🚌

Bus Ticket Booking Portal

Objective

> 📌 The Bus Ticket Booking Portal is a comprehensive platform that facilitates seamless bus ticket booking for travelers and allows bus agencies to efficiently manage and schedule their trips. The portal provides a user-friendly interface for both agencies and customers, streamlining the entire booking process.

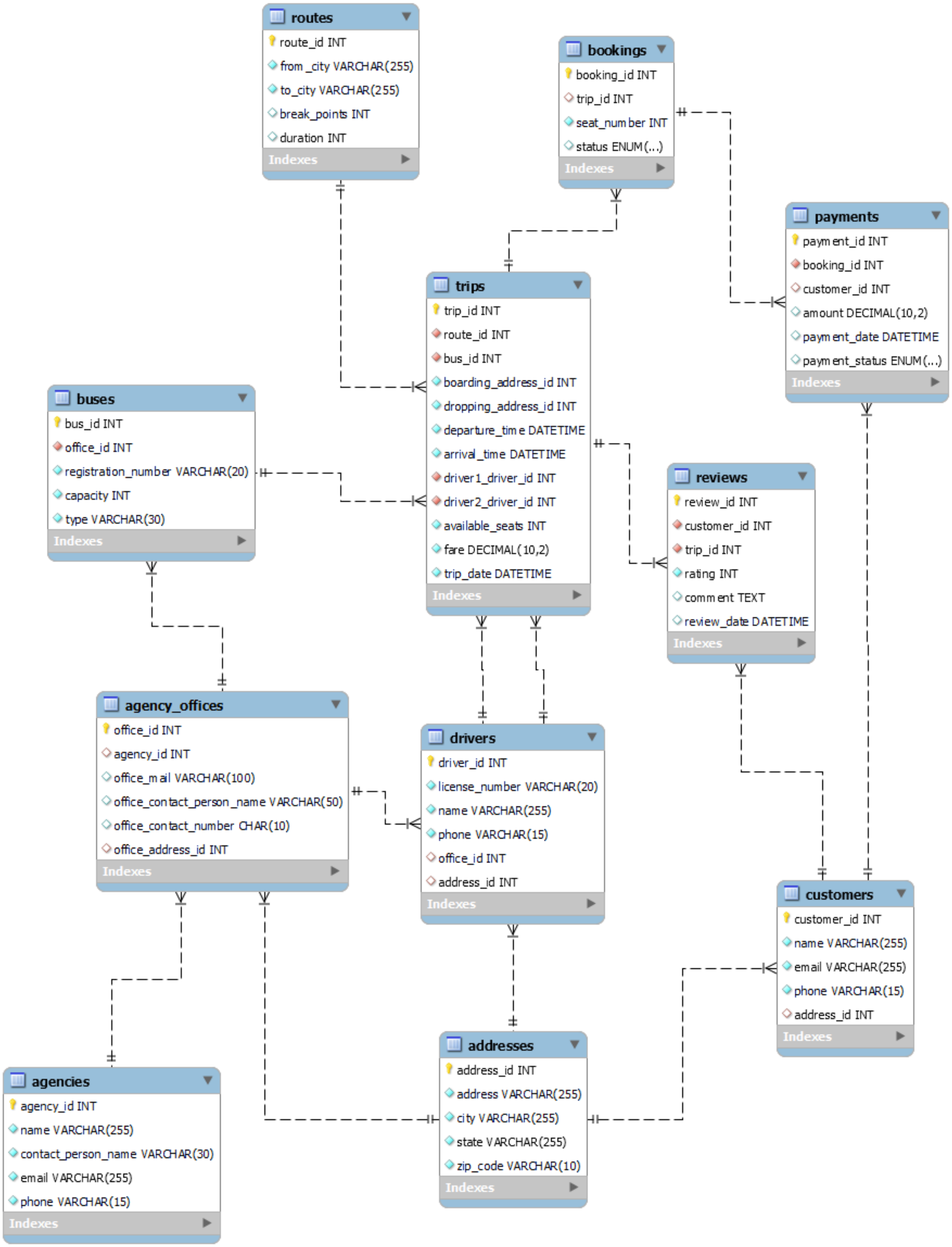
Design guidelines:

