



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.

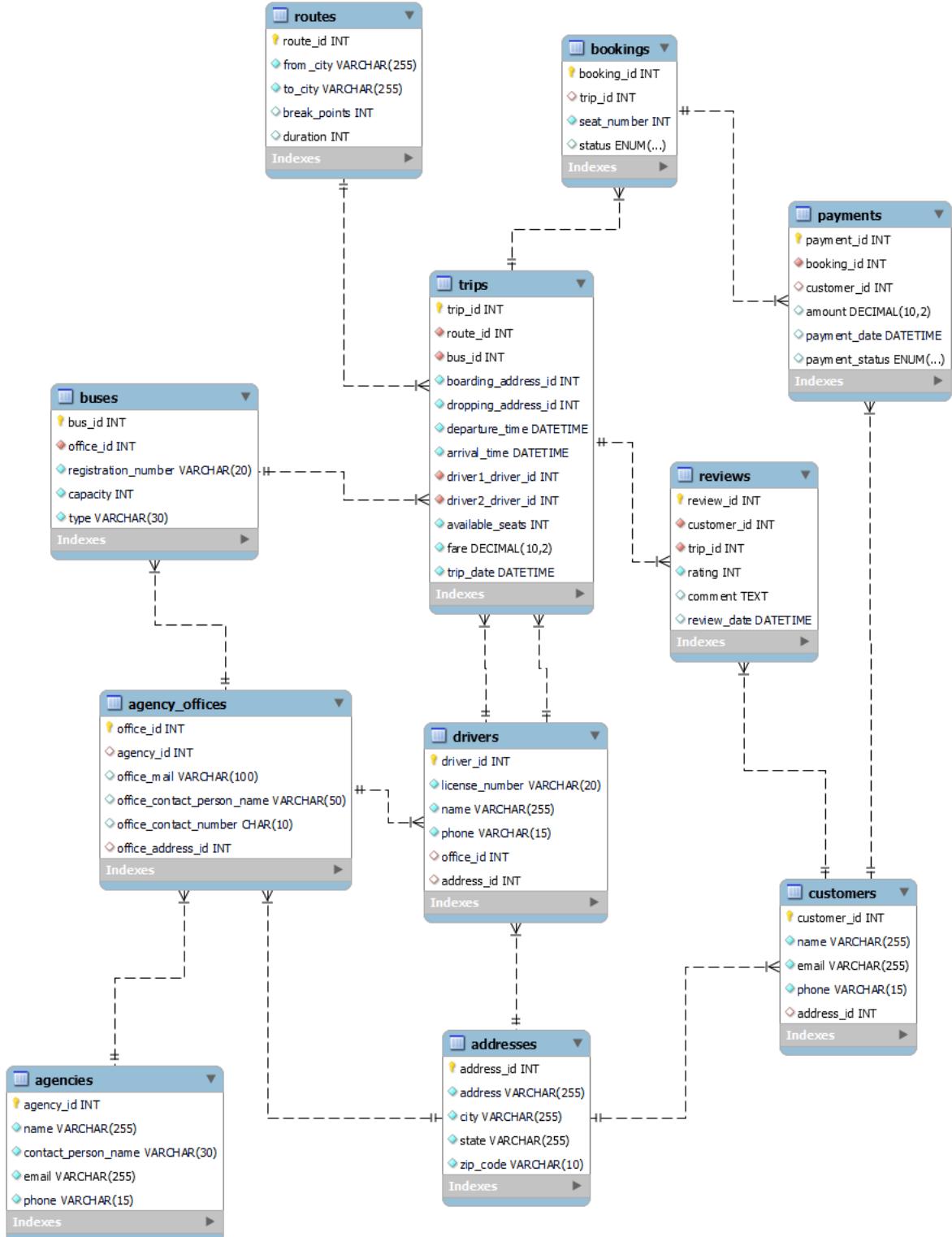
Design guidelines:

- 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
<u>/api/films/post</u> —	POST	<u>Add new Film object in DB</u>	String: Record Created Successfully
<u>/api/films/title/{title}</u> —	GET	<u>Search Films by Title</u>	Collection of Films
<u>/api/films/year/{year}</u> —	GET	<u>Search Films by Release Year</u>	Collection of Films
<u>/api/films/duration/gt/{rd}</u> —	GET	<u>Search Films where Rental Duration is greater than {rd}</u>	Collection of Films
<u>/api/films/rate/gt/{rate}</u> —	GET	<u>Search Films where Rental Rate is greater than {rate}</u>	Collection of Films
<u>/api/films/length/gt/{length}</u> —	GET	<u>Search Films where Length is greater than {length}</u>	Collection of Films

