API Documentation:

User:

Represents a User entity for all user types.

User Attributes:

- username (String): unique identifier

- password (String): password for the user

- name (String): full name

- address (String): address of the user

- contact (String): contact information

- role (String): user type (i.e. WORKER, CUSTOMER, ADMIN)

- created_at (Date): Date the entity was created

- updated_at (Date): Date the entity was updated

Create User (POST):

- URL: /api/user

- Body:

{

   "username": "Username",

   "password": "Password",

   "name": "JohnDoe",

   "address": "SampleAddress",

   "contact": "SampleContact",

   "role": "CUSTOMER"

}

- Response:

     o 201:

§ Will return a body containing the user entity json

- o 400:

    § A non-blank field is left blank and the response body will say which part if missing.

- o 500:

    § An unexpected error occurred, for example, username is left blank.

Get Users based on Role (GET):

- URL: /api/user/role/{role}

- Body: Not needed

- Response:

    - o 200:

        § Will return a list of JSONs with user details of users with the specified role.

    - o 400:

        § If the role specified is an invalid role (i.e. not CUSTOMER, WORKER, ADMIN).

Delete User (DELETE):

- URL: /api/user/{username}

- Body: Not needed

- Response:

    - o 200:

        § Will return "User with username: '{username}' was deleted when the user is successfully deleted.

    - o 400:

        § If the username provided does not correspond with any of the current users.

Services:

Represents a Service entity.

User Attributes:

- id (long): Unique Identifier

- assigned_employee (User): A User, with the WORKER role, who is assigned to the service.

- created_at (Date): Date the entity was created

- updated_at (Date): Date the entity was updated

Create Service (POST):

- URL: /api/services

- Body:

```
{

    "service": "TEST_SERVICE",

    "assigned_employee":

    {

    "username": "work",

    "password": "pass",

    "name": "John",

    "address": "TEST",

    "contact": "TEST",

    "role": "WORKER"

    }

}
```

- Response:

- o 200:

  - § Return the full JSON of the service details of the newly created service.

- o 500

  - § An unexpected error occurred.

Get Service from Worker username (GET):

- - URL: /api/services/findService/{employeeUsername}

- - Body: Not needed

- - Response:

  - o 200:

    - § Returns a list of JSONs with the list of services, where the assigned employee has the username from the path variable {employeeUsername}.

  - o 500

    - § An unexpected error occurred.

Get Workers from Service name (GET):

- - URL: /api/services/findEmployees/{service}

- - Body: Not needed

- - Response:

  - o 200:

    - § Returns a list of JSONs with the list of employees (i.e. users with WORKER role), where the service has the name provided in the path variable {service}.

  - o 500

    - § An unexpected error occurred.

Booking:

Represents a Booking entity.

User Attributes:

- id (long): Unique Identifier

- booking_date (Date): Date which booking is being booked on

- booking_time (Date): Time which the booking is being booked for

- duration (Integer): How long the booking will go for

- assigned_employee (User): A User, with the WORKER role, who is assigned to the service.

- note (String): Any additional notes about the booking

- customer (User): A User, with the CUSTOMER role, who made the booking

- bookedService (Service): Service that has been booked

- created_at (Date): Date the entity was created

- updated_at (Date): Date the entity was updated

Create Booking (POST):

- URL: /api/booking/{workerName}/{servicename}

- Body: Although an assigned_worker and a bookedService can be specified in the body, if it is not and the worker name and service name is specified in path variable, then the API will find the corresponding worker and service.

```
{

    "booking_date": "2020-09-20",

    "booking_time": "23:30:00",

    "notes": "TEST",

    "customer" :

    {

    "username": "cust",

        "password": "pass",
```

```
        "name": "Bob",

        "address": "TEST",

        "contact": "TEST",

        "role": "CUSTOMER"

    },

    "duration": 1

}
```

- Response:

    o 201:

        § Will return a body containing the user entity JSON

    o 400:

        § A non-blank field is left blank and the response body will say
          which part if missing.

    o 500:

        § An unexpected error occurred.

Get Customer Booking from username (GET):

- URL: /api/services/findCustBooking/{username}

- Body: Not needed

- Response:

    o 200:

        § Returns a list of JSONs with the list of bookings, where the
          customer (i.e. User with CUSTOMER role) has the username
          provided in the path variable {username}.

    o 500

        § An unexpected error occurred.

Get Worker Booking from username (GET):

- URL: /api/services/findWorkerBooking/{username}

- Body: Not needed

- Response:

    o 200:

        § Returns a list of JSONs with the list of bookings, where the
          worker (i.e. User with WORKER role) has the username
          provided in the path variable {username}.

    o 500

        § An unexpected error occurred.