Frontend/Backend Endpoints

Overview

This page outlines the endpoints currently in use/in development/planned. The team should ensure all existing/removed endpoints are documented here for consistency between sub-teams

URLs

All testing for the project are done on the following ip address: mymedicalsecretary.uk.to with the 8080 port, which is a temporary hosting location just for our team for testing. The protocol used is HTTPS. **This testing location is still a publicly accessible IP address, so no secure/real data will be sent on this network**.

The API endpoints can all be found under the api section, followed by the table name, followed by the request type, followed by any request params required. For example, getting a patient with the id 1 can be done by using the URL: http://mymedicalsecretary.uk.to:8080/api/users/get_patient/1

All endpoints (except login) require authentication to access. This is done through JSON Web Tokens (JWTs), which authenticate and authorise a specific user. For any request asside from login/refresh, a valid JWT must be provided in the request header, and the service will throw 403 forbidden if the JWT is not valid. Tokens expire after 1 hour, but refresh tokens are available for a longer period of time, with the refresh token being valid for 1 day (this is somewhat flexible to the client, however).

Endpoints In Detail

Permissions

- ADMIN - Any admin can access
- PATIENT - Any patient can access
- SPECIFIC PATIENT - Patients with an ID matching either the specific ID or linked to the specific row in the DB
  - I.e. To get a doctor by ID, you must be a patient with appointments relating to that doctor
- ANYONE - Anyone with an internet connection can access
- TEST ONLY - Won't exist in the final product, can't be accessed by anyone

Table

Endpoint	Type	Request Params	Response	Status	Permission
Get Patient By ID	GET	Int user ID	User Object (Fields documented fully on postman)	LIVE	ADMIN SPECIFIC PA...
Get All Patients	GET		List of User Objects	LIVE	ADMIN
Get Admin by ID	GET	Int mms ID	User Object	LIVE	ADMIN
Create Patient	POST	User Object (Patient)	String message	TEST ONLY LIVE	ADMIN
Create Admin	POST	User Object (Admin)	String message	TEST ONLY LIVE	ADMIN

Get Doctor By ID	GET	Int doctor ID	Doctor Object	LIVE	ADMIN PATIENT
Get all doctors by Patient ID	GET	Int Patient ID	List of Doctor Objects	LIVE	ADMIN SPECIFIC PA...
Get All doctors	GET		List of Doctor Objects	LIVE	ADMIN
Create Doctor	POST	Doctor Object	String message	LIVE	ADMIN
Delete Doctor	POST	Doctor id	Success status	LIVE	ADMIN
Get Facility By ID	GET	Int FacilityID	Facility Object	LIVE	ADMIN PATIENT
Get Facilities by Type	GET	Type	List of Facility objects	LIVE	ADMIN PATIENT
Get All Facilities	GET		List of Facility objects	LIVE	ADMIN PATIENT
Create Facility	POST	FacilityObject	String message	LIVE	ADMIN
Delete Facility	POST	Int Facility id	Success status	LIVE	ADMIN
Get Appointment	GET	Appointment ID	Appointment Object	LIVE	ADMIN SPECIFIC PA...
Create Appointment	POST	Appointment Object	Success status	TEST ONLY LIVE	ADMIN
Get all appointments	GET	User ID	List of Appointment objects	LIVE	ADMIN SPECIFIC PA...
Update User note	POST	Note, appointment id	Success status	LIVE	SPECIFIC PA...
Delete Appointment	POST	Int Appoinement ID	Success status	LIVE TEST ONLY	ADMIN
Upload Patient File	POST	File Url	Success status	LIVE	ADMIN
Upload Appt File	POST	File Url	Success status	LIVE	ADMIN
Login	POST	Username & password	JWT, Refresh token	LIVE	ANYONE
Refresh	POST	Refresh token	JWT	LIVE	ANYONE
Update Password	PUT	Username, old & new passwords	Success status	LIVE	ADMIN PATIENT

Delete Resource	POST	Int resource ID	Success status	LIVE	ADMIN
Create Resource	POST	Resource Object	Success status	LIVE	ADMIN
Get Resource by ID	GET	Int resource ID	Resource Object	LIVE	ADMIN PATIENT
Get All Resources by patient id	GET	Int user ID	List of Resources	LIVE	ADMIN SPECIFIC PA...