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

↳ Path URL	HTTP Methods	Description	✓ Success Response
/api/films/update/title/{id}	PUT	<u>Update Title of a Film</u>	Film
–			
/api/films/update/releaseyear/{id}	PUT	<u>Update Release Year of a Film</u>	Film
–			
/api/films/update/rentaldurtrion/{id}	PUT	<u>Update Rental Duration of a Film</u>	Film
–			
/api/films/update/rentalrate/{id}	PUT	<u>Update Rental Rate of a Film</u>	Film
–			
/api/films/update/rating/{id}	PUT	<u>Update Rating of a Film</u>	Film
–			
/api/films/update/language/{id}	PUT	<u>Update Language of a Film</u>	Film
–			
/api/films/update/category/{id}	PUT	<u>Update Category of a Film</u>	Film
–			
/api/actors/post	POST	<u>Add new Actor object in DB</u>	String: Record Created Successfully
–			
/api/actors/last name/{ln}	GET	<u>Search Actors by Last Name</u>	Collection of Actors
–			
/api/actors/first name/{fn}	GET	<u>Search Actors by First Name</u>	Collection of Actors
–			
/api/actors/update/lastname/{id}	PUT	<u>Update Last Name of an Actor</u>	Actor
–			
/api/actors/update/firstname/{i	PUT	<u>Update First Name of an Actor</u>	Actor
–			

↳ Path URL	HTTP Methods	Description	✓ Success Response
–	–	–	–
<u>/api/actors/{id}/films</u>	GET	<u>Search Films of an Actor by Actor Id</u>	Collection of Films
–	–	–	–
<u>/api/actors/{id}/film</u>	PUT	<u>Assign Film to an Actor</u>	Collection of Films
–	–	–	–
<u>/api/actors/top10enbyfilmcount</u>	GET	<u>Find top 10 Actors by Film Count</u>	Id & Name of Actors with Number of Films
–	–	–	–
<u>/api/customers/post</u>	POST	<u>Add new Customer Object in DB</u>	String: Record Created Successfully
–	–	–	–
<u>/api/customers/lastname/{ln}</u>	GET	<u>Search Customers by last name</u>	Collection of Customers
–	–	–	–
<u>/api/customers/firstname/{fn}</u>	GET	<u>Search Customers by first name</u>	Collection of Customers
<u>/api/customers/email/{email}</u>	GET	<u>Search Customers by email</u>	Customer
–	–	–	–
<u>/api/customers/{id}/{addressId}</u>	PUT	<u>Assign Address to a Customer</u>	Customer & Address details
–	–	–	–
<u>/api/customers/city/{city}</u>	GET	<u>Search Customers by City</u>	Collection of Customers with Address
–	–	–	–
<u>/api/customers/country/{country}</u>	GET	<u>Search Customers by Country</u>	Collection of Customers with

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

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

↳ Path URL	HTTP Methods	Description	✓ Success Response
<u>/api/store/{storeId}/address/{addressId}</u>	PUT	<u>Assign Address to a Store</u>	Store & Address details
<u>/api/store/country/{country}</u>	GET	<u>Search Store by Country</u>	Collection of Store with Address
<u>/api/store/city/{city}</u>	GET	<u>Search Store by City</u>	Collection of Store with Address
<u>/api/store/phone/{phone}</u>	GET	<u>Search Store by phone number</u>	Store with Address
<u>/api/store/update/{storeId}/{phone}</u>	PUT	<u>Update phone number of a Store</u>	Store
<u>/api/store/{storeId}/manager/{manager_staff_id}</u>	PUT	<u>Assign manager to a Store</u>	Store & Address details
<u>/api/store/staff/{storeId}</u>	GET	<u>Display all Staff of a Store</u>	Collection of Staff
<u>/api/store/customer/{storeId}</u>	GET	<u>Display all Customers of a Store</u>	Collection of Customers
<u>/api/store/manager/{storeId}</u>	GET	<u>Display Manager Details of a Store</u>	Staff

