

JMAP, il nuovo protocollo standard per la comunicazione email

Contesto

Nonostante la forte crescita degli strumenti di messaggistica istantanea, l'email resta ancora oggi lo strumento di comunicazione digitale per eccellenza, con indici di crescita annuali molto alti. Già nel 2020, il 50% della popolazione mondiale faceva uso della posta elettronica e nel 2022 il numero di utenti è cresciuto fino ad arrivare a 4,26 miliardi. Nel corso dei prossimi due anni la crescita nell'utilizzo delle email è stimata al 3% (100 milioni di nuovi utenti ogni anno). Nel 2025 il numero degli utenti che inviano e ricevono messaggi di posta dovrebbe attestarsi intorno ai 4,6 miliardi.

L'email è un'invenzione del 1971 di Ray Tomlinson, ingegnere informatico statunitense in forze presso una società direttamente coinvolta nello sviluppo di ARPANET. È quindi ampiamente antecedente al World Wide Web (1989). Eppure la posta elettronica non sembra mai invecchiare: il suo utilizzo è in continuo aumento e ad oggi vengono inviati ogni secondo 3,13 milioni di email.

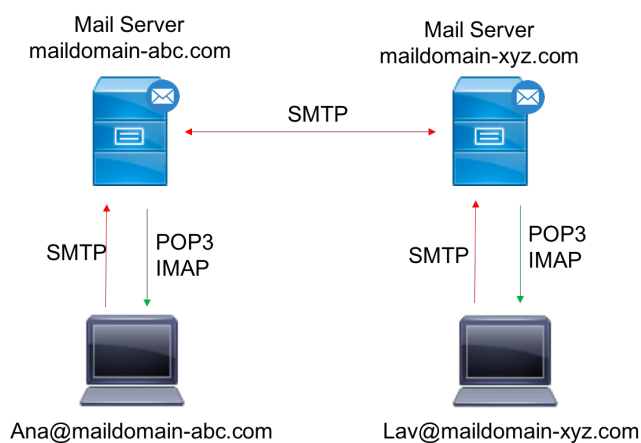
Il protocollo IMAP

IMAP (Internet Message Access Protocol) è un protocollo di comunicazione utilizzato per accedere alle email conservate su un server di posta compatibile. Risale al lontano 1986, come successore del precedente protocollo POP (*Post Office Protocol*) e consente un accesso più avanzato e interattivo al contenuto delle caselle di posta elettronica, ad esempio permettendo la sincronizzazione di una casella di posta in più dispositivi. Utilizzando IMAP infatti lo **stato** di ciascuna email (letto, non letto, etichette speciali, tag, etc) risulta memorizzato sul server: grazie a questo cambiamento architetturale è quindi possibile ritrovare tutte le modifiche apportate alla casella in tutti i client in cui è inizializzata.

Per riassumere possiamo quindi dire che i principali miglioramenti che IMAP ha apportato allo strumento dell'email sono quindi:

- **Sincronizzazione:** mantiene lo stato della casella di posta nel server, consentendo a uno o più client contemporaneamente di vedere le stesse email e le stesse modifiche da qualunque dispositivo.
- **Gestione delle cartelle:** consente di creare, rinominare e cancellare cartelle in qualsiasi client inizializzato, propagando il cambiamento nel server e di conseguenza in tutti gli altri client
- **Accesso selettivo:** consente agli utenti di scaricare solo parti specifiche delle email, come l'intestazione o il corpo del messaggio, risparmiando larghezza di banda e riducendo il tempo di download dei messaggi.
- **Operazioni lato server:** supporta una serie di operazioni che possono essere effettuate sul server, come la ricerca dei messaggi, la gestione delle etichette e l'eliminazione definitiva dei messaggi.

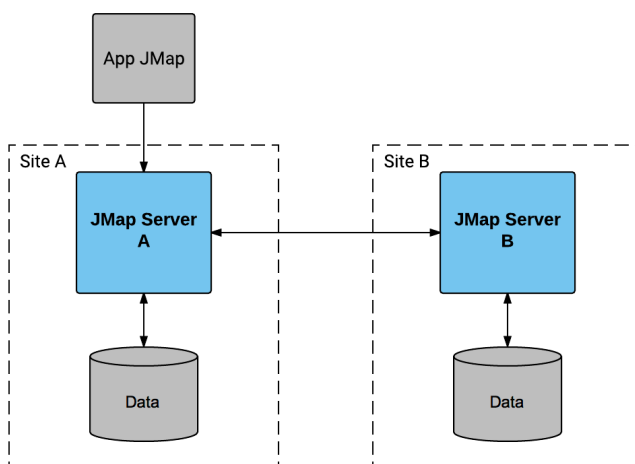
IMAP è diventato uno standard ampiamente utilizzato per l'**accesso alle email**, ed è ormai da tempo supportato da tutti i client email e dei server di posta. La sua adozione ha contribuito a migliorare l'esperienza degli utenti nell'accesso e nella gestione delle email da dispositivi diversi.