- All Rest API request and response format must be in JSON.
- 200, 201, 400, 404, 401, 500, 505 – use appropriate http status code in your response object.
- Handle the Rest exceptions globally using Rest exception handler for all endpoints.
- Sample Error Responses are shown in below table, participants to identify error paths for all the Endpoints.
- Test your Rest API using Junit and Mockito.
- Participants should not alter database schema under any circumstances except wherever changes are mentioned explicitly.
- Add separate tables for user authentication and authorization module.

Database schema:

▼ ER Diagram

busticketbookingER.mwb.bak



▼ MySQL script files

create.sql

insert.sql

Modules

Endpoints

Path URL	HTTP Methods	Description	Success Response
/api/films/post	POST	Add new Film object in DB	String: Record Created Successfully
/api/films/title/{title}	GET	Search Films by Title	Collection of Films
/api/films/year/{year}	GET	Search Films by Release Year	Collection of Films
/api/films/duration/gt/{rd}	GET	Search Films where Rental Duration is greater than {rd}	Collection of Films
/api/films/rate/gt/{rate}	GET	Search Films where Rental Rate is greater than {rate}	Collection of Films
/api/films/length/gt/{length}	GET	Search Films where Length is greater than {length}	Collection of Films
/api/films/duration/lt/{rd}	GET	Search Films where Rental	Collection of Films

Path URL	HTTP Methods	Description	✓ Success Response
		Duration is lower than {rd}	
/api/films/rate/lt/{rate}	GET	Search Films where Rental Rate is lower than {rate}	Collection of Films
/api/films/length/lt/{length}	GET	Search Films where Length is lower than {length}	Collection of Films
/api/films/betweenyear/{from}/{to}	GET	Search Films which are Released between {from} and {to}	Collection of Films
/api/films/rating/lt/{rating}	GET	Search Films where Rating is lower than {rating}	Collection of Films
/api/films/rating/gt/{rating}	GET	Search Films where Rating is greater than {rating}	Collection of Films
/api/films/language/{lang}	GET	Search Films where Language is {lang}	Collection of Films
/api/films/countbyyear	GET	Display number of Films released by each Year	Year & Count of Films
/api/films/{id}/actors	GET	Find all Actors of a Film by Film id	Collection of Actors
/api/films/category/{category}	GET	Find all Films of specified {category}	Collection of Films

Path URL	HTTP Methods	Description	Success Response
/api/films/{id}/actor	PUT	Assign Actor to a Film	Collection of Actors
/api/films/update/title/{id}	PUT	Update Title of a Film	Film
/api/films/update/releaseyear/{id}	PUT	Update Release Year of a Film	Film
/api/films/update/rentaldurtion/{id}	PUT	Update Rental Duration of a Film	Film
/api/films/update/rentalrate/{id}	PUT	Update Rental Rate of a Film	Film
/api/films/update/rating/{id}	PUT	Update Rating of a Film	Film
/api/films/update/language/{id}	PUT	Update Language of a Film	Film
/api/films/update/category/{id}	PUT	Update Category of a Film	Film
/api/actors/post	POST	Add new Actor object in DB	String: Record Created Successfully
/api/actors/lastname/{ln}	GET	Search Actors by Last Name	Collection of Actors
/api/actors/firstname/{fn}	GET	Search Actors by First Name	Collection of Actors
/api/actors/update/lastname/{id}	PUT	Update Last Name of an Actor	Actor
/api/actors/update/firstname/{id}	PUT	Update First Name of an Actor	Actor
/api/actors/{id}/films	GET	Search Films of an Actor by Actor Id	Collection of Films

