



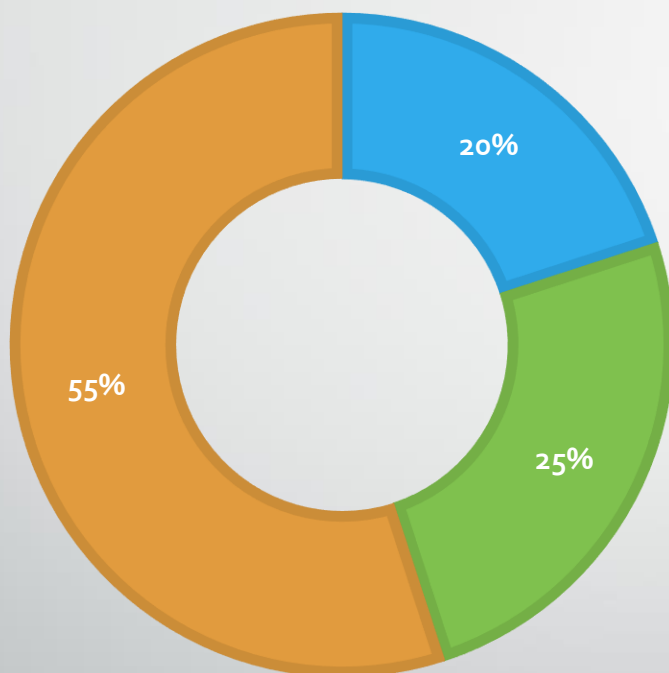
Presentazione Progetto Stage

Progettazione e sviluppo di un sistema di reportistica aziendale interno

Descrizione progetto

- **Obiettivo progetto:** analisi, progettazione e sviluppo di un sistema di consuntivazione delle ore lavorate dai dipendenti dell'azienda.
- **Requisiti:** il sistema deve tenere traccia delle ore giornaliere erogate da ciascun dipendente e dei relativi progetti sui quali sono stati impegnati. Devono essere realizzate le sezioni dedicate alla consuntivazione e quelle dedicate al monitoraggio delle attività. I dati devono poter essere aggregati per singolo dipendente, per team e per progetto, consentendo l'analisi dei costi di progetto.
- **Strumenti:** HTML, CSS, Bootstrap, TypeScript, Javascript (framework Angular), Java (framework Spring Boot), MongoDB.

Progettazione



Definizione strutture
database

Sviluppo back-end

Progettazione front-
end

Pro e contro di MongoDB

Contro:

- Duplicazione dei dati
- Difficoltà di gestione di query complesse
- Possibilità di inconsistenze

Pro:

- Buona scalabilità su grandi moli di dati
- Minor numero di interrogazioni
- Performance ottimizzata
- Struttura schema non necessaria

