API Documentation

Concert Ticketing Endpoints

(There were issues trying to set up Swagger, so I've opted to type the documentation out here instead due to time constraints.)

Concert Endpoints

api/concert/{concertId} (GET)

concertId (Integer): the ID of the concert to get.

Returns status **NOT_FOUND** if the concert does not exist, **OK** if it exists.

api/concert/{concertId} (DELETE)

concertId (Integer): the ID of the concert to delete.

Returns Response saying which concert ID has been deleted.

api/concert/ (GET)

Returns a List of all concerts in the database.

api/concert/upcoming (GET)

Returns a List of all concerts in the database that begin after the current time.

api/concert/update (POST)

Accepts a Request Body with the following attributes:

Name	Type	Example	Description
concert_id	Integer	100	ID of the concert
concert_name	String	A's World Tour	Name of the concert
concert_description	String	A's First Concert In..	Description of the concert
concert_time	String	2026-01-08T09:05:00	Format in YYYY-MM-DDTHH:MM:SS Time the concert begins

Returns status **NOT_FOUND** if the concert does not exist, **OK** if it exists.

api/concert/create (POST)

Accepts a Request Body with the following attributes:

Name	Type	Example	Description
concert_name	String	A’s World Tour	Name of the concert
concert_description	String	A’s First Concert In..	Description of the concert
concert_time	String	2026-01-08T09:05:00	Format in YYYY-MM-DDTHH:MM:SS Time the concert begins

Returns status **BAD_REQUEST** if the input is not in the right format, **CREATED** if the concert has been successfully created.

BookingTimeSlot Endpoints

api/timeslot/{timeSlotId} (GET)

timeSlotId (Integer): the ID of the time slot to get.

Returns status **NOT_FOUND** if the time slot does not exist, **OK** if it exists.

api/timeslot/{timeSlotId} (DELETE)

timeSlotId (Integer): the ID of the time slot to delete.

Returns Response saying which time slot ID has been deleted.

api/timeslot/ (GET)

Returns a List of all the time slots in the database.

api/timeslot/concert/{concertId} (GET)

concertId (Integer): the ID of the concert which the time slots searched are for.

Returns a List of time slots for a given concert ID.

api/timeslot/upcoming (GET)

Returns a List of time slots which begin in the future.

api/timeslot/available (GET)

Returns a List of time slots which are currently available (beings before the current time and ends after the current time) and is not already full.

api/timeslot/update (POST)

Accepts a Request Body with the following attributes:

Name	Type	Example	Description
time_slot_id	Integer	101	ID of the time slot
concert_id	Integer	102	ID of the concert the time slot is for
open_time	String	2025-01-08T09:05:00	Format in YYYY-MM-DDTHH:MM:SS Time the time slot opens for booking
close_time	String	2025-01-13T09:05:00	Format in YYYY-MM-DDTHH:MM:SS Time the time slot closes and doesn’t accept more bookings
max_booking_count	Integer	1000	The maximum amount of bookings that can be made in a given time slot

Returns status **NOT_FOUND** if the time slot does not exist, **OK** if it exists.

api/timeslot/create (POST)

Accepts a Request Body with the following attributes:

Name	Type	Example	Description
concert_id	Integer	102	ID of the concert the time

			slot is for
open_time	String	2025-01-08T09:05:00	Format in YYYY-MM-DDTHH:MM:SS Time the time slot opens for booking
close_time	String	2025-01-13T09:05:00	Format in YYYY-MM-DDTHH:MM:SS Time the time slot closes and doesn’t accept more bookings
max_booking_count	Integer	1000	The maximum amount of bookings that can be made in a given time slot

Returns status **BAD_REQUEST** if the input is not in the right format, **CREATED** if the time slot has been successfully created.

Booking Endpoints

api/booking/{bookingId} (GET)

bookingId (Integer): the ID of the booking to get.

Returns status **NOT_FOUND** if the booking does not exist, **OK** if it exists.

api/booking/{bookingId} (DELETE)

bookingId (Integer): the ID of the booking to delete.

Returns Response saying which booking ID has been deleted.

api/booking (GET)

Returns a List of all the bookings in the database.

api/booking/timeslot/{timeSlotId} (GET)

timeSlotId (Integer): the ID of the time slot for which the bookings being searched are made for.

Returns a List of the bookings made for a given time slot.

api/booking/update (POST)

Accepts a Request Body with the following attributes:

Name	Type	Example	Description
booking_id	Integer	103	ID of the booking
time_slot_id	String	101	ID of the time slot the booking is for
booking_time	String	2026-01-08T09:05:00	Format in YYYY-MM-DDTHH:MM:SS The time the booking was made
user_id	Integer	2121	ID for the user that made the booking

Returns status **NOT_FOUND** if the booking does not exist, **OK** if it exists.

api/booking/create (POST)

Accepts a Request Body with the following attributes:

Name	Type	Example	Description
time_slot_id	String	101	ID of the time slot the booking is for
user_id	Integer	2121	ID for the user that made the booking

Returns status **BAD_REQUEST** if the input is not in the right format or the booking cannot be made, **CREATED** if the booking has been successfully created.

api/booking/create_bulk (POST)

Accepts a Request Body with the following attributes:

Name	Type	Example	Description
time_slot_id	String	101	ID of the time slot the booking is for
user_id	Integer	2121	ID for the user that made the booking
ticket_count	Integer	8	The amount of bookings to me made

Returns status **BAD_REQUEST** if the input is not in the right format or the booking cannot be made, **CREATED** if the bookings have been successfully created.