Path URL	HTTP Methods	Description	✓ Success Response
/api/actors/{id}/film	PUT	Assign Film to an Actor	Collection of Films
/api/actors/toptenbyfilmcount	GET	Find top 10 Actors by Film Count	Id & Name of Actors with Number of Films
/api/customers/post	POST	Add new Customer Object in DB	String: Record Created Successfully
/api/customers/lastname/{ln}	GET	Search Customers by last name	Collection of Customers
/api/customers/firstname/{fn}	GET	Search Customers by first name	Collection of Customers
/api/customers/email/{email}	GET	Search Customers by email	Customer
/api/customers/{id}/{addressId}	PUT	Assign Address to a Customer	Customer & Address details
/api/customers/city/{city)	GET	Search Customers by City	Collection of Customers with Address
/api/customers/country/{country}	GET	Search Customers by Country	Collection of Customers with Address
/api/customers/active	GET	Search all active Customers	Collection of Customers

Path URL	HTTP Methods	Description	Success Response
/api/customers/inactive	GET	Search all inactive Customers	Collection of Customers
/api/customers/phone/{phone}	GET	Search Customers by phone number	Customer with Address
/api/customers/update/{id}/{fn}	PUT	Update first name of Customer.	Customer
/api/customers/update/{id}/ln	PUT	Update last name of Customer	Customer
/api/customers/update/{id}/email	PUT	Update email of Customer	Customer
/api/customers/update/{id}/store	PUT	Assign Store to a Customer	Customer
/api/customers/update/{id}/phone	PUT	Update phone number of a Customer	Customer
/api/staff/post	POST	Add new Staff object to DB	String: Record Created Successfully
/api/staff/lastname/{ln}	GET	Search Staff by last name	Collection of Staff
/api/staff/firstname/{fn}	GET	Search Staff by first name	Collection of Staff
/api/staff/email/{email}	GET	Search Staff by email	Staff
/api/Staff/{id}/address	PUT	Assign Address to a Staff	Staff & Address details

Path URL	HTTP Methods	Description	Success Response
/api/staff/city/{city}	GET	Search Staff by City	Collection of Staff with Address
/api/staff/country/{country}	GET	Search Customers by Country	Collection of Staff with Address
/api/staff/phone/{phone}	GET	Search Staff by phone number	Staff with Address
/api/staff/update/fn/{id}	PUT	Update first name of Staff	Staff
/api/staff/update/ln/{id}	PUT	Update last name of Staff	Staff
/api/staff/update/email/{id}	PUT	Update email of Staff	Staff
/api/staff/update/store/{id}	PUT	Assign Store to a Staff	Staff
/api/staff/update/phone/{id}	PUT	Update phone number of a Staff	Staff
/api/store/post	POST	Add new Store object to DB	String: Record Created Successfully
/api/store/{storeId}/address/{addressId}	PUT	Assign Address to a Store	Store & Address details
/api/store/country/{country}	GET	Search Store by Country	Collection of Store with Address
/api/store/city/{city}	GET	Search Store by City	Collection of Store with

Path URL	HTTP Methods	Description	Success Response
			Address
/api/store/phone/{phone}	GET	Search Store by phone number	Store with Address
/api/store/update/{storeId}/{phone}	PUT	Update phone number of a Store	Store
/api/store/{storeId}/manager/{manager_staff_id}	PUT	Assign manager to a Store	Store & Address details
/api/store/staff/{storeId}	GET	Display all Staff of a Store	Collection of Staff
/api/store/customer/{storeId}	GET	Display all Customers of a Store	Collection of Customers
/api/store/manager/{storeId}	GET	Display Manager Details of a Store	Staff
/api/store/managers	GET	Display Manager details (first name, last name, email, phone) and Store details (Address, City, Phone) of all stores	Custom Object
/api/inventory/add	POST	Add one more Film to a Store	String: Record Created Successfully
/api/inventory/films	GET	Display inventory(count) of all Films in all Stores	Collection of Film title, no. of copies