Come nasce JMAP

Progettato per semplificare e rendere standard l'interazione tra client e server nelle applicazioni di posta elettronica, JMAP va ben oltre i suoi predecessori ampliando il suo raggio d'azione, ad esempio, a calendario e contatti. Supportato da IETF (Internet Engineering Task Force) e approvato come standard, JMAP è un progetto open source che si è subito guadagnato il sostegno di tante organizzazioni.

Questo ha il **vantaggio** di definire un modello comune per la gestione delle varie tipologie di dato: così è possibile rendere i client di posta più facilmente interoperabili. Utilizza il **formato JSON** (JavaScript Object Notation) e definisce un set di API per permettere ai client di accedere ai dati e gestirli in modo efficiente.



L'idea alla base del protocollo JMAP è quella di ridurre la **quantità di dati** scambiati tra client e server, ottimizzando le richieste e le risposte per migliorare le prestazioni delle applicazioni. Grazie ai meccanismi di **sincronizzazione** efficienti sui quali si basa JMAP, è possibile mettere in comunicazione più client con il server facendo sì che tutte le modifiche vengano simultaneamente visualizzate.

JMAP nasce inoltre per essere **estendibile**, consentendo l'introduzione di nuove funzionalità senza dover modificare il protocollo.

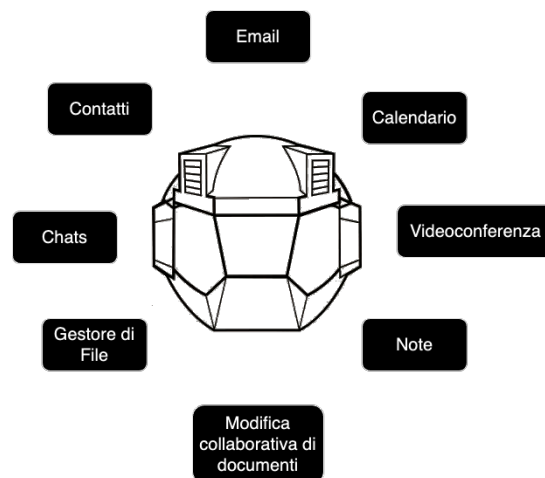
Principali differenze tra IMAP e JMAP

Soppesando le **differenze tra IMAP e JMAP** si possono capire le motivazioni che hanno spinto sviluppatori da tutto il mondo e la IETF a rimettere in discussione un protocollo ampiamente utilizzato come IMAP, cercando di crearne uno più efficiente e moderno:

- Mentre IMAP utilizza una **struttura di dati gerarchica** basata su cartelle per organizzare le email e altre informazioni di posta elettronica, JMAP utilizza un modello di dati basato su oggetti JSON
- Le funzionalità di **sincronizzazione** di IMAP sono meno sofisticate rispetto a quelle di JMAP: il più recente protocollo consente al client di gestire facilmente le modifiche apportate alle email ma anche ai contatti e agli eventi del calendario. Cosa che IMAP non permette di fare. Questo favorisce il supporto diretto di più applicazioni: email, calendario, contatti e altri strumenti possono essere gestiti allo stesso modo attraverso un'**unica piattaforma**
- JMAP è pensato per essere più **efficiente** in termini di trasmissione dei dati rispetto a IMAP. Le richieste e risposte JSON sono compatte e aiutano a ridurre la quantità di dati scambiati tra client e server, migliorando le prestazioni dell'applicazione.

Il Problema

Il prodotto principale di Zextras è Carbonio, una soluzione di collaborazione online on-premise che offre un'insieme di funzionalità il cui fulcro ruota attorno al servizio principale: l'email personale. Carbonio è distribuito come prodotto Open Source e senza supporto da parte di Zextras, oppure a pagamento con funzionalità aggiuntive e supporto diretto con un team di professionisti dedicati.



Come detto precedentemente, una delle caratteristiche principali di Carbonio è essere un software on-premise: Carbonio viene infatti rilasciato come pacchetto installabile nell'infrastruttura server del cliente, contrariamente ai prodotti a servizio (SaaS) erogati tramite un server centralizzato e gestito direttamente dal produttore (come ad esempio GMail o Spotify). I prodotti on-premise rispondono ad alcune esigenze particolari delle aziende:

- Alcune realtà statali e parastatali hanno l'obbligo della sovranità del dato, ossia i dati scambiati all'interno dei software utilizzati dai dipendenti devono risiedere in un server all'interno del territorio nazionale
- In alcune nazioni vige il vincolo di utilizzare solo software che garantiscono il pieno controllo del dato e che i server siano all'interno del territorio nazionale
- Alcune aziende hanno bisogno di strutture complesse dove modellare dei piani a consumo specifici, così da poter vendere account email con determinate quantità di quota utilizzabile dalla casella di un utente, a fronte del pagamento di un abbonamento rapportato al denaro speso.

