

mayHotel API

User

POST /user - user register

Parameters: firstname: String, lastname: String, user_id: String, phone: String, address: String

response: UserSchema

GET /user/{user_id} - get user by id

Parameters: user_id: String

response: UserSchema

PUT /user - user register

Parameters: user_id: String, phone: String, address: String

response: UserSchema

Hotel

POST /hotel - creates new hotel

Parameters: name: String, logo: String

response: hotel schema

Room

POST /hotel/rooms - add rooms to hotel

Parameters: hotel_id: String, min: Number, max: Number, exc: String

response: Array[roomSchema]

GET /hotel/rooms/{room_id} - get room by id

Parameters: room_id: String

response: roomSchema

GET /hotel/rooms/all/{hotel_id} - get all rooms by hotel

Parameters: hotel_id: String

response: Array[roomSchema]

GET /hotel/rooms/available/{hotel_id} - get all available rooms by hotel

Parameters: hotel_id: String

response: Array[roomSchema]

POST /hotel/rooms/checkin - add guest to room

Parameters: room_id: String, user_id: String

response: roomSchema

POST /hotel/rooms/checkout - remove guest from room

Parameters: room_id: String, user_id: String

response: roomSchema

Service

GET /hotel/rooms/services/{hotel_id} - get list of all hotel calls

Parameters: hotel_id: String

```
response: [ {
  "id": "5cb73801fa928b47804aecbe",
  "room_num": 1,
  "type": "missing_items",
  "date": "2019-04-25T12:05:33.610Z",
  "item": "towel",
  "quantity": 2
},
{
  "id": "5cc1a61c83f45018c41baa1a",
  "room_num": 2,
  "type": "maintenance",
  "date": "2019-04-25T12:07:47.166Z",
  "desc": "tv not working"
}]
```

PUT /hotel/rooms/services/missing - add missing

Parameters: room_id: String, item: String, quantity: Number

response: roomSchema

DELETE /hotel/rooms/services/missing - remove missing

Parameters: call_id: String

response: roomSchema

PUT /hotel/rooms/services/maintenance

Parameters: room_id: String , desc: String

response: roomSchema

DELETE /hotel/rooms/services/maintenance

Parameters: call_id: String

response: roomSchema

PUT /hotel/rooms/services/alarm

Parameters: room_id: String, date, time

response: roomSchema

DELETE /hotel/rooms/services/alarm

Parameters: room_id: String

response: roomSchema

PUT /hotel/rooms/services/clean

Parameters: room_id: String, value: bool

response: roomSchema

Meal

POST /hotel/meals

Parameters: hotel: String, name: String, startTime: String, endTime: String
response: mealSchema

DELETE /hotel/meals

Parameters: meal_id: String
response:

DELETE /hotel/meals/all

Parameters: hotel_id: String
response:

POST /hotel/meals/enter

Parameters: hotel_id: String, user_id: String
response: free_table: Number , status: String

POST /hotel/meals/exit

Parameters: hotel_id: String, user_id: String
response: free_table: Number

Schedule

POST /hotel/schedule/ - add schedule to hotel

Parameters: hotel_id: String, date: Date
response: Array[scheduleSchema]

GET /hotel/schedule/{hotel} - get schedule by hotel

Parameters: hotel: String
response: Array[scheduleSchema]

DELETE /hotel/schedule/ - delete all past schedules

Parameters:
response: Array[scheduleSchema]

SPA

POST /hotel/schedule/spa - add spa appointments to spa in schedule

Parameters: hotel: String, date: Date, therapist: String
response: Array[SpaSchema]

GET /hotel/schedule/spa/ {hotel} - get all available appointments by hotel

Parameters: hotel_id: String
response: Array[SpaSchema]

GET /hotel/schedule/spa/by_date/ {hotel}/{date} - get all available appointments by hotel
Parameters: hotel: String, date/Date
response: Array[SpaSchema]

PUT /hotel/schedule/spa/add - add appointment to guest
Parameters: appointment_id: String, user_id: String, treatment: String
response: SpaSchema

PUT /hotel/schedule/spa/cancel - cancel guest appointment
Parameters: appointment_id: String
response: appointment id

Table

POST /hotel/tables/ add tables
Parameters: hotel: String, number: Number, sits: Number
response: TableSchema

GET /hotel/tables/{table_id} - get table by id
Parameters: table_id: String
response: TableSchema

GET /hotel/tables/all/{hotel_id} - get all tables by hotel
Parameters: hotel_id: String
response: TableSchema

DELETE /hotel/tables/ - delete table by id
Parameters: table_id: String
response: TableSchema

PUT /hotel/tables/ - edit table
Parameters: table_id: String, number: Number, sits: Number
response: TableSchema

DELETE /hotel/tables/clearOrders - delete all orders from table
Parameters: table_id: String
response: TableSchema

Order

PUT /hotel/tables/orders - add guest order
Parameters: meal:String, user:String, date:Date, sits: Number
response: TableSchema

DELETE /hotel/tables/orders - cancel guest order
Parameters: order_id: String
response: TableSchema

Schemas

UserSchema = {

```
_id:    {type:String, required: true},
firstname: {type:String, required: true},
lastname: {type:String, required: true},
phone:    {type:String, required: true},
address:  {type:String, required: true},
room:     {type: objectID , ref : 'Room', default: null},
QRcode:   {type:String}
}
```

HotelSchema = {

```
name: {type:String, required: true},
logo: String,
}
```

RoomSchema = {

```
hotel: {type: objectID , ref : 'Hotel', required: true},
number: {type:Number, required: true},
user:   {type: String , ref : 'User', default: null},
room_service: {
  missing_items: [{item: String, quantity: Number, date: {type: Date, default: new Date()}}],
  maintenance: [{desc: String, date: {type: Date, default: new Date()}}],
  isCleanable: {type:Boolean, default: true},
  alarmClock: {type:Date, default: null}
}
}
```

ScheduleSchema = {

```
hotel: {type: objectID , ref : 'Hotel', required: true},
date: {
  type: Date,
  required: [true, 'date param missing'],
  get: v => `${v.getDate()}/${v.getMonth()}/${v.getFullYear()}`
},
day: {
  type:Number,
  min: [1, 'day is illegal. min is 1'],
  max: [7, 'day is illegal. max is 7']
},
spa: [SpaSchema],
events: [EventSchema]
}
```

```

SpaSchema = {
  therepist: {type: String , required: true},
  user:      {type: String , ref : 'User', default: null},
  treatment: {type: String, default: null},
  status:    {type: Boolean, default: false},
  time: {type: String, required: true},
  int: {time: Number}
}

```

```

MealSchema = new Schema({
  hotel: {type: objectID , ref : 'Hotel', required: true},
  name:  {type: String, required: true},
  startTime: { type: String, required: [true, 'time param missing'] },
  endTime: {type: String, required: [true, 'time param missing']},
  int: {
    startTime: Number,
    endTime: Number
  }
}
)

```

```

OrderSchema = {
  meal: {type: objectID, ref : 'Meal', required: false},
  user: {type: String , ref : 'User', required: true},
  at:    {type: Number, required: true}, //INT -> YYYYMMDD
  status: {
    isArrived: {type:Boolean, default: false},
    isLeft:    {type:Boolean, default: false}
  }
}

```

```

TableSchema = {
  hotel: {type: objectID , ref : 'Hotel', required: true},
  number: {type:Number, required: true, min: [1, 'room number illegal']},
  sits:   {type: Number, required: true, min: [1, 'at least 1 sit']},
  curr_user: {type: String , ref : 'User', default: null},
  orders: [OrderSchema]
}

```