Path URL	HTTP Methods	Description	Success Response
/api/inventory/store/{id}	GET	Display inventory of all Films by a Store	Collection of Film title, no. of copies
/api/inventory/film/{id}	GET	Display inventory(count) of a Film in all Stores	Collection of Store address, no. of copies
/api/inventory/film/{id}/store/{id}	GET	Display inventory(count) of a Film in a Store	Store Address, no. of copies
/api/rental/add	POST	Rent a Film	String: Record Created Successfully
/api/rental/customer/{id}	GET	Display all Films rented to a customer	Collection of Films
/api/rental/toptenfilms	GET	Display top 10 most rented Films	Collection of Films
/api/rental/toptenfilms/store/{id}	GET	Display top 10 most rented Films of a Store	Collection of Films
/api/rental/due/store/{id}	GET	Display list of Customers who have not yet returned the Film	Collection of Customers
/api/rental/update/returndate/{id}	POST	Update return date	Rental
/api/payment/add	PUT	Make one Payment entry	String: Record Created Successfully

Path URL	HTTP Methods	Description	Success Response
/api/payment/revenue/datewise	GET	Calculate Cumulative revenue of all Stores	Collection of payment_date, amount
/api/payment/revenue/datewise/store/{id}	GET	Calculate Cumulative revenue of a Store	Collection of payment_date, amount
/api/payment/revenue/filmwise/	GET	Calculate Cumulative revenue of all Films (across all stores)	Collection of Film title, amount
/api/payment/revenue/film/{id}	GET	Calculate Cumulative revenue of a Film Store wise	Collection of Store address, amount
/api/payment/revenue/films/store/{id}	GET	Calculate Cumulative revenue of all Films by Store	Collection Film title, amount

Endpoints

Endpoints (1)

Path URL	HTTP Methods	Description	Success Response
/api/agencies	POST	Add new Agency details in DB.	String: Record Created Successfully

Path URL	HTTP Methods	Description	Success Response
/api/agencies/{agency_id}	GET	Search Agency details by agency_id	Agency Object
/api/agencies/{agency_id}	PUT	Update Agency details by agency_id	String: Record Updated Successfully.
/api/agencies/{agency_id}	DELETE	Remove Agency details.	String: Record Deleted Successfully.
/api/agencies/offices/{agency_id}	GET	Search all offices of an Agency by agency_id	Collection of offices.
/api/agencies/offices/{office_id}	GET	Search office by office_id	Office object.
/api/agencies/offices/{office_id}	DELETE	Remove office details by office_id.	String: Record Deleted Successfully.
/api/buses	POST	Add new bus details.	String: Record Created Successfully
/api/buses	GET	Search all bus details.	Collection of buses.
/api/buses	PUT	Update bus details.	String: Record Updated Successfully.
/api/buses/{bus_id}	GET	Search bus details by	Bus Object.

Path URL	HTTP Methods	Description	Success Response
		bus_id.	
/api/buses/{bus_id}	DELETE	Remove bus details by bus_id.	String: Record Deleted Successfully.
/api/buses/office/{office_id}	GET	Search all buses details by office_id.	Collection of Buses
/api/customers	POST	Add new Customer Object in DB	String: Record Created Successfully
/api/customers	GET	Search all Customers.	Collection of Customers
/api/customers/{customer_id}	GET	Search Customer by customer_id.	Customer Object.
/api/customers/email/{email}	GET	Search Customers by email.	Collection of Customers.
/api/customers/phone	GET	Search Customers by phone.	Collection of Customers.
/api/customers/city/{city)	GET	Search Customers by City	Collection of Customers.
/api/customers/country/{country}	GET	Search all Customers by Country	Collection of Customers with Address
/api/customers/update/{id}/ln	PUT	Update last name of Customer	Customer

Path URL	HTTP Methods	Description	Success Response
/api/customers/update/{id}/email	PUT	Update email of Customer	Customer
/api/routes	POST	Add new Route.	String: Record Created Successfully
/api/routes	GET	Search all Routes.	Collection of Routes.
/api/routes/{route_id}	GET	Search Route by route_id.	Route Object
/api/routes/fromcity/{from_city}	GET	Search Routes by from_city.	Collection of Routes.
/api/routes/tocity/{to_city}	GET	Search Routes by to_city	Collection of Routes.
/api/routes/{from_city}/{to_city}	GET	Search Route by from_city and to_city.	Route Object.
/api/trips	POST	Add new trip details.	String: Record Created Successfully
/api/trips	GET	Search all trips details.	Collection of trips.
/api/trips/{trip_id}	GET	Search trip details by trip_id.	Trip Object.
/api/trips	PUT	Update trip details.	String: Record Updated Successfully

Path URL	HTTP Methods	Description	✓ Success Response
/api/trips/from_city/{from_city}	GET	Search trips details by from_city.	Collection of trips.
/api/trips/to_city/{to_city}	GET	Search trips details by to_city.	Collection of trips.
/api/trips/bus_type/{type}	GET	Search trips bus type.	Collection trips.
/api/store/bus_type/{type}/trip_date/{trip_date}	GET	Search trips by bus type and trip_date	Collection of Staff
/api/reviews	POST	Add new review.	String: Record Created Successfully
/api/reviews	GET	Search all reviews.	Collection of reviews.
/api/reviews	PUT	Update review details.	String: Record Updated Successfully.
/api/reviews	DELETE	Remove review details.	String: Record Deleted Successfully.
/api/reviews/tripid/{trip_id}	GET	Search reviews by trip_id.	Collection of reviews.
/api/payment	POST	Add new Payment details.	String: Record Created Successfully

Path URL	HTTP Methods	Description	Success Response
/api/payment	GET	Search all payment details.	Collection of payments.
/api/payment/{payment_id}	GET	Search payment details by payment_id.	Payment Object.
/api/payment/customer/{customer_id}	GET	Search payment details by customer_id.	Collection of payments.
/api/payment/agency/{agency_id}	GET	Search payment details by agency_id.	Collection of payments.
/api/payment/office/{office_id}	GET	Search payment details by agency_id.	Collection of payments.
/api/trips/{trip_id}	DELETE	Remove trip details by trip_id.	String: Record Deleted Successfully
/api/reviews/{review_id}	GET	Search review by review_id.	Review object.
Untitled			
/api/reviews/office/{office_id}	GET	Search reviews by office_id.	Collection of reviews.
/api/trips/{from_city}/{to_city}/{trip_date}/{bus_type}	GET	Search trips details by from_city, to_city, trip_date and bus type.	Collection of trips.

Path URL	HTTP Methods	Description	Success Response
/api/reviews/customerid/{customer_id}	GET	Search reviews by customer_id.	Collection of reviews.
/api/trips/{from_city}/{to_city}/{trip_date}	GET	Search trips details by from_city, to_city and trip_date	Collection of trips.
Untitled			
/api/reviews/agency/{agency_id}	GET	Search reviews by agency_id.	Collection of reviews.
/api/agencies/offices/{office_id}	PUT	Update Office details by office_id.	String: Record Updated Successfully.
/api/trips/trip_date/{trip_date}	GET	Search trips details by trip_date	Collection of trips.
/api/reviews/driver/{driver_id}	GET	Search reviews by driver_id.	Collection of reviews.
/api/customers	PUT	Update customer details.	String: Record Updated Successfully
/api/agencies	GET	Search all agencies details.	Collection of agencies.
/api/customers/state/{state)	GET	Search Customer by State.	Collection of Customers.
Untitled			
/api/agencies/offices	GET	Serach all offices.	Collection of offices.

Path URL	HTTP Methods	Description	Success Response
/api/agencies/addoffice	POST	Add new branch office of an agency.	String: Record Created Successfully.
Untitled			
/api/routes	PUT	Update Route details.	String: Record Updated Successfully
/api/routes/{route_id}	DELETE	Remove Route.	String: Record Deleted Successfully.