella direzione giusta con lo sviluppo del progetto.

3) Domande e risposte

3.1) Invio e recupero di una mail è sufficiente per l'RTB?

Sì, è sufficiente. È più importante focalizzarsi sulla struttura del codice, e sui test di unità/di integrazione: poche cose ma buone. Una buona struttura del codice consente facile estensibilità.

3.2) Come gestire inoltro tra domini istanze del server?

All'interno della specifica di JMAP non è presente alcuna informazione relativa all'inoltro delle mail ad altri server. Si può quindi considerare "libera".
I server JMAP esistenti utilizzano direttamente SMTP per l'inoltro ad altri domini, riservando JMAP solamente per lo scambio dati tra client mail e il server.

L'azienda consiglia di ignorare questo dettaglio fino a che non siano implementati tutti i requisiti obbligatori. Se in futuro verrà gestito questo punto, l'azienda consiglia due strade:
1. utilizzare endpoint custom per l'accettazione di mail in ingresso;

2. utilizzare una libreria per inoltro e ricezione tramite SMTP.

3.3) Allegati come blob su database?

Per ora gli allegati sono salvati come “blob” nel database. Nonostante la soluzione sia accettabile, l’azienda consiglia di utilizzare un “object storage” come minIO per migliorare le performance.

3.4) Utilità dei test di unità per il nostro progetto?

La maggior parte delle funzionalità del server dipendono direttamente dai “metodi JMAP” definiti nello standard. Di conseguenza, la maggior parte delle funzionalità di alto livello seguono un flusso del genere: ricezione di una richiesta -> dispatch -> chiamate al database.

I test di unità (con mock), sono generalmente meno utili di quelli di integrazione per funzioni poco complesse.
Si prenda ad esempio la funzionalità di autenticazione: la logica è piuttosto semplice, ma serve una chiamata al database (il recupero della password).
Poichè l’interazione con il database è *sempre* richiesta, ha più senso utilizzare un test di integrazione che uno di unità.

Inoltre, testare le funzionalità di alto livello con istanze reali del database testa implicitamente anche le funzioni di basso livello, seppur più sommariamente.

Ha senso quindi preferire test di integrazione a quelli di unità, e riservare questi ultimi per le funzioni intermedie (nel call stack) e per quelle “complesse”.

Inoltre, l’utilizzo dei test di integrazione permette di trovare errori di implementazione del protocollo JMAP (fondamentale per un server).