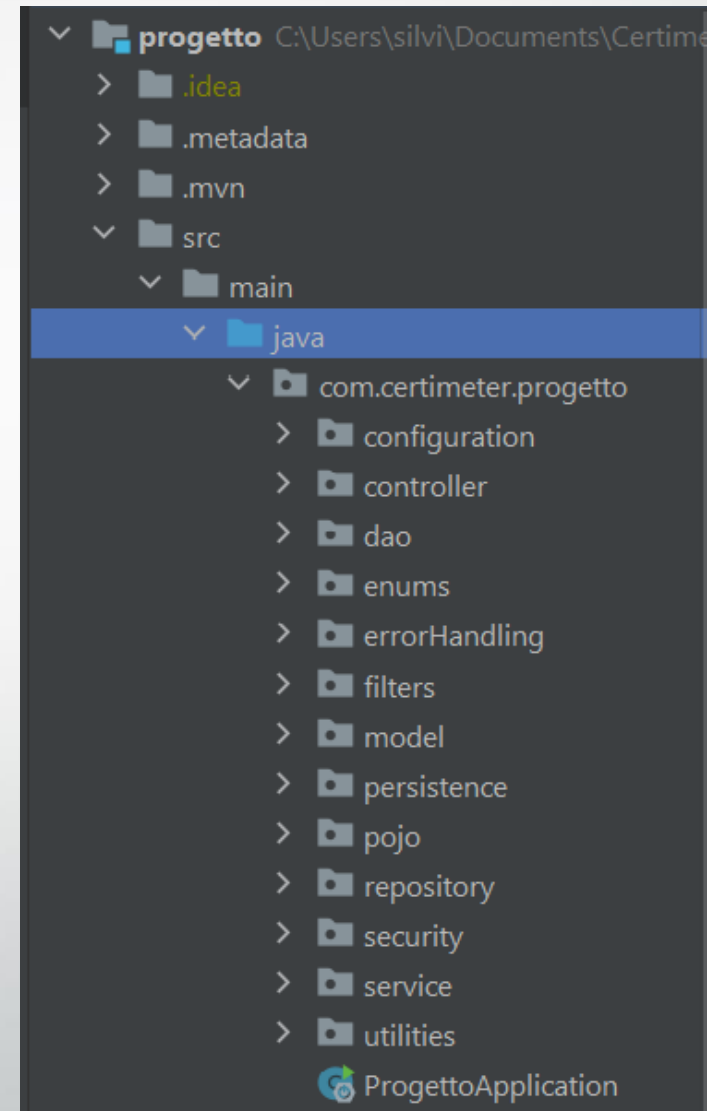
Collezioni DB utilizzate

- **MACRO:** la struttura consente di individuare la totalità delle informazioni riguardanti le macro. I riferimenti ai documenti di altre collezioni sono stati espressi con il solo reference-id al documento stesso. All'interno dei documenti Macro si trovano inoltre tutte le informazioni riguardanti le sue attività e le rispettive sotto attività.
- **REPORTS:** rappresentazione del lavoro svolto dal dipendente. All'interno del documento Report sono indicati i minuti lavorati in una specifica macro e/o attività/sotto attività.
- **USERS:** rappresentazione completa degli utenti. Oltre a semplici dati anagrafici, all'interno del documento User si trovano le credenziali di accesso personali e i ruoli correlati.

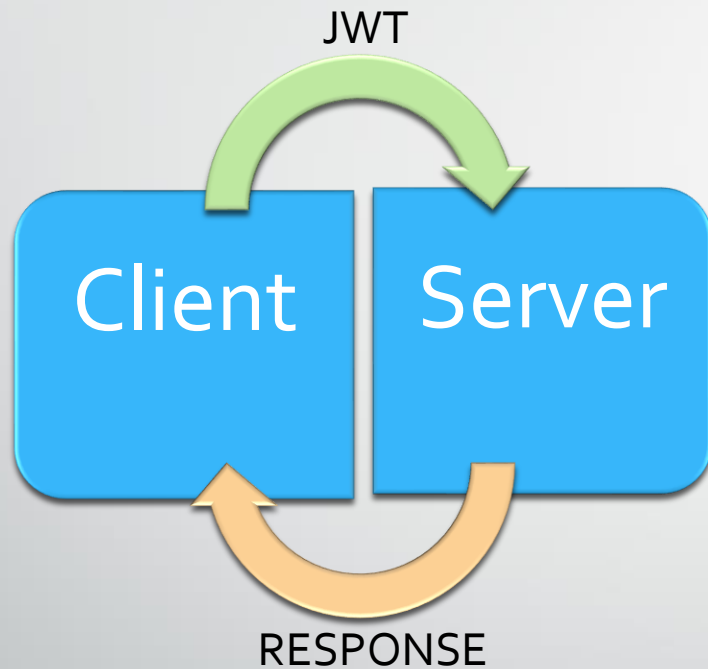


Back-end

- Suddivisione a livelli dei modelli
- Pacchetti strutturati per funzionalità
- Libreria MongoRepository a supporto delle interrogazioni al DB
- Sicurezza gestita con JWT e ruoli utente



