Flight Reservation Application

documentație de proiect

Swagger Doc: <http://[HOST]/booking/swagger-ui.html#/>

Aplicația dezvoltată își propune să permită utilizatorului, după crearea/logarea într-un cont al aplicației, rezervarea unui zbor cu o anumită cursă aeriană/într-o anumită zi.

Pachetul **model** cuprinde entitățile care numesc tabelele de date ale aplicației – *Flight, Passenger, Reservation, User, Role.*

**Flight{**

|  |  |
| --- | --- |
| **arrivalCity** | **string** |
| **dateOfDeparture** | **string($date-time)** |
| **departureCity** | **string** |
| **estimatedDepartureTime** | **Timestamp{**   |  |  | | --- | --- | | **date** | **integer($int32)** | | **day** | **integer($int32)** | | **hours** | **integer($int32)** | | **minutes** | **integer($int32)** | | **month** | **integer($int32)** | | **nanos** | **integer($int32)** | | **seconds** | **integer($int32)** | | **time** | **integer($int64)** | | **timezoneOffset** | **integer($int32)** | | **year** | **integer($int32)** |   **}** |
| **flightNumber** | **string** |
| **id** | **integer($int64)** |
| **operatingAirlines** | **string** |

**}**

**Passenger{**

|  |  |
| --- | --- |
| **email** | **string** |
| **firstName** | **string** |
| **id** | **integer($int64)** |
| **lastName** | **string** |
| **middleName** | **string** |
| **phone** | **string** |

**}**

**Reservation{**

|  |  |
| --- | --- |
| **checkedIn** | **boolean** |
| **flight** | **Flight{**   |  |  | | --- | --- | | **arrivalCity** | **string** | | **dateOfDeparture** | **string($date-time)** | | **departureCity** | **string** | | **estimatedDepartureTime** | **Timestamp{...}** | | **flightNumber** | **string** | | **id** | **integer($int64)** | | **operatingAirlines** | **string** |   **}** |
| **id** | **integer($int64)** |
| **numberOfBags** | **integer($int32)** |
| **passenger** | **Passenger{**   |  |  | | --- | --- | | **email** | **string** | | **firstName** | **string** | | **id** | **integer($int64)** | | **lastName** | **string** | | **middleName** | **string** | | **phone** | **string** |   **}** |

**}**

Pachetul **service** salvează *business logic*-ul aplicației – realizarea unei rezervări și salvarea acesteia, partea de autentificare a utilizatorului și partea de reținere a detaliilor de autentificare. În **repository** se apelează la *JpaRepository* prin extinderea acestuia in fiecare interfață specifică entității din baza de date.

**Controller** conține *end-point­*-urile fiecărui serviciu:

* pentru zboruri (căutare zbor, adăugare zbor)
* pentru *user*i ( înregistrare, autentificare, redirecționare *post-authentication*)
* pentru rezervări - un *controller* normal și unul *REST* ( completare rezervare, confirmare rezervare, căutare rezervare, modificare rezervare)

**Funcționalitate** – utilizatorul este întâmpinat pe pagina de *home* în care are opțiunea de a se loga/de a își crea cont. După ce trece de acest pas, este direcționat către pagina de *search* a zborurilor în funcție de orașele plecare/destinație și în funcție de ziua în care e planificat zborul. Introducând parametrii, mai departe îi vor fi afișate opțiunile care sunt *available* conform căutării și clientul își alege zborul la care dorește să facă rezervarea. Pe pagina următoare, se vor completa detaliile rezervării, anume telefon, nume, email, împreună cu detaliile de tranzacționare. Dacă totul este în regulă, rezervarea este făcută cu succes, iar utilizatorul este înștiințat de acest fapt prin afișarea unui mesaj.

Demonstrarea funcționalității este ilustrată atât prin GUI – *Thymeleaf*, cât și *Postman* pentru paginile care nu au interfață.