

Projekt Bazy Danych 2

Biuro turystyczne

Grzegorz Liana Krzysztof Usnarski

Technologie:

Backend: Spring Boot

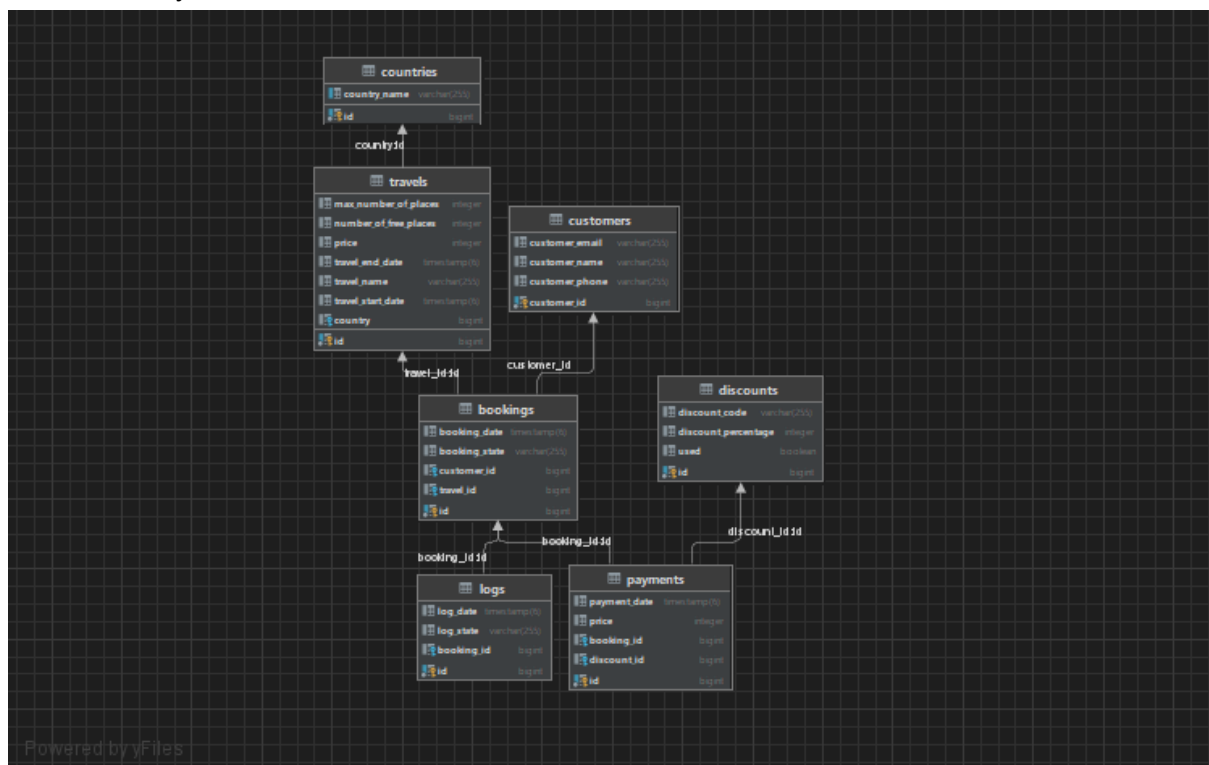
Baza danych: PostgreSQL

Biuro Turystyczne

Aplikacja biura turystycznego umożliwiająca użytkownikom rezerwacje miejsc na wycieczkach.

Dla użytkownika będzie dostępna również historia odbytych/kupionych wycieczek.

Schemat bazy:



Tabele

Tabela Countries - tabela słownikowa dla krajów.

Tabela Travels - tabela przechowuje dostępne wycieczki oraz aktualny stan wolnych miejsc dla danej wycieczki.

Tabela Customers - tabela przechowująca dane klientów.

Tabela Bookings - historia złożonych rezerwacji/zakupów wycieczek.

Tabela Logs - przechowuje historię zmian statusu rezerwacji.

Tabela Payments - przechowuje zrealizowane płatności klientów.

Tabela discounts - przechowuje kody rabatowe dla danych klientów. Kody rabatowe są w postaci jednorazowych zniżek do wykorzystania przy płatności.

Opis API

OpenAPI definition

Overview

Paths

GET /api/v1/travels/{id}

Get travel details by given ID

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

PUT /api/v1/travels/{id}

Update travel details with given id

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

DELETE /api/v1/travels/{id}

Delete travel with given id

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

PUT /api/v1/payments/customer/{id}/pay

Pay for bookings by customer id

Parameters

Type	Name	Description	Schema
query	id <i>required</i>		integer (int64)
query	paymentRequest <i>required</i>		PaymentRequest

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

GET /api/v1/customers/{id}

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

PUT /api/v1/customers/{id}

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

DELETE /api/v1/customers/{id}

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

GET /api/v1/countries/{id}

Get country details by ID

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

PUT /api/v1/countries/{id}

Update country with given id

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

DELETE /api/v1/countries/{id}

Delete country with given id

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

PUT /api/v1/booking/{id}/cancel

Updates state of Booking with given ID to CANCELLED

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

GET /api/v1/travels

Get all travels from database

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

POST /api/v1/travels

Create new travel with given details

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

POST /api/v1/discount

Creates new Discount

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

GET /api/v1/customers

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

POST /api/v1/customers

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

GET /api/v1/countries

Get all countries from database

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

POST /api/v1/countries

Create new country from given details

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

GET /api/v1/booking

Get all bookings from database

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

POST /api/v1/booking

Creates new Booking

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

GET /api/v1/payments/customer/{id}

Get customer payments by customer id

Parameters

Type	Name	Description	Schema
query	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