Carbonio attualmente basa il proprio servizio email sui protocolli standard: IMAP, POP, Exchange ActiveSync. Questi protocolli - come spiegato precedentemente - presentano delle limitazioni che non permettono alcuni cambiamenti:

- Non essendo estendibili, le funzionalità di Carbonio sono vincolate a ciò che questi standard offrono
- Le funzionalità del calendario e dei contatti non sono erogati tramite protocollo specifico, ma tramite delle chiamate non-standard basate su protocollo SOAP
- L'uso di licenze ActiveSync richiede il pagamento di onerosi diritti a Microsoft

Obiettivo

L'obiettivo a cui vogliamo arrivare alla fine di questo capitolato e' capire se ha senso investire tempo e denaro per estendere questo standard in Carbonio, mantenendo i vecchi standard per compatibilità con i client attualmente supportati ma permettendoci di espandere le funzionalità offerte ai client di nuova generazione, offrendo anche uno standard implementativo alla versione Open Source di Carbonio.

Per raggiungere questo obiettivo, ci aspettiamo lo sviluppo di un servizio di demo testabile con uno dei client pubblicizzati nella pagina <https://jmap.io/software.html> che copra i seguenti requisiti funzionali e tecnici. Questo servizio verra' poi utilizzato internamente a Zextras (assieme agli stress test che opzionalmente verranno forniti dal gruppo) per valutare le prestazioni, la manutenibilità e la completezza del protocollo JMAP, paragonandolo agli attuali protocolli attualmente sviluppati in Carbonio.

Requisiti Funzionali

- l'invio e la ricezione di una mail
- la gestione delle cartelle
- la gestione dei contenuti di una cartella
- l'eliminazione di un oggetto
- l'eliminazione di una cartella
- la condivisione di una cartella
- l'eliminazione di una condivisione di cartella
- [Opzionale] l'implementazione di un sistema di sincronizzazione che permetta ad un client di mantenersi aggiornato con gli ultimi aggiornamenti della casella di posta visualizzata (contenente anche Calendari e Rubriche contatti)
- [Opzionale] Implementazione degli stessi requisiti funzionali sopra elencati per calendari e appuntamenti
- [Opzionale] Implementazione degli stessi requisiti funzionali sopra elencati per contatti e rubriche contatti

Requisiti Tecnici

- l'utilizzo di un linguaggio a scelta del gruppo di lavoro, con preferenza per Java (essendo il linguaggio principale dello stack tecnologico di Carbonio)
- l'utilizzo della libreria [INPUTmice/jmap](#) per l'implementazione del protocollo JMAP. Alternativamente nel caso si scelga di optare per un linguaggio diverso, utilizzare l'implementazione presente in uno dei progetti elencati qui [JMAP Software Implementations](#) per sviluppare la propria soluzione

- Il servizio sviluppato deve essere eseguibile in un sistema container, come Docker. Questo ci permette di poter lanciare in maniera molto semplice una o più istanze del servizio, permettendoci di eseguire in batteria test di funzionalità e di performance
- Il servizio sviluppato deve essere scalabile mediante l'inizializzazione di più nodi *stateless*. Per *stateless* si intende che alla richiesta di uno specifico client fatta ad un'architettura contenente più di un'istanza del servizio dato, può rispondere una qualsiasi istanza del servizio, perché nessuna istanza contiene dati specifici di stato rispetto alle richieste dei client
- [Opzionale] aggiungere stress test che riescano a misurare le performance della soluzione per poter poi paragonare la soluzione che utilizza questa tecnologia al nostro attuale prodotto

E' preferibile coprire - anche parzialmente - il requisito opzionale riguardante gli stress test rispetto a coprire tutti i requisiti funzionali.

Link utili:

HomePage JMAP: [JSON Meta Application Protocol Specification \(JMAP\)](#)

Documentazione JMAP mail: [JMAP Mail Specification](#)

Documentazione JMAP Calendar (in via di sviluppo): [JMAP Calendars Specification](#)

Documentazione JMAP Contacts (in via di sviluppo): [JMAP Contacts Specification](#)

Libreria per l'implementazione di un servizio java: [INPUTmice/jmap](#)

Libreria per testare un servizio che implementa JMAP:

Lista dei client, dei servizi backend e dei client attualmente disponibili: [JMAP Software Implementations](#)