Application Notes

Contents

[Database setup 1](#_Toc87214429)

[Description: 1](#_Toc87214430)

[Attendance Table 1](#_Toc87214431)

[Classes Table 1](#_Toc87214432)

[Registration Table 1](#_Toc87214433)

[Students Table 1](#_Toc87214434)

[Teachers Table 1](#_Toc87214435)

[The Web client 2](#_Toc87214436)

[Description: 2](#_Toc87214437)

[The WebAPi 3](#_Toc87214438)

[Description: 3](#_Toc87214439)

# Database setup

## Description:

Essentially the database is an innoDB on MYSQL 5.7.30. No stored procedures the database has just 5 tables.

## Attendance Table

* Keeps attendance record for all classes attended when the register is marked.
* The user can specify any date in the past or future (not restriction)

## Classes Table

* Keeps a record of all the classes that can be taught by grade and which is the designate teacher of the class

## Registration Table

* Keeps a list of all the classes a student has registered into.
* The class registered implies the grade of the student as opposed to the student having a grade.

## Students Table

* This is just a list of students that can register for a class
* Students cannot register or log onto the system

## Teachers Table

* Keeps a list of all the teachers
* Teachers have password fields to grant system access
* There’s is a file named ***DatabseDump20211107.sq***l in the Database folder of the repository which can be used to recreate the database
* You would need to setup MYSQL server (or get one running off a docker image)
* Use MYSQL workbench to connect to the MYSQL instance and run the script ***DatabseDump20211107.sq***l to create the database

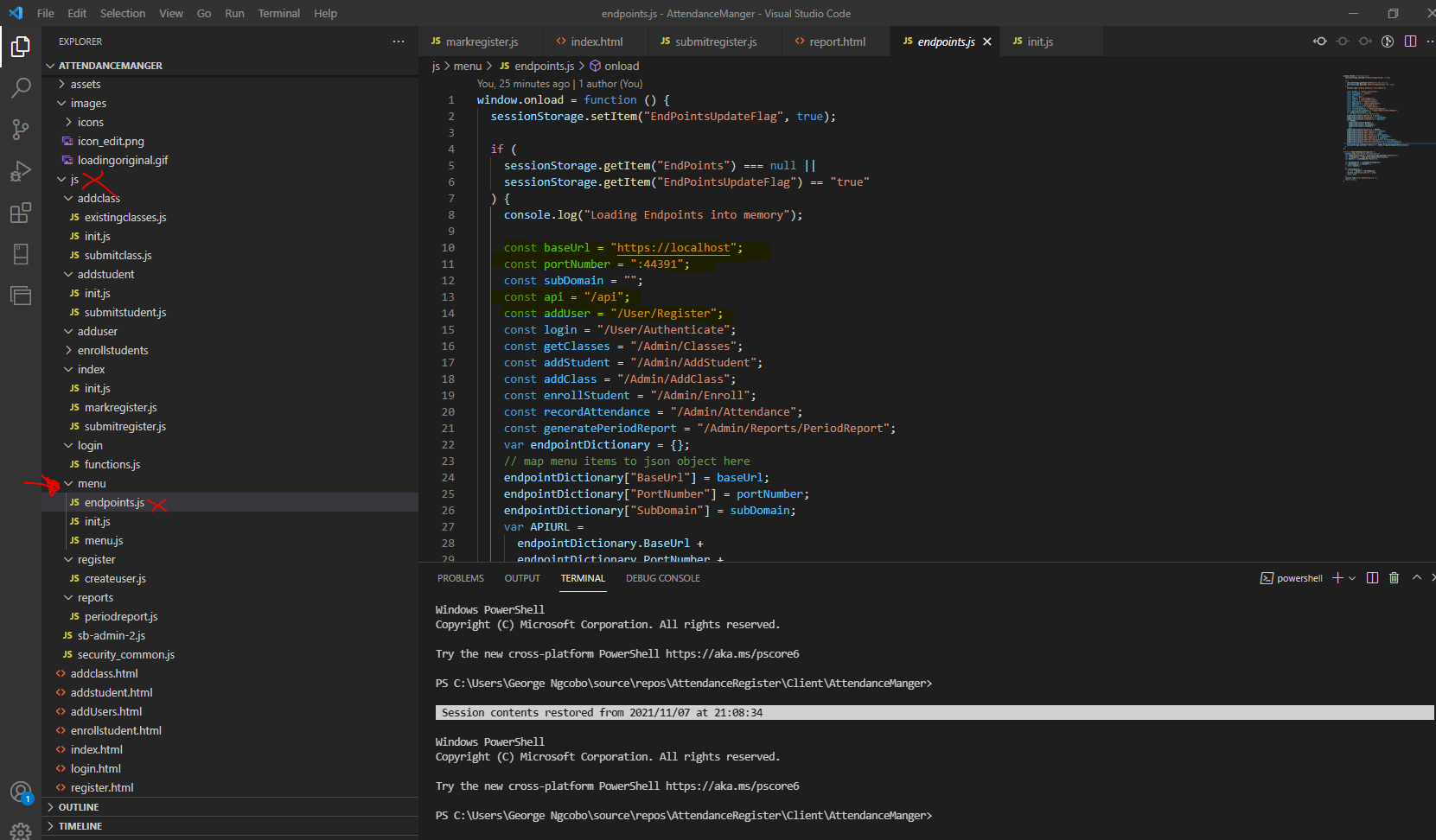
# The Web client

## Description:

The web client is just pure html and JavaScript using the jQuery library.

To configure the client to point to the web API the client uses a file called endpoints.js in the client repository under the “js” folder

Based on whether or not you decide to host the client in IIS or Apache or just run it directly from the folder (which I did) you would have to configure the client in this file to point to the API



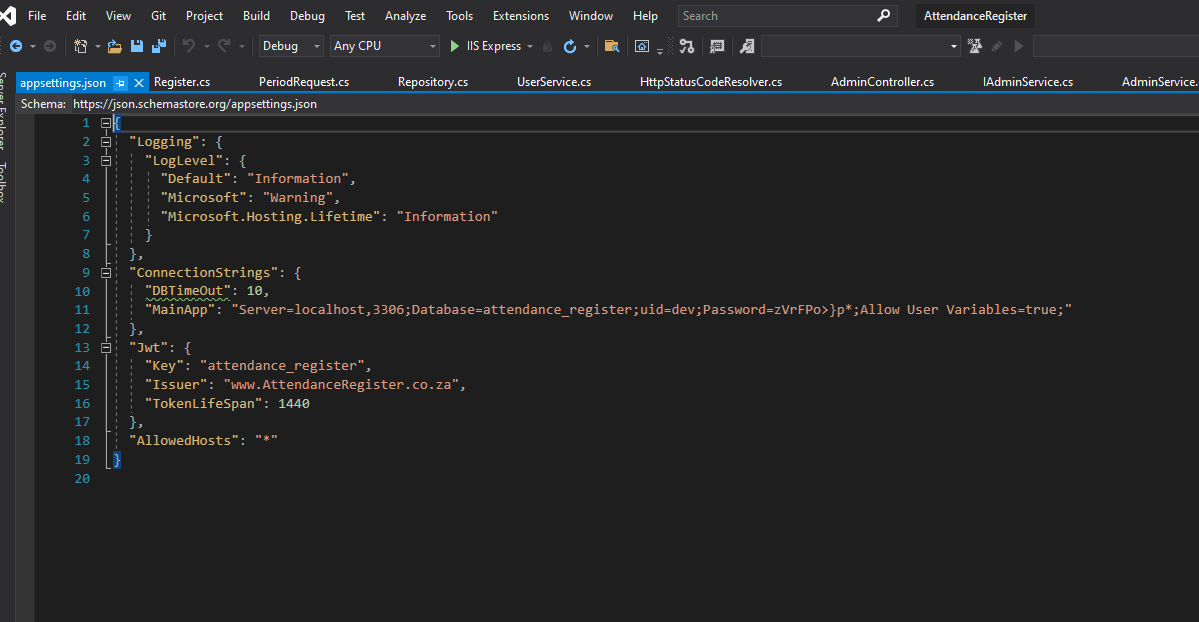
* Line 10 base URI points to the webserver the API will be hosted
* Line 11 allows you to specify the port number
* Line 12 need only be filled in if the API is hosted in some sub directory of your web server.
* Each page has a folder in the “JS” folder to cater for the java script code to manage the site.

# The WebAPi

## Description:

The Web API is essentially a .NET5 API The API has API docs that loads by default to show its methods and features [Swagger UI](https://localhost:44391/swagger/index.html)

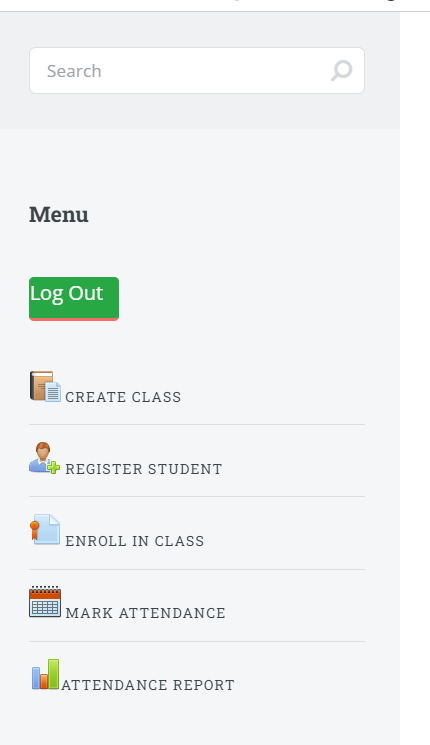
Use “appSettings,json “ to configure the API to point o the correct database as per line 11 on the screenshot



The API uses JWT to authenticate users.

# How to Use the application

## Firstly, the API must be running.

1. Login or register
   1. You just need an email address and a password
   2. You can launch any page , if you are not authenticated you will be redirected to login,html
2. The menu has the following Options:
   1. 
3. Create a Class that students can register to and Teachers can teach by clicking on “Create Class”
4. Register a new student onto the system by clicking on “Register Student”
5. Enroll a student into a class by clicking “Enroll in Class”
6. When necessary Mark the attendance registers by clicking on “Mark Attendance” note that lists available classes which have students registered to take it.
   1. A user can backdate marking the register.
   2. The user could also postdate the attendance register
   3. The attendance is marked as submitted by the Teacher logged or submitting the register not the one designated as a teacher.
7. Reporting just lists the attendance register for any give period