

Swagger Hotel Booking - OpenAPI 3.0

1.0.11

OAS 3.0

[Todo](#)

[Terms of service](#)

[Contact the developer](#)

[Apache 2.0](#)

[Find out more about Swagger](#)

Servers

<https://petstore3.swagger.io/api/v3>

booking Access to hotel bookings

PUT /**booking** Update an existing booking

POST /**booking** Add a new booking to the store

GET /**booking/{bookingId}** Find booking by ID

room Operations about user

GET /**room** Get all rooms

GET /**room/{roomId}** Find room by ID

Schemas

booking >

room >

ApiResponse >

<https://editor.swagger.io/>

openapi: 3.0.3

info:

title: Swagger Hotel Booking - OpenAPI 3.0

description: |-

 Todo

termsOfService: <http://swagger.io/terms/>

contact:

 email: apiteam@swagger.io

license:

 name: Apache 2.0

 url: <http://www.apache.org/licenses/LICENSE-2.0.html>

version: 1.0.11

externalDocs:

 description: Find out more about Swagger

 url: <http://swagger.io>

servers:

 - url: <https://petstore3.swagger.io/api/v3>

tags:

 - name: booking

 description: Access to hotel bookings

 - name: room

 description: Operations about user

paths:

 /booking:

 put:

 tags:

 - booking

 summary: Update an existing booking

 description: Update an existing booking by Id

 operationId: updateBooking

 requestBody:

 description: Update an existent booking in the store

 content:

 application/json:

 schema:

\$ref: '#/components/schemas/booking'

application/xml:

schema:

\$ref: '#/components/schemas/booking'

application/x-www-form-urlencoded:

schema:

\$ref: '#/components/schemas/booking'

required: true

responses:

'200':

description: Successful operation

content:

application/json:

schema:

\$ref: '#/components/schemas/booking'

application/xml:

schema:

\$ref: '#/components/schemas/booking'

'400':

description: Invalid ID supplied

'404':

description: Booking not found

'405':

description: Validation exception

post:

tags:

- booking

summary: Add a new booking to the store

description: Add a new booking to the store

operationId: addBooking

requestBody:

description: Create a new booking in the store

content:

application/json:

schema:

\$ref: '#/components/schemas/booking'

application/xml:

schema:

\$ref: '#/components/schemas/booking'

application/x-www-form-urlencoded:

schema:

\$ref: '#/components/schemas/booking'

required: true

responses:

'200':

description: Successful operation

content:

application/json:

schema:

\$ref: '#/components/schemas/booking'

application/xml:

schema:

\$ref: '#/components/schemas/booking'

'405':

description: Invalid input

/booking/{bookingId}:

get:

tags:

- booking

summary: Find booking by ID

description: Returns a single booking

operationId: getBookingById

parameters:

- name: bookingId

in: path

description: ID of booking to return

required: true

schema:

type: integer

format: int64

responses:

'200':

description: successful operation

content:

application/json:

schema:

\$ref: '#/components/schemas/booking'

application/xml:

schema:

\$ref: '#/components/schemas/booking'

'400':

description: Invalid ID supplied

'404':

description: room not found

/room:

get:

tags:

- room

summary: Get all rooms

description: get all rooms

operationId: getAllRooms

responses:

'200':

description: successful operation

content:

application/json:

schema:

type: array

items:

\$ref: '#/components/schemas/room'

application/xml:

schema:

type: array

items:

\$ref: '#/components/schemas/room'

'400':

description: Invalid status value

/room/{roomId}:

get:

tags:

- room

summary: Find room by ID

description: Returns a single room

operationId: getRoomById

parameters:

- name: roomId

in: path

description: ID of room to return

required: true

schema:

type: integer

format: int64

responses:

'200':

description: successful operation

content:

application/json:

schema:

\$ref: '#/components/schemas/room'

application/xml:

schema:

\$ref: '#/components/schemas/room'

'400':

description: Invalid ID supplied

'404':

description: room not found

components:

schemas:

booking:

type: object

properties:

id:

type: integer

format: int64

example: 10

title:

type: string

example: title

additional_notes:

type: string

example: notes

name:

type: string

example: Muster

surname:

type: string

example: Max

email:

type: string

format: email

example: max.muster@gmail.com

room_id:

type: integer

format: int64

example: 10

created_at:

type: string

format: date-time

example: 2024-01-01T00:00:00.000Z

booking_start:

type: string

format: date-time

example: 2024-01-01T00:00:00.000Z

booking_end:

type: string

format: date-time

example: 2024-01-01T00:00:00.000Z

breakfast:

type: boolean

total_cost:

type: integer

format: int64

example: 10

cancelled:

type: boolean

xml:

name: booking

room:

type: object

properties:

id:

type: integer

format: int64

example: 10

title:

type: string

example: title

dedscription:

type: string

example: details

price:

type: integer

format: int64

example: 10

optimal_for:

type: integer

format: int64

example: 2

image:

type: string

example: "https://example.com/images/room.jpg"

xml:

name: room

ApiResponse:

type: object

properties:

code:

type: integer

format: int32

type:

type: string

message:

type: string

xml:

name: '##default'

requestBodies:

booking:

description: booking object that needs to be added to the store

content:

application/json:

schema:

\$ref: '#/components/schemas/booking'

application/xml:

schema:

\$ref: '#/components/schemas/booking'