GET /api/v1/logs/travel/{id}

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

GET /api/v1/logs/customer/{id}

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

GET /api/v1/countries/search

Get all countries which contains query string

Parameters

Type	Name	Description	Schema
query	query <i>required</i>		string

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

GET /api/v1/booking/{id}

Get Booking by given ID

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

DELETE /api/v1/booking/{id}

Delete Booking by ID

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

GET /api/v1/booking/travel/{id}

Get all Booking assigned to Travel with given ID

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

GET /api/v1/booking/customer/{id}

Get all Booking assigned to Customer with given ID

Parameters

Type	Name	Description	Schema
path	id <i>required</i>		integer (int64)

Responses

Code	Description	Links
200	OK <i>Content</i> /	No Links

Components

Schemas

TravelRequest

Properties

Name	Description	Schema
travelName <i>required</i>		string
travelStartDate <i>optional</i>		string (date-time)
travelEndDate <i>optional</i>		string (date-time)

Name	Description	Schema
maxNumberOfPlaces <i>optional</i>		integer (int32)
price <i>optional</i>		integer (int32)
countryID <i>optional</i>		integer (int64)

TravelResponse

Properties

Name	Description	Schema
id <i>optional</i>		integer (int64)
travelName <i>optional</i>		string
travelStartDate <i>optional</i>		string (date-time)
travelEndDate <i>optional</i>		string (date-time)
maxNumberOfPlaces <i>optional</i>		integer (int32)
numberOfFreePlaces <i>optional</i>		integer (int32)
price <i>optional</i>		integer (int32)
countryID <i>optional</i>		integer (int64)

PaymentRequest

Properties

Name	Description	Schema
bookingIDs <i>optional</i>		< integer > array
discountCode <i>optional</i>		string

Booking

Properties

Name	Description	Schema
id <i>optional</i>		integer (int64)

Name	Description	Schema
customer <i>optional</i>		Customer
travel <i>optional</i>		Travel
bookingDate <i>optional</i>		string (date-time)
bookingState <i>optional</i>		enum (PAID,CANCELLED,NEW)

Country

Properties

Name	Description	Schema
id <i>optional</i>		integer (int64)
countryName <i>optional</i>		string
travels <i>optional</i>		< Travel > array

Customer

Properties

Name	Description	Schema
customer_id <i>optional</i>		integer (int64)
customerName <i>optional</i>		string
customerEmail <i>optional</i>		string
customerPhone <i>optional</i>		string
bookings <i>optional</i>		< Booking > array
payments <i>optional</i>		< Payment > array

Payment

Properties

Name	Description	Schema
id <i>optional</i>		integer (int64)
customer <i>optional</i>		Customer
travel <i>optional</i>		Travel
paymentDate <i>optional</i>		string (date-time)
price <i>optional</i>		integer (int32)

Travel

Properties

Name	Description	Schema
id <i>optional</i>		integer (int64)
travelName <i>optional</i>		string
travelStartDate <i>optional</i>		string (date-time)
travelEndDate <i>optional</i>		string (date-time)
maxNumberOfPlaces <i>optional</i>		integer (int32)
numberOfFreePlaces <i>optional</i>		integer (int32)
price <i>optional</i>		integer (int32)
country <i>optional</i>		Country
bookings <i>optional</i>		< Booking > array
payments <i>optional</i>		< Payment > array

CustomerRequest

Properties

Name	Description	Schema
customerName <i>required</i>		string
customerEmail <i>required</i>		string

Name	Description	Schema
customerPhone <i>required</i>		string

CustomerResponse

Properties

Name	Description	Schema
id <i>optional</i>		integer (int64)
customerName <i>optional</i>		string
customerEmail <i>optional</i>		string
customerPhone <i>optional</i>		string

CountryRequest

Properties

Name	Description	Schema
countryName <i>required</i>		string

CountryResponse

Properties

Name	Description	Schema
id <i>optional</i>		integer (int64)
countryName <i>optional</i>		string

DiscountRequest

Properties

Name	Description	Schema
discountCode <i>optional</i>		string
discountPercentage <i>optional</i>		integer (int32)

Discount

Properties

Name	Description	Schema
id <i>optional</i>		integer (int64)
discountCode <i>optional</i>		string
discountPercentage <i>optional</i>		integer (int32)
used <i>optional</i>		boolean

CreateBookingRequest

Properties

Name	Description	Schema
travelID <i>optional</i>		integer (int64)
customerID <i>optional</i>		integer (int64)

PaymentResponse

Properties

Name	Description	Schema
id <i>optional</i>		integer (int64)
paymentDate <i>optional</i>		string (date-time)
price <i>optional</i>		integer (int32)
travelName <i>optional</i>		string

BookingLog

Properties

Name	Description	Schema
id <i>optional</i>		integer (int64)
booking <i>optional</i>		Booking
bookingDate <i>optional</i>		string (date-time)

Name	Description	Schema
bookingState <i>optional</i>		enum (PAID,CANCEL LED,NEW)