

Appointment Management API

Introduction

Our appointment management application was built to make it easier for clinic managers and their clients to manage their appointments.

In this API there are several basic abilities, like viewing appointments by dates, managing the clinic's users but the main feature of our API is to use free time blocks in a smart way and schedule an appointment right away, without waiting for the clinic's secretary to look for an available time for the next appointment.

It is also planned to build a mechanism that will auto-schedule appointments for new clients, and another feature that will show the clinic managers future appointments that can be schedule for sooner date, which will make the whole experience better for new customers. (will not be developed in this course because of time capacity issues).

Technical

The API is built with Java in use of Jersey framework for RESTful web services which saved a lot of time managing and organising the API calls.

MySQL is used for saving all of the clinic's data and Redis for caching the same data, for performance considerations. Redis is also used for saving the free time blocks for the next 2 weeks (for now), and is updated when appointments are created and updated (note: in this part I've only implemented change on creation of appointments, and on first fetch).

The free times are getting cleaned and refreshed every 12 hours (by demand, meaning only if some end user request it).

In This Document

This document summers the objects the API is working with for getting, creating and updating data in the server. Follow the API details in each section to test or use the API.

It is recommended to use a REST client like Postman or Paw for mac. (if you use Paw, there's a .paw file attached to this document which contains the whole API ready for testing).

Each API details contains the HTTP method (*GET POST PUT DELETE*), the URI, the required body, and the expected response.

About The Creator

Name: Ariel Pollack

ID: 301784898

Year: 3rd

Group: Maof

Objects

1. API Response:

```
{  
  "code": HTTP status code,  
  "data": JSON object or array,  
  "error": String or null,  
  "status": "success" or "error"  
}
```

2. User:

```
{  
  "email": String,  
  "first_name": String,  
  "id": String or integer,  
  "id_number": String,  
  "last_name": String,  
  "phone": String  
}
```

3. Appointment:

```
{  
  "date": Date in milliseconds,  
  "id": String or integer,  
  "therapist": User object,  
  "user": User object  
}
```

4. Free Time Block:

```
{  
  "date": Date in milliseconds,  
  "day": Integer (Day in week e.g. "1" is Sunday etc),  
  "duration": Integer (Time in minutes),  
  "hour": Integer (Hour in a 24h format),  
  "minute": Integer (0-59)  
}
```

User API

1. Login

```
# POST /APTManage/users/login
```

```
+ Request (application/json)
```

```
{
    "id_number": "1234567",
    "password": "test"
}
```

```
+ Response 200 (application/json)
```

```
{
  "code": 200,
  "data": {
    "appointments": [Appointment objects...],
    "user": User object
  },
  "error": null,
  "status": "success"
}
```

2. Update Profile

```
# PUT /APTManage/users/update
```

```
+ Request (application/json)
```

```
{
  "email": "ariel@zeek.me",
  "first_name": "Ariel",
  "id": 2,
  "id_number": "301784898",
  "last_name": "Pollack",
  "phone": null
}
```

```
+ Response 204 (application/json)
```

3. Get Appointments

```
# GET /APTManage/users/appointment?  
    start=milliseconds timestamp #optional  
    &end=milliseconds timestamp #optional
```

+ Response 200 (application/json)

```
{  
  "code": 200,  
  "data": [Appointment objects...],  
  "error": null,  
  "status": "success"  
}
```

4. Get Available Time Blocks

```
# GET /APTManage/users/appointment/available
```

+ Response 200 (application/json)

```
{  
  "code": 200,  
  "data": [Time block objects...],  
  "error": null,  
  "status": "success"  
}
```

5. Create Appointment

```
# POST /APTManage/users/appointment
```

+ Request (application/json)

```
{  
  "date": "1420205400000"  
}
```

+ Response 201 (application/json)

```
{  
  "code": 201,  
  "data": Appointment object,  
  "error": null,  
  "status": "success"  
}
```

6. Update Appointment

```
# PUT /APTManage/users/appointment
```

```
+ Request (application/json)
```

```
{  
  "date": 1420191000000, # new date  
  "id": 2  
}
```

```
+ Response 204 (application/json)
```

Admin API

1. Create User

POST /APTManage/admin/user

+ Request (application/json)

```
{
  "user": {
    "email": "bob@building.com",
    "first_name": "Bob",
    "id_number": "11122244",
    "last_name": "The Builder"
  },
  "password": "qwerty"
}
```

+ Response 201 (application/json)

```
{
  "status": "success",
  "data": User object,
  "code": 201
}
```

2. Get user by id

GET /APTManage/admin/user/{id}

+ Response 200 (application/json)

```
{
  "code": 200,
  "data": User object,
  "error": null,
  "status": "success"
}
```

3. Search user by id number or query

```
# GET /APTManage/admin/user?
                                query={query} or id_number={id number}

+ Response 200 (application/json)

{
  "code": 200,
  "data": [User objects...],
  "error": null,
  "status": "success"
}
```

4. Get appointments

```
# GET /APTManage/admin/appointment?
                                start={start} #required
                                &end={end} #required

+ Response 200 (application/json)

{
  "code": 200,
  "data": [Appointment objects...],
  "error": null,
  "status": "success"
}
```

5. Get appointments for user

```
# GET /APTManage/admin/appointment/user/{id}?
                                start={start} #required
                                &end={end} #required

+ Response 200 (application/json)

{
  "code": 200,
  "data": [Appointment objects...],
  "error": null,
  "status": "success"
}
```

6. Create appointment

```
# POST /APTManage/admin/appointment
```

```
+ Request (application/json)
```

```
{
  "date": "1420777200000",
  "therapist": {
    "id": "4"
  },
  "user": {
    "id": "3"
  }
}
```

```
+ Response 201 (application/json)
```

```
{
  "code": 201,
  "data": Appointment object,
  "error": null,
  "status": "success"
}
```

7. Delete appointment

```
# DELETE /APTManage/admin/appointment/{id}
```

```
+ Response 204 (application/json)
```

8. Update appointment

```
# PUT /APTManage/admin/appointment
```

```
+ Request (application/json)
```

```
{
  "date": "1420896600000",
  "id": "4",
  "therapist": {
    "id": "2"
  },
  "user": {
    "id": "3"
  }
}
```

```
+ Response 204 (application/json)
```