Università degli Studi di Napoli Federico II
Esame di Advanced Computer Programming

Proff. De Simone, Della Corte

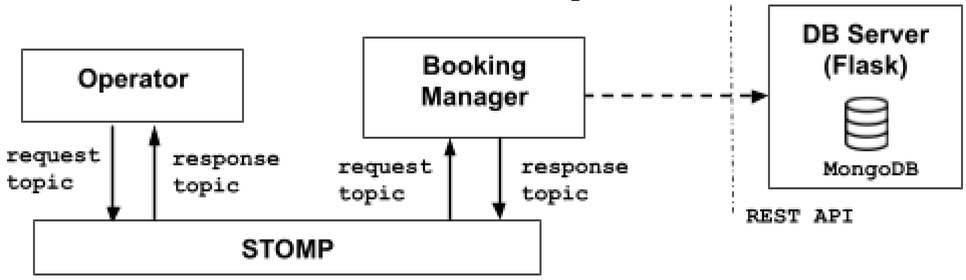
Prova pratica del giorno 26/07/2024
Durata della prova: 120 minuti

°°

Lo studente legga attentamente il testo e produca il programma ed i casi di test necessari per dimostrarne il funzionamento. Al termine della prova lo studente dovrà far verificare il funzionamento del programma ad un membro della Commissione.

°°

Testo della prova



Il candidato implementi un sistema distribuito in **Python** per la prenotazione di hotel basato su **STOMP**, **Flask** e **MongoDB**. Il sistema è caratterizzato dai seguenti componenti.

Operator. E' un operatore che può effettuare richieste di creazione prenotazione ed applicazione sconto ad alcune prenotazioni (in base all'operatore ed al numero di notti), verso il **Booking Manager**. L'invio di una richiesta consiste nella creazione di un *frame STOMP* nel quale inserire una stringa che concateni il *tipo di richiesta* ed i relativi *parametri*. Le richieste sono inviate sul *topic request*.

- Per creare una prenotazione, l'Operator invia una richiesta di tipo CREATE con i seguenti attributi: *client* (username del cliente), *hotel* (nome dell'hotel), *operator* (username dell'operatore), *nights* (numero di notti), *people* (numero di persone), *cost* (costo in Euro)
- Per applicare uno sconto alle prenotazioni create da un operatore e con almeno un certo numero di notti, l'Operator invia una richiesta di tipo UPDATE con i seguenti attributi: *discount* (sconto in Euro da applicare), *operator* (operatore), *nights* (numero di notti). N.B.: *operator* e *nights* rappresentano i criteri per l'applicabilità dello sconto.

Le risposte a tali richieste saranno ricevute in maniera asincrona sul *topic response*. Una volta ricevuta una risposta, l'Operator mostra a video il contenuto della risposta.
Operator avvia 6 thread: i primi 4 thread generano una richiesta CREATE, mentre i successivi 2 generano una richiesta UPDATE. N.B.: i valori dei campi delle richieste possono essere scelti in maniera casuale, tranne quello *operator* che dovrà essere quello passato da riga di comando.

Booking Manager. E' un manager delle prenotazioni che riceve le richieste da un Operator in maniera asincrona attraverso il topic *request*. Ricevuta una richiesta, il Booking Manager analizza il frame STOMP, ed estrae il tipo di richiesta, i.e., CREATE o UPDATE, ed i relativi parametri. In base al tipo di richiesta, il Booking Manager genera una richiesta verso uno degli endpoint esposti dal DB Server.

- Nel caso di CREATE, la richiesta dovrà prevedere nel body i parametri ricevuti dall'Operator in formato json, e.g., {"client":"ILoveTravel", "hotel":"Vesuvio", "operator":"operator1", "nights":3, "people":2, "cost": "200"}.
- Nel caso di UPDATE, la richiesta dovrà prevedere nel body i parametri ricevuti dall'operator in formato json, e.g., {"operator":"operator1", "nights":3, "discount":50}.

Inviata una richiesta al DB Server, il Booking Manager si mette in attesa della risposta, che invierà poi all'Operator attraverso un frame STOMP sul *topic response*.

DB Server. Implementa un server Flask che gestisce una collection MongoDB contenente i dati delle prenotazioni, e che espone una REST API con due endpoint:

- Un endpoint per la creazione di una prenotazione che, ricevuta una richiesta, crea un document all'interno di una collection MongoDB che contenga tutti i campi della prenotazione.
- Un endpoint per l'applicazione di uno sconto ad alcune prenotazioni che, ricevuta una richiesta, cerca nella collection MongoDB tutte le prenotazioni create dall'operatore specificato nel body richiesta e che abbiano almeno un numero di notti maggiore o uguale al valore specificato nel body della richiesta. Per tutte le prenotazioni trovate, il costo della prenotazione viene aggiornato sottraendo al costo attuale il discount ricevuto nel body della richiesta (impostando a zero il costo, se la sottrazione porta ad un risultato negativo).

Lo studente progetti la REST API, scegliendo opportunamente i metodi HTTP da utilizzare.