Sicurezza con JWT



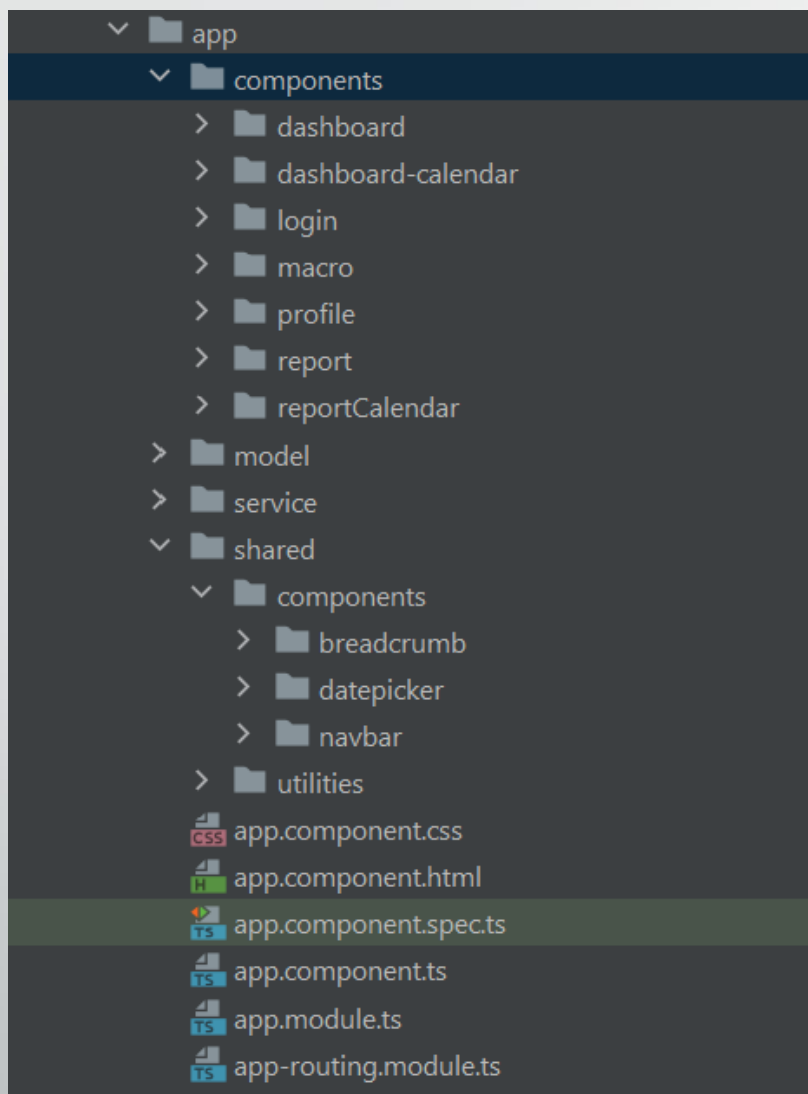
E' stato utilizzato il JSON Web Token come forma di autenticazione successiva al login con inserimento delle credenziali. Il JWT, compatto, completo e autovalidante rende sicure le chiamate al server, verificandone sempre la corretta origine.

Come ulteriore misura di sicurezza, oltre al JWT di accesso (con scadenza di pochi minuti) ne viene generato uno di refresh (che ha scadenza di più lungo respiro): in questo modo l'utente potrà ottenere un nuovo token di accesso senza dover immettere nuovamente le credenziali.

Struttura a livelli

- Per gestire le trasformazioni necessarie ai dati durante il passaggio dal DB al front-end, il back-end sfrutta una struttura a tre livelli: model, pojo e dao.
- Durante i passaggi dall'uno all'altro modello i dati possono essere trasformati per essere il più vicini possibile alle strutture del DB o ai consumatori degli stessi (front-end).
- Un'ulteriore strutturazione a livelli la si trova nella parte di gestione delle richieste. Queste ultime vengono catturate da un controller, gestite da un service e consumate da un repository. In questo modo durante i passaggi da un livello all'altro vengono fatte le giuste verifiche sulle richieste e vengono richiamate le conversioni dei modelli di dati.

Front-End



- I modelli utilizzati rispecchiano quelli in arrivo dal back-end
- Un interceptor di autorizzazione si occupa dell'inserimento del JWT ad ogni richiesta al server
- Il JWT viene conservato nella sessionstorage del client browser