*Note that all the ‘create’ endpoints are also update endpoints if object already exists

Example Requests

This section does not give a sample usage for every single request, just the most important cases to understand the general usage of the API. For full details of every request see Postman section

Users

Get Patient By ID (Genie ID)

URL: http://{{ip}}:{{port}}/api/users/get_patient/{id}

Get Admin By ID (Internal ID)

URL: http://{{ip}}:{{port}}/api/users/get_admin/{id}

Create Patient

URL: http://{{ip}}:{{port}}/api/users/create_patient

Body:

```
{
    "patientId": 19,
    "firstname": "john",
    "middleName": "Doe",
    "surname": "Smith",
    "dob": "1990-01-01T00:00:00Z",
    "email": "john2.doe@example.com",
    "street": "123 Main St",
    "suburb": "Cityville",
    "state": "California"
}
```

Create Admin

URL: http://{{ip}}:{{port}}/api/users/create_admin

Body:

```
{
    "email": "john.doe@example.com",
    "username": "admin2",
    "password": "password"
}
```

Doctors

Get Doctor By ID

URL: http://{{ip}}:{{port}}/api/doctors/get/{id}

Delete Doctor By ID

URL: http://{{ip}}:{{port}}/api/doctors/delete/{id}

Get Doctor By Patient ID

URL: http://{{ip}}:{{port}}/api/doctors/get/{id}

Create Doctor

URL: http://{{ip}}:{{port}}/api/doctors/create

Body:

```
{
    "name": "John",
    "address": "1 Street street",
    "contact": "2394823948",
    "email": "john.doe@example.com",
    "expertise": "Arms and legs",
    "website": "doctor.com"
}
```

Facilities

Get Facility By ID

URL: http://{{ip}}:{{port}}/api/facilities/get/{id}

Delete Facility By ID

URL: http://{{ip}}:{{port}}/api/facilities/delete/{id}

Create Facility

URL: http://{{ip}}:{{port}}/api/facilities/create

Body:

```
{
  "name": "Example Hospital",
  "contact": "123-456-7890",
  "address": "123 Example St, City, Country",
  "fax": null,
  "website": "http://examplehospital.com",
  "facilityType": "PATHOLOGY"
}
```

Appointments

Get Appointment By ID

URL: http://{{ip}}:{{port}}/api/appointments/get/{id}

Get All Appointments By User ID

URL: http://{{ip}}:{{port}}/api/appointments/get_all/{id}

Create Appointment

URL: http://{{ip}}:{{port}}/api/appointments/create

Body:

```
{
  "id": 3,
  "detail": "Routine checkup",
  "reason": "General health.",
  "note": "No specific notes",
  "dateCreate": "2022-04-08T10:30:00Z",
  "lastUpdated": "2022-04-08T12:45:00Z",
  "startTime": "10:30:00",
  "startDate": "2022-04-15",
  "duration": 60,
  "patientId": 3,
  "providerId": 3
}
```

Files

Upload User HTML From Genie

URL: http://{{ip}}:{{port}}/api/files/upload/patients

Body:

- Use ‘form-data’
- Create a key-value pair as follows:
  - Key: ‘file’
  - Value: Patient.html file
    - Example file format: Patient.html

Upload User HTML From Genie

URL: http://{{ip}}:{{port}}/api/files/upload/appointments

Body:

- Use ‘form-data’
- Create a key-value pair as follows:
  - Key: ‘file’
  - Value: Patient.html file
    - Example file format: Appointment.html

Postman

The API endpoints are most clearly documented in the Team Postman, where all request params/response bodies are clearly laid out with example requests. For access to this, please contact Jonathan Latti at jlatti@student.unimelb.edu.au