↳ Path URL	HTTP Methods	Description	✓ Success Response
<u>/api/store/managers</u> —	GET	<u>Display Manager details (first name, last name, email, phone) and Store details (Address, City, Phone) of all stores</u>	Custom Object
<u>/api/inventory/add</u> —	POST	<u>Add one more Film to a Store</u>	String: Record Created Successfully
<u>/api/inventory/films</u> —	GET	<u>Display inventory(count) of all Films in all Stores</u>	Collection of Film title, no. of copies
<u>/api/inventory/store/{id}</u> —	GET	<u>Display inventory of all Films by a Store</u>	Collection of Film title, no. of copies
<u>/api/inventory/film/{id}</u> —	GET	<u>Display inventory(count) of a Film in all Stores</u>	Collection of Store address, no. of copies
<u>/api/inventory/film/{id}/store/{id}</u> —	GET	<u>Display inventory(count) of a Film in a Store</u>	Store Address, no. of copies
<u>/api/rental/add</u> —	POST	<u>Rent a Film</u>	String: Record Created Successfully
<u>/api/rental/customer/{id}</u> —	GET	<u>Display all Films rented to a customer</u>	Collection of Films
<u>/api/rental/top10films</u> —	GET	<u>Display top 10 most rented Films</u>	Collection of Films

↗ Path URL	HTTP Methods	Description	✓ Success Response
/api/rental/toptenfilms/store/{id}	GET	<u>Display top 10 most rented Films of a Store</u>	Collection of Films
- /api/rental/due/store/{id}	GET	<u>Display list of Customers who have not yet returned the Film</u>	Collection of Customers
/api/rental/update/returndate/{id}	POST	<u>Update return date</u>	Rental
/api/payment/add	PUT	<u>Make one Payment entry</u>	String: Record Created Successfully
/api/payment/revenue/datewise	GET	<u>Calculate Cumulative revenue of all Stores</u>	Collection of payment_date, amount
/api/payment/revenue/datewise/store/{id}	GET	<u>Calculate Cumulative revenue of a Store</u>	Collection of payment_date, amount
/api/payment/revenue/filmwise/	GET	<u>Calculate Cumulative revenue of all Films (across all stores)</u>	Collection of Film title, amount
/api/payment/revenue/film/{id}	GET	<u>Calculate Cumulative revenue of a Film Store wise</u>	Collection of Store address, amount
/api/payment/revenue/films/store/{id}	GET	<u>Calculate Cumulative revenue of all Films by Store</u>	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
/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.

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

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

↳ Path URL	↗ HTTP Methods	⇒ Description	✓ Success Response
<u>/api/store/bus_type/{type}/trip_date/{trip_date}</u>	GET	<u>Search trips by bus type and trip_date</u>	Collection of Staff
<u>/api/reviews</u>	POST	<u>Add new review.</u>	String: Record Created Successfully
<u>/api/reviews</u>	GET	<u>Search all reviews.</u>	Collection of reviews.
<u>/api/reviews</u>	PUT	<u>Update review details.</u>	String: Record Updated Successfully.
<u>/api/reviews</u>	DELETE	<u>Remove review details.</u>	String: Record Deleted Successfully.
<u>/api/reviews/tripid/{trip_id}</u>	GET	<u>Search reviews by trip_id.</u>	Collection of reviews.
<u>/api/payment</u>	POST	<u>Add new Payment details.</u>	String: Record Created Successfully
<u>/api/payment</u>	GET	<u>Search all payment details.</u>	Collection of payments.
<u>/api/payment/{payment_id}</u>	GET	<u>Search payment details by payment_id.</u>	Payment Object.
<u>/api/payment/customer/{customer_id}</u>	GET	<u>Search payment details by customer_id.</u>	Collection of payments.
<u>/api/payment/agency/{agency_id}</u>	GET	<u>Search payment details by agency_id.</u>	Collection of payments.
<u>/api/payment/office/{office_id}</u>	GET	<u>Search payment details by agency_id.</u>	Collection of payments.

↳ Path URL	↗ HTTP Methods	⇒ Description	✓ Success Response
/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.
<u>Untitled</u>			
/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.
/api/reviews/customer/{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.
<u>Untitled</u>			
/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.

↳ Path URL	↗ HTTP Methods	➤ Description	✓ Success Response
<u>Untitled</u>			
<u>/api/agencies/offices</u>	GET	<u>Serach all offices.</u>	Collection of offices.
<u>/api/agencies/addoffice</u>	POST	<u>Add new branch office of an agency.</u>	String: Record Created Successfully.
<u>Untitled</u>			
<u>/api/routes</u>	PUT	<u>Update Route details.</u>	String: Record Updated Successfully
<u>/api/routes/{route_id}</u>	DELETE	<u>Remove Route.</u>	String: Record Deleted Successfully.