

# DOCUMENTAZIONE NOTO MORENO ESERCITAZIONE FINALE TRI-TECH ACADEMY

Nell'esercitazione assegnatami veniva richiesta la creazione di un software di gestione prenotazioni per le sale conferenze di una determinata azienda. Per svilupparla mi sono avvalso degli strumenti indicati come Angular, Spring Boot, SQL Developer e PowerBI.

Partendo dalla sezione DB:

Ho creato quattro tabelle fondamentali: utenti, sedi, sale e prenotazioni (reperibili all'interno della connessione condivisa tutte precedute dal mio cognome. e.g. "notoprenotazioni").

Fra le varie tabelle ho creato, dove necessario, dei vincoli che potessero metterle in comunicazione.

In particolare, fra sedi e sale esiste una relazione di tipo uno a molti, una sede può possedere più stanze al suo interno ma, ovviamente, una stanza può appartenere solo ad una sede.

Altri vincoli sono presenti tra le tabelle stanze e utenti con la tabella prenotazioni.

All'interno della prenotazione sono infatti presenti la stanza a cui questa si riferisce e l'utente che l'ha effettuata.

Dunque, anche in questo caso, sia utenti che stanze hanno nei confronti della tabella prenotazioni una relazione di uno a molti, infatti un utente può effettuare più prenotazioni e la stessa stanza può essere prenotata più volte, mentre una prenotazione può riguardare solo un utente e una stanza.

Fra le altre cose che è necessario specificare, per quanto riguarda la sezione DB, è la presenza, all'interno della tabella utenti, del campo password. Ovviamente questa è una pratica scorretta, poiché le password di ogni utente andrebbero conservate in un cloud virtuale e soprattutto criptate, tuttavia, per esigenze di livello temporale sono stato costretto ad inserirle all'interno della stessa tabella.

BE:

L'ambiente DB, comunque, è collegato a quello BE, gestito tramite Spring Boot.

All'interno ho definito più entità, ovvero Booking, Room, Site, User, che corrispondono a quelle presenti all'interno del DB e a cui, effettivamente si riferiscono, ma ne è presente anche un'altra, ovvero Reservation. Non era un'istanza richiesta, tuttavia è stata necessaria nell'estrazione delle date dalle prenotazioni già esistenti, e si è rivelata fondamentale nello sviluppo di un punto chiave del progetto: la gestione della sovrapposizione delle prenotazioni.

Mediante questo controllo è stato possibile confrontare le date della nuova prenotazione in entrata con quelle già presenti relative alla stessa stanza. È stato così possibile bloccare la creazione di nuove prenotazioni che potessero essere in contrasto con quelle già originate. Il metodo in questione si chiama "checkValidityDate" ed è possibile trovarlo nell'implementazione del service delle prenotazioni. Altra particolarità è poi la differente gestione delle prenotazioni a seconda che queste fossero aggiunte o modifiche di prenotazioni già esistenti, infatti nel caso dell'update sarà possibile registrare la prenotazione per lo stesso orario in cui era già fissata, o, eventualmente, espanderla o rimpicciolirla, senza ovviamente andare ad intaccare prenotazioni differenti da quella presa in esame.

Sono poi presenti anche i CRUD per ognuna delle 4 entità presenti nel DB, accompagnati, in alcuni casi, da particolari richieste al DB a seconda delle necessità riscontrate in corso d'opera, le 4 repositories, i vari services e i 4 controllers.

Oltre questi, è presente un controllo dell'identità dell'utente che prova ad accedere al software.

Viene effettuato facendo un confronto, prima dell'esistenza dell'individuo nel DB tramite email e, successivamente, verificando che la password inserita sia corretta, garantendo dunque l'accesso o inviando un messaggio di errore.

Ho poi sviluppato, come richiesto nei punti bonus un sistema per l'invio di mail all'aggiunta o all'updating di prenotazioni già esistenti attraverso il servizio SpringBoot Starter Mail, che mi ha permesso di generare, dopo la necessaria preparazione sull'application properties, un servizio che si occupa, appunto, dell'invio mail.

Su questo punto: il servizio è stato testato con una mail personale, che su consiglio di uno dei formatori sarà rimossa al momento della consegna. Lascero comunque inserito il protocollo mail necessario per l'utilizzo di domini legati a Google che ho utilizzato, consiglio inoltre di settare nuovamente sull'EmailSenderService indirizzo di partenza e di arrivo della comunicazione, i quali sono rispettivamente il mio indirizzo personale e l'indirizzo mail, preso dinamicamente, dell'utente che effettua la prenotazione.

Infine ho realizzato, all'interno del controller di booking, un endpoint che permetta la generazione su file csv dell'elenco delle prenotazioni tramite OpenCSV.

Poichè alcuni degli attributi dell'entità prenotazione erano essi stessi altre entità ho deciso di effettuare una formattazione della prenotazione prima di essere mandata in stampa.

Dei campi in questione avevo scelto, inizialmente, di estrarre il nome, tuttavia, nella possibilità futura di arricchimento dei DBs, per evitare problemi di omonimia delle stanze o degli utenti, a danno di una meno immediata comprensione dei dati, ho scelto l'utilizzo dell'id di quei determinati campi.

FE:

Infine tutte le logiche sviluppate sono state collegate alla sezione FE sviluppata tramite Angular.

In questa sezione ho cercato di realizzare in primis tutti i punti richiesti utilizzando tecnologie e librerie già usate come, ad esempio, moment.js, o importandone di nuove come bootstrap e angular Material.

Anche a livello di funzionalità ho cercato di mantenermi in linea con quanto visto ed utilizzato durante le nostre lezioni, aggiungendo, dove necessario, qualche nuova implementazione. Alcuni esempi potrebbero essere la guard, che mi ha permesso di impedire l'accesso alle varie sezioni del programma agli utenti non loggati, oppure i local storage che mi hanno permesso di archiviare alcune delle scelte effettuate dall'utente durante l'uso del programma, non costringendolo a dover rifare il login o a ripartire dalla scelta della sede nel caso di una nuova prenotazione. Inoltre, in funzione dell'archiviazione del login, considerando che tutto ciò che viene salvato in local storage è accessibile da parte dell'utente direttamente dal browser, e consapevole del problema della password presente nell'interfaccia di definizione dell'utente stesso, ho deciso di criptare, e successivamente decrittare, i dati mediante le funzioni atob e btoa, che permettono la codifica e decodifica in base64.

Fra le altre cose ho implementato, per ogni stanza, un'agenda all'interno della quale vengono mostrate le prenotazioni relative a quel giorno, disposte in ordine cronologico. È possibile modificare o eliminare una prenotazione anche passata ma è possibile aggiungerne solo dalla data odierna in poi.

Inoltre tutti possono aggiungerne, ma solo gli admin possono andare a modificare o eliminare tutte le prenotazioni. Gli user, invece, possono modificare o eliminare solo quelle create da essi stessi.

I controlli delle date sono stati riportati anche sul FE, e viene inviato un determinato errore a seconda del tipo d'invalidità della data (stanza già occupata o prenotazione impossibile (indietro nel tempo, o fine prima dell'inizio)). Per gli altri campi è richiesta la presenza di tutti ad esclusione delle necessità tecniche.

Infine, anche a FE è stata implementata la possibilità di download di tutte le prenotazioni su file csv, mediante il comando che compare nell'header in alto a destra dopo il login.

In generale ho sviluppato componenti per la creazione, l'update, la visualizzazione dettagliata del post, del calendario (agenda) virtuale e scelta di sedi e stanze, e infine un home page dalla quale è possibile effettuare log in e log out.

Per ognuna delle entità presenti a BE (compresa Reservation) è stata creata un'interfaccia. Sono presenti anche service per ognuna delle 4 entità fondamentali più uno generale, chiamato main, all'interno del quale sono contenuti i metodi che permettono a componenti differenti di comunicare.

Infine, per la gestione dei numerosi endpoint di collegamento al BE, ho utilizzato un enum.

PBI:

Su powerBI ho effettuato un report dell'andamento settimanale delle prenotazioni sale all'interno dell'azienda, ho perciò fatto in modo, dopo aver creato una nuova colonna con un formato data valido su powerBI, che venisse estratta la weekOfTheYear delle prenotazioni, specificando che la nostra tipologia di settimana parte da lunedì.

È perciò stato generato un grafico su due colonne, relative alle due settimane sulle quali sono disposte tutte le prenotazioni presenti in DB.

Nel caso in cui, dunque, nel FE, non venissero trovate prenotazioni per una particolare stanza, basterà andare poco indietro nel tempo per vederle spuntare, infatti, per una questione di correttezza logica, ho fatto in modo di lasciare in archivio solo le prenotazioni sulle quali era stato generato il report di PBI.

Spero che il lavoro svolto, compresa questa documentazione, possa essere all'altezza delle aspettative e del lavoro svolto insieme durante questi mesi.

Grazie mille.

Noto Moreno