**API Documentation**

\*In every object deleted attribute is by default false.

**User**

Data transfer object

class UserDto{

username: string;

password: string;

email: string;

mobileNo: string;

addressLine1: string; *optional*

addressLine2: string; *optional*

type: string; *can be either 'admin', 'customer', 'merchant'*

image: string; *optional*

deleted: boolean;

}

**To create user:**

Request: post

Data Type: json

url: <http://localhost:3000/user>

**To login:**

Request: post

Data Type: json

url: <http://localhost:3000/auth/login>

Data example: {"email": "example@example.com",

"password": "abc123”,

“type”: “customer”}

\*login request returns a token. Use that token in **authorization** as **Bearer Token** type in every request you make.

**Merchant**

Data transfer object

class MerchantDto{

user:string;

name:string;

longitude:string;

latitude:string;

openTime:string;

closeTime:string;

}

**To create merchant:**

Request: post

Data Type: json

url: <http://localhost:3000/merchant>

**To update Merchant**

\*to update merchant, send json data with attributes of user if you need to update user attributes. The same applies for customer and admin.

Data transfer object

class MerchantDto{

user:string;

name:string;

longitude:string;

latitude:string;

openTime:string;

closeTime:string;

username: string;

password: string;

email: string;

mobileNo: string;

addressLine1: string;

addressLine2: string;

type: string; *can be either 'admin', 'customer', 'merchant'*

image: string;

deleted: boolean;

}

Request: post

Data Type: json

url: <http://localhost:3000/merchant>

**To find Merchant**

\*To find a merchant by id just add id.

Request: get

Data Type: json

url: http://localhost:3000/ merchant /001

\*To find all:

Request: get

Data Type: json

url: http://localhost:3000/ merchant /

\* To find by values, send a string parameter which could be convertible to json object. In that string include the attribute as key (ex: name) and value as object value.

**\*Same applies for admin, customer, user, service.**

Request: get

Data Type: json

url: http://localhost:3000/merchant?filter={"name":"john"}

**Customer**

Data transfer object

CustomerDto{

user:Types.ObjectId;

firstName:string;

lastName:string;

}

**Service**

Data transfer object

class ServiceDto {

merchant: Types.ObjectId

name: string

price: number

duration: number

type: string *can be either 'number', 'time'*

maxCount: number *default: 1*

deleted: boolean;

}

To create service:

Request: post

Data Type: json

url: <http://localhost:3000/service>

**Appointment**

Data transfer object

class AppointmentDto {

    customer: Types.ObjectId;

    service: Types.ObjectId;

    merchant: Types.ObjectId;

    date: Date;

    startTime: Date;

    endTime: Date; ***No need to send endTime api wiil generate.***

    subService: number;

    price: number;

    token: number;

    status: string; ***Status can be either*** 'pending', 'accepted', 'canceled', 'completed'

}

\***If** **the service is token based no need of time slot, if it is time based no need of token.**

To create appointment

Request: post

Data Type: json

url: [http://localhost:3000/appointment/](http://localhost:3000/appointment/me)

example data

{

    "customer": "6050598fad08410015beb5fa",

    "service": "603e2c3960160a408833b4f4",

    "merchant": "604cb5aed5b4b70015454d1f",

    "date": "Sat May 03 2021 05:30:00",

    "startTime": "Sat May 03 2021 16:00:00",

    "subService": "1",

    "price": "10000",

    "status": "pending"

}

To get appointments of logged in user:

Request: get

Data Type: json

url: <http://localhost:3000